

Jihočeská univerzita v Českých Budějovicích

Pedagogická fakulta

Katedra informatiky



Formátování www stránek pomocí CSS Level 3

Bakalářská práce

Josef Lombart

Vedoucí práce: PaedDr. Petr Pexa

ČESKÉ BUDĚJOVICE 2007

Abstrakt

Cílem této bakalářské práce je vypracovat příručku pro kaskádové styly CSS level 3, jako aktuální metody pro sazbu a formátování internetových stránek. Provést její porovnání s tradičními postupy pomocí starších verzí CSS a otestovat funkčnost v nejnovějších dostupných verzích prohlížečů webových stránek. Práce by měla být v České Republice unikátní publikací.

Abstract

The target of my bachelor's work is to develop an handbook for cascade styles CSS level 3 as actual methods for composition and formatting the web pages and also to accomplish the comparison with traditional techniques using the previous CSS versions and also test the functionality in the last available version of web browsers. The work should be an unique publication in the Czech republic.

Děkuji **PaedDr. Petru Pexovi** za odborné a organizační vedení při zpracování této práce.

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách.

24.dubna 2007

Josef Lombart

Obsah

1	Úvod.....	7
2	Syntaxe.....	8
3	Presentation levels.....	12
4	Reader media type.....	14
5	Selektory	15
6	Kaskádovost a dědičnost.....	26
7	Jmenné prostory	30
8	Hodnoty a jednotky.....	33
9	Barva	41
10	Řeč	47
11	Hypertextové odkazy	58
12	Seznamy	60
13	Písma.....	67
14	Písma pro web.....	75
15	Text	79
16	Box model.....	90
17	Pozadí a okraje.....	109
18	Pokročilé rozmístění prvků	126
19	Základní uživatelské rozhraní	137
20	Vícesloupcové uspořádání	145
21	Ruby.....	149
22	Line	153
23	Stránkovaná média.....	163
24	Generovaný a nahrazovaný obsah.....	172
25	Generovaný obsah pro stránkovaná média	177
26	Závěr	196

1 Úvod

Během posledních deseti let se stal internet nedílnou součástí života většiny z nás. Zatímco dříve internet využívali především studenti a vědečtí pracovníci pro komunikaci a výměnu dat, dnes internet nabízí široké spektrum informací, zábavy a možností pro podnikání. Kdo chce být v dnešní době úspěšný a známý, musí mít kvalitní webovou prezentaci.

První webové stránky měly stejné znaky. Byly jednoduché, strohé a statické. Tvůrce webu měl k dispozici pouze omezenou paletu barev, nízké rozlišení obrazovky a základní formátování textu. Výsledkem byly vzájemně si podobné, pro návštěvníka ne příliš atraktivní, stránky.

S rozšířením jazyka HTML začaly vznikat graficky vyspělejší prezentace. Avšak ani HTML není ideálním jazykem pro formátování. Hlavní nevýhodou je zejména nutnost formátovat každý prvek zvlášť. To znamená, že i malá úprava vzhledu stránky vyžaduje velkou úpravu zdrojového kódu.

Vznik kaskádových stylů (CSS – Cascade Style Sheet) umožnil oddělit formátování stránky od jejího obsahu. Můžeme tedy vytvořit šablonu stylů, které poté přiřadíme k jednotlivým prvkům. Tato metoda přináší mnohé usnadnění. Například změna velikosti písma pro nadpis druhého řádu znamená pouze změnu jedné řádky namísto zdlouhavého procházení celého dokumentu a dílčích oprav. Další výhodou je možnost použít jednu šablonu pro více dokumentů.

Práce na verzi CSS level3 začaly již v roce 1998. Skládá z cca třiceti oddělených specifikací, které se nazývají moduly. Tato filosofie umožňuje na jednotlivých modulech pracovat nezávisle. Modul se musí nacházet v jednom z pěti stupňů vývoje: pracovní návrh, přestudování změn, kandidát na schválení, kandidát na doporučení a schváleno. V době, kdy dokončuji tuto práci, jsou pouze tři moduly ve stavu kandidáta na schválení.

Cílem této práce je přiblížit možnosti poslední verze kaskádových stylů, tzn. popsat všechny dostupné moduly a vyzkoušet jejich implementaci v prohlížečích: FireFox 2.0, Opera 9.2 a Internet Explorer 7.0.

2 Syntaxe

Tato část popisuje základní syntaxi CSS3 a konvence užívané v ostatních modulech. Syntaxe má nějaké požadavky na obsluhu chyb, ale to je spíše záležitost prohlížečů.

2.1 Obecná syntaxe

2.1.1 Znaky a argumenty

Všechny CSS šablony jsou nezávislé na velikosti písmen. Výjimkou ty jsou části, které jsou mimo kontrolu CSS. Např. hodnoty *id* a *class* velikost písmen rozlišují. Naopak některé elementy v HTML písmena nerozlišují, ale v XML ano.

V CSS3 jsou identifikátory tvořeny znaky A-Z a 0-9 a znaky ze znakové sady ISO 10646 od 161 výše. Dále jsou to znaky "-" a "_". Identifikátory nesmí začínat číslem nebo "-" a číslem. Mohou také obsahovat escape sekvence a jakýkoli znak vyjádřen pomocí číselného kódu. Např. identifikátor B&W lze zapsat jako "`B\&W\?`" nebo "`B\26 W\3F`".

Lomítko "\" uvozuje tři druhy escape sekvencí.

1. Lomítko uvnitř řetězce, následováno přechodem na novou řádku, je ignorováno.
2. Ruší význam speciálních znaků.
3. Umožňuje odkázat na znak z kódové tabulky.

Escape sekvence jsou vždy považovány za část identifikátoru nebo řetězce.

2.1.2 Klíčová slova

Klíčová slova jsou speciální identifikátory. Nesmí být umístěna uvnitř uvozovek nebo apostrofů.

Příklad:

red

je klíčové slovo

```
width: "auto";  
border: "none";  
background: "red";  
nejsou klíčová slova!
```

2.1.3 '@' pravidla

Tato pravidla začínají znakem '@' ihned následovaným identifikátorem (např. *@import*, *@page*). Pokud se prohlížeč setká s nerozpoznatelným pravidlem, pak by ho měl ignorovat a pokračovat dalším.

2.1.4 Bloky

Blok je obsah uvnitř složených závorek. Mohou se tam vyskytnout jakékoliv znaky. Kulaté, složené, hranaté závorky musí být párové. Stejně tak apostrofy a uvozovky. Uvozené znaky jsou parsovány na řetězce.

2.1.5 Deklarace a vlastnosti

Deklarace se skládá z vlastnosti následované dvojtečkou a příslušné hodnoty. Okolo mohou být bílé znaky. Vícenásobné deklarace pro jeden selektor musejí být odděleny středníkem.

Příklad:

```
h1 { font-weight: bold }  
h1 { font-size: 2em }  
h1 { line-height: 1.2 }  
h1 { font-family: Helvetica, Arial, sans-serif }  
h1 { font-variant: normal }  
h1 { font-style: normal }
```

je stejné jako

```
h1 {  
  font-weight: bold;  
  font-size: 2em;  
  line-height: 1.2;  
  font-family: Helvetica, Arial, sans-serif;
```

```
font-variant: normal;
font-style: normal
}
```

2.2 Přiřazování šablon k dokumentům

2.2.1 Typy médií

Jedna z nejdůležitějších možností šablon je, že specifikují, jak má být dokument prezentován na různých médiích: na obrazovce, na papíře, pomocí syntetizátoru řeči, pomocí zařízení pro slepé, atd.

Některé CSS vlastnosti jsou navrženy pouze pro určitá média (např. vlastnost *page-break-before* se aplikuje pouze pro stránkovaná média). Přesto lze některé vlastnosti použít pro různá média, ale s různými hodnotami. Např. vlastnost *font-size* je použitelná pro obrazovku i pro tisková média. Tyto dva typy médií jsou ale natolik odlišné, takže musíme pro stejnou vlastnost použít rozdílné hodnoty. Na obrazovce většinou používáme větší písmo než na papíře. Proto je nutné vybrat, pro která média je šablona vytvořena.

2.2.2 Specifikace šablon závislých na médiích

Média můžeme specifikovat dvěma způsoby: pomocí pravidla *@media* nebo pomocí pravidla *@import*.

Příklad:

```
@media screen, print {
    body { line-height: 1.2; }
}
@import url("styl.css") screen;
```

2.2.3 Používané typy médií

Jména pro CSS média jsou zvolena tak, aby co nejlépe vystihovala smysl jejich použití. Jejich jména odpovídají normě. Jejich popis je pouze informativní. Názvy médií nezávisí na velikosti písmen.

all – vhodné pro všechna zařízení.

braille – určeno pro braillovo čtecí zařízení.

embossed – určeno pro tiskárny braillova písma.

handheld – určeno pro kapesní počítače (malý displej)

print – určeno pro tisk

projection – určeno pro projekci.

screen – určeno primárně pro monitory.

speech – určeno pro syntetizátory řeči.

tty – určeno pro média s pevným rozlišením.

tv – určeno pro televize (nižší rozlišení, barvy, možnost zvuku)

3 Presentation levels

Každý element v dokumentu má celočíselnou hodnotu, která vyjadřuje úroveň prezentace. Elementy, které jsou pod, nad nebo na stejné úrovni mohou být různě stylovány. Tato možnost má dvojí zajímavé využití. Za prvé, můžeme vytvořit prezentaci s přechody mezi stránkami. Např. položky seznamu mohou postupně přijíždět ze strany. Za druhé, můžeme vytvořit osnovu dokumentu, kde budou zobrazeny pouze nadpisy zvolené úrovně.

Každý element spadá do jedné ze tří pseudotříd:

:below-level – prvky pod danou úrovní.

:at-level – prvky na stejné úrovni.

:above-level – prvky na vyšší úrovni.

3.1 Vlastnost "presentation-level"

- Vlastnost nastavuje úroveň pro prezentaci.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnota: `<integer>` (explicitní nastavení hodnoty) | `"same"` (přiřadí elementu stejnou úroveň, jako má předešlý prvek) | `"increment"` (přiřadí elementu o jednu vyšší úroveň, než má předešlý prvek)
- Implicitní hodnota: `0`.

Příklad:

```

value
EPL

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
0
<STYLE>
0
  html { color: black; background: white }
  @media projection {
    h1 { page-break-before: always }
    li { presentation-level: increment }
    :below-level { color: black }

```

```

        :at-level { color: red }
        :above-level { color: silver }
    }
</STYLE>
<BODY>                                0
<H1>Strategie</H1>                 0
<H2>Naše strategie</H2>               0
<UL>                                    0
    <LI>rozděl                          1
    <LI>panuj                            2
</UL>
<H2>Jejich strategie</H2>            0
<UL>                                    0
    <LI>popleť                          1
    <LI>propaguj                        2
</UL>

```

Příklad postupné prezentace se zvýrazněním aktuálního prvku.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>                                  0
<STYLE>
    h1 { presentation-level: 0; }
    h2 { presentation-level: 1; }
    h3 { presentation-level: 2; }
    body * { presentation-level: 3; }
    :above-level { display: none; }
</STYLE>
<BODY>                                  0
<H1>Strategie</H1>                 3
<H2>Naše strategie</H2>               2
<UL>                                    0
<LI>rozděl                             0
<LI>panuj                               0
</UL>
<H2>Jejich strategie</H2>            2
<UL>                                    0
<LI>popleť                             0
<LI>propaguj                           0
</UL>

```

Příklad použití prezentace pomocí hlavních bodů.

4 Reader media type

HTML, XML, SVG, CSS a jiné formáty mohou připojit jeden nebo více šablon, které pomáhají specifikovat, jak bude dokument interpretován pro čtenáře. Media Queries mohou být připojeny k indikátorům, které ukazují, pro jaký typ zařízení nebo médií jsou šablony navrženy, např. velká barevná obrazovka, malá jednobarevná obrazovka, tiskárna, zpětný projektor nebo syntetizátor řeči.

Media Queries obsahují typ média (obrazovku, tisk, projekci, řeč, atd.) volitelně s dalšími omezeními (min-width, max-width, color, atd.). 'Reader' je typ médií, který používá zařízení jako je snímání obsahu obrazovky, který najednou zobrazuje dokument po vizuální stránce a čte ho za pomoci hlasového syntetizátoru.

Klíčové slovo **reader** je "typ médií" definován v Media Queries. Může být použito na stejných místech jako Media Queries.

Reader se vylučuje se všemi ostatními typy médií definovanými v Media Queries, mimo média s klíčovým slovem *all*. To znamená, že prohlížeč, který používá pravidla označená pro *reader*, nesmí ve stejné chvíli používat pravidla pro jiná média.

Přesto může jeden prohlížeč pracovat s různými módy. Například může poskytovat náhled před tiskem (použitím stylu pro *print* media) nebo simulovat kapesní počítač (použitím stylu pro *handheld*).

Tato specifikace nedefinuje, co má prohlížeč pro daná média použít. Prohlížeč (nebo uživatel) může vybrat, která média se pro daná média nejvíce hodí. Následující pokyny jsou pouze informativní. Prohlížeč použije média reader v těchto případech:

- snímání obsahu obrazovky (často navržený jako snadná pomoc)
- zobrazení dokumentu na obrazovce a jeho přečtení
- rolování obrazovky synchronně s řečí, čeká, než začne uživatel rolovat
- dovolí uživateli rozpoznat místo mluvení, ukáže nebo použije rychlý přesun vpřed/ rychlý přesun vzad nebo podobné funkce
- může také poskytnout zobrazený text ve slepeckém zařízení.

5 Selektory

Selektory reprezentují strukturu. Tato struktura je použita jako podmínka, určující jaké elementy odpovídají selektoru ve stromovém dokumentu nebo jako popis HTML, či XML části, odpovídající struktuře. Mohou být rozsáhlé, od jednoduchých jmen prvků k související reprezentaci. Selektory jsou nyní modulem CSS3 a jsou nezávislou specifikací. Jiné specifikace se mohou nyní odkazovat na tento dokument nezávisle na CSS

5.1 Rozdíly od CSS2

- Jasnější seznam základních definic (selektor, skupiny selektorů, jednoduché selektory, apod.).
- Univerzální selektory a vlastnosti selektorů, možnost použít jmenné prostory.
- Nové kombinace.
- Nové jednoduché selektory obsahující podřetězec odpovídající vlastnostem selektorů a nové pseudotřídy.
- Nové pseudoelementy a všechny nyní začínají s „::“, což je nová konvence pseudoelementů.
- Přepracovaná gramatiky selektorů.
- Profily přidané normy začleněných Selektorů a definování kolekce selektorů, které jsou aktuálně podporovány každou specifikací.

5.2 Skupiny selektorů

Pokud mají nějaké selektory stejné vlastnostmi, mohou být seskupeny do jednoho seznamu odděleného čárkami.

Příklad: Různé zápisy stejných vlastností

```
h1 { font-family: sans-serif }
h2 { font-family: sans-serif }
h3 { font-family: sans-serif }
h1, h2, h3 { font-family: sans-serif }
```

5.3 Jednoduché selektory

Typové selektory umožňují nepovinné komponenty ve jmenném prostoru. Předpony deklarované ve jmenném prostoru, mohou být přiřazeny ke jménu prvku, který je oddělený jmenným prostorem, oddělovačem (|). Jmenný prostor může být ponechán prázdný, což znamená, že selektor znázorňuje elementy se žádným deklarovaným jmenným prostorem.

- ns|E elementy se jménem E ve jmenném prostoru ns
- *|E elementy se jménem E v jakémkoliv jmenném prostoru, obsahující ty které jsou mimo jakýkoliv deklarovaný jmenný prostor
- |E elementy se jménem E bez jakéhokoliv deklarovaného jmenného prostoru
- E jestliže není specifikován žádný jmenný prostor, je toto ekvivalentní s *|E. Jinak je ekvivalentní s ns|E kde ns je defaultní jmenný prostor.

Příklad:

```
@namespace foo url(http://www.example.com);  
foo|h1 { color: blue }
```

nastaví jen elementy h1 v "http://www.example.com" jmenném prostoru

```
foo|* { color: yellow }
```

nastaví všechny elementy v "http://www.example.com" jmenném prostoru

```
|h1 { color: red }
```

nastaví jen element h1, mimo jakýkoliv deklarovaný jmenný prostor

```
*|h1 { color: green }
```

nastaví element h1 v kterémkoliv jmenném prostoru

```
h1 { color: green }
```

není deklarován žádný defaultní jmenný prostor

5.4 Atributy selektorů

Tyto selektory dovolují reprezentaci vlastností prvkům. Pokud je element použit jako výraz, pak musí být brán zřetel, aby se selektor shodný s cílovým elementem.

5.4.1 Výskyt vlastností a hodnoty selektorů

CSS2 uvedlo čtyři atributy selektorů:

- **[att]** Reprezentující atribut att, jakákoliv hodnota atributů.
- **[att=val]** Reprezentuje atribut att s přesnou hodnotou "val".
- **[att~=val]** Představuje atribut att jehož hodnota je mezerou oddělený seznam slov, jedno které je přesně "val". Když je tento selektor použit, slova v této hodnotě nesmí obsahovat mezeru (od místa, kde jsou odděleny právě mezerou).
- **[att|=val]** hodnota tohoto selektoru může být buď "val" nebo může začínat "val" bezprostředně následovaný mezerou (-). Toto zejména odpovídá subkódu. Jazyk subkódu je nastíněn v pseudotřídě :lang.

Hodnoty atributů musí být identifikátory nebo řetězce. Case-sensitivity jmen atributů a hodnot v selektorech závisí na jazyku dokumentu, ve kterém jsou psány.

Příklad:

```
h1[title]
```

Selektor představuje element h1 který má titulek atributu, kterékoli jeho hodnoty.

```
span[class=example]
```

Selektor představuje element span, jehož atribut třídy má hodnotu přesně "example".

```
span[ahoj="Praha"][sbohem="Brno"]
```

Mnohonásobný atribut selektorů může být použit k reprezentaci různých atributů elementů, nebo různé podmínky stejných atributů. V tomto případě selektor je aplikován na element span jehož atribut *ahoj* nabývá přesně hodnoty "Praha" a jehož atribut *sbohem* má přesně hodnotu "Brno".

```
link[hreflang|"en"]
```

Selektor je aplikován na ty, jejichž hodnota atributu hreflang začíná na "en", obsahuje "en". Např. "en-US", a "en-cockney".

5.4.2 Podřetězce odpovídající atributům selektorů

V CSS3 byly přidány tři dodatečné atributy selektorů, které jsou poskytovány pro odpovídající podřetězce v hodnotách atributů:

- `[att^=val]` - atribut `att` jehož hodnota začíná předponou "val".
- `[att$=val]` - atribut `att` jehož hodnota končí příponou "val".
- `[att*=val]` - atribut `att` jehož hodnota obsahuje při nejmenším jednu instanci podřetězce "val".

Hodnoty atributů musí být také identifikátory nebo řetězce. Case-sensitivita názvů atributů v selektorech závisí na jazyku dokumentu, ve kterém stylové předpisy tvoříme.

Příklad:

```
p[title*="hello"]
```

Odstavec s atributem `title` jehož hodnota obsahuje podřetězec "hello".

5.5 ID selektory

Dokument může obsahovat typ ID. Ten je zvláštní tím, že žádné dva atributy v dokumentu nemohou mít stejnou hodnotu. V html jsou ID atributy pojmenovány "id".

ID atributy umožňují přiřadit identifikátor právě jednomu elementu. ID selektory jsou ve tvaru "#`id`" ihned následovaný hodnotou identifikátoru.

Příklad:

```
h1#chapter1
```

Selektor reprezentuje h1 element, jehož `id` je "chapter1".

```
#chapter1
```

Selektor reprezentuje jakýkoliv element, jehož `id` je "chapter1".

5.6 Pseudotřídy

Tento koncept povoluje výběr na základě informací, které leží vně dokumentu nebo nemohou být vyjádřeny pomocí jiných jednoduchých selektorů.

Pseudotřída vždy obsahuje dvojtečku (:), která je následována jménem a volitelnou hodnotou mezi závorkami. Pseudotřída je dovoleno používat u všech jednoduchých selektorů. Některé z nich se vzájemně vylučují, zatímco jiné mohou být aplikovány současně na tentýž prvek. Pseudotřídami mohou být dynamické, v tom smyslu, že prvek ji může nabýt nebo ztratit zatímco uživatel pracuje se stránkou.

5.6.1 Dynamické pseudotřídami

Dynamické pseudotřídami klasifikují element na jiných vlastnostech než na jejich jménu, attributech nebo obsahu. Jsou založeny na vlastnostech, které nemohou být odvozeny z dokumentu.

:link, *:visited* – Prohlížeče rozlišují mezi navštíveným nebo nenavštíveným odkazem.

:link se aplikuje na nenavštívené odkazy.

:visited se aplikuje na navštívené odkazy.

:hover, *:active*, *:focus* – Interaktivní prohlížeče reagují na akci uživatele.

:hover se aplikuje, pokud uživatel najede myší na element.

:active se aplikuje, pokud je element aktivován. Např. kliknutím levého tlačítka.

:focus se aplikuje, pokud je element uživatelem zaměřen.

Příklad:

`a:link`

nenavštívený odkaz

`a:visited`

navštívený odkaz

`a:hover`

najetí myši

`a:active`

aktivní odkaz

5.6.2 Pseudotřída `:target`

Některá URI se odvolávají na umístění uvnitř zdroje. Tento druh URI končí znakem "#" a je následovaný identifikátorem. Cílové umístění může být reprezentováno pseudotřídou `:target`.

Příklad:

```
p.note:target
```

Tento selektor reprezentuje odstavec třídy `note`, na který je odkazováno.

```
*:target { color : red }  
*:target::before { content : url(target.png) }
```

Cíl odkazu bude červený a bude před ním umístěn obrázek `target.png`

5.6.3 Pseudotřída `:lang`

Jestliže jazyk dokumentu specifikuje, jak bude přednastaven jazyk prvků, je možné psát selektory, tak, že reprezentují prvky založené na tomto jazyce. Například v HTML4, je jazyk určený kombinací atributu `lang`, elementem `meta` a popřípadě informací z protokolu (tak jako v hlavičce HTTP). XML užívá atribut nazývaný `xml:lang`, a mohou zde být i jiné dokumenty specifikující jazyk metod pro nastavení jazyka.

Pseudotřída `:lang(C)` popisuje element, který je v jazyku C. Jazyk C, tak jak je specifikovaný v HTML 4.01 a v RFC 3066.

Příklad:

```
html:lang(fr-be)  
html:lang(de)  
:lang(fr-be) > q  
:lang(de) > q
```

Tento příklad ukazuje HTML dokument, který je psán ve vlámštině, francouzština nebo němčině. Dva následující selektory zachycují `q` citaci v libovolném elementu, ve všech třech jazycích.

5.6.4 Pseudotřída uživatelského rozhraní

`:enabled` autorovi umožňuje přizpůsobit vzhled prvkům uživatelského rozhraní v nějakém módním stylu.

:disabled autorovi umožňuje zakázat vzhled prvkům uživatelského rozhraní.

:checked uživateli umožňuje elementy upravovat, zaškrtnávat je apod. Vztahuje se to především na checkboxy a radiobuttony. Když je takovýto prvek zaškrtnut, pak je aktivována tato pseudotřída, která je dynamická. Používá se na všechna media.

:indeterminate se používá na stejné prvky jako předchozí, ale jen v takovém případě, že tyto prvky jsou i nejsou zaškrtnuté. To může být způsobeno atributem prvku. Pseudotřída je ve stylu listů definována pomocí dvojtečky jako oddělovače (`:pseudotřída {specifikace}`). Následné volání potom probíhá zcela automaticky, a pokud prvek odpovídá některému kontextu, automaticky se provede formátování dle daného kontextu.

5.6.5 Strukturální pseudotřídy

Tento koncept povoluje výběr na základě informací, které leží vně dokumentu nebo nemohou být vyjádřeny pomocí jiných jednoduchých selektorů nebo jejich kombinacemi.

:root – zastupuje každý element, který je v dokumentu kořenovým. V HTML 4 je to element HTML. V XML je to kterýkoliv vhodný pro DTD nebo schéma a jmenný prostor pro ten který XML dokument.

:nth-child($an+b$) – říká, že element má již někde dříve v dokumentovém $an+b-1$ sourozenců. Pro daná kladná celá čísla nebo nulové hodnoty n . Jinak řečeno, toto dává potomky elementů do skupin jednotlivých elementů. Toto například dovoluje selektorům adresovat každou jednotlivou řádku a může být použita například ke střídání barev nebo odstavců textu v cyklech po čtyřech. Hodnoty a a b , mohou být nastaveny na nulu, záporné, či kladné celé číslo. Index prvního potomka elementu je 1. `:nth-child()` může mít také hodnotu 'odd' a 'even' pro argumenty. 'Odd' má stejný význam jako $2n+1$, a 'even' má stejný výraz jako $2n$.

Příklad:

```
tr:nth-child(2n+1)
```

```
tr:nth-child(odd)
```

každá lichá řádku HTML tabulky

```
tr:nth-child(2n)
tr:nth-child(even)
```

každá sudá řádka HTML tabulky

```
p:nth-child(4n+1) { color: navy; }
p:nth-child(4n+2) { color: green; }
p:nth-child(4n+3) { color: maroon; }
p:nth-child(4n+4) { color: purple; }
```

Střídání barev v odstavci v CSS

Jestliže je $a=0$, pak není použito žádného opakování. Takže například `:nth-child(0n+5)` nastaví pouze pátého potomka. Pokud je $a=0$, pak není třeba tuto část zahrnovat, tudíž syntaxe se zjednoduší na `:nth-child(b)`.

Jestliže se $a = 1$, pak může být zápis zjednodušen na `:nth-child(n+b)`.

Příklad:

```
bar:nth-child(1n+0)
bar:nth-child(n+0)
bar:nth-child(n)
```

všechny elementy bar jsou specifikovány (0,1,1)

```
tr:nth-child(2n)
```

každá sudá řádka HTML tabulky

Hodnota a může být záporná, ale pouze pro kladné hodnoty $an+b$,

```
html|tr:nth-child(-n+6)
```

prvních šest řádků XHTML tabulky

`:nth-last-child(an+b)` – tento zápis znamená, že prvek, který má někde v dokumentu nejméně $an+b-1$ příbuzného. Syntaxe zápisu je stejná jako u předešlé pseudo-třídy.

```
tr:nth-last-child(-n+2)
```

dvě poslední řádky HTML tabulky

```
foo:nth-last-child(odd)
```

všechny liché řádky ve jmenném prostoru foo v rodičovských elementech, počítáno od poslední.

`:nth-of-type(an+b)` – tento zápis znamená, že element, který má $an+b-1$ má někde dříve v dokumentu nějakého příbuzného se stejným názvem.

```
img:nth-of-type(2n+1) { float: right; }  
img:nth-of-type(2n) { float: left; }
```

Tento příklad umožňuje CSS střídat pozice plovoucích obrázků.

:nth-last-of-type() – Tato pseudotřída je analogicky opačná k třídě předchozí, tudíž příbuzný prvek existuje někde dále v dokumentu.

```
body > h2:nth-of-type(n+2):nth-last-of-type(n+2)
```

Pomocí tohoto příkladu můžeme na všechny potomky elementu h2 XHTML dokumentu použít selektor, kromě první a poslední řádky.

:first-child – stejné jako *:nth-child(1)*

:last-child – stejné jako *:nth-last-child(1)*

:first-of-type – stejné jako *:nth-of-type(1)*

:last-of-type – stejné jako *:nth-last-of-type(1)*

:only-child – stejné jako *:first-child* nebo *:last-child*, ale s nižší specifikací

:only-of-type – stejné jako *:nth-of-type* nebo *:nth-last-of-type*, ale s nižší specifikací

:empty – Tato pseudotřída představuje taková prvek, který nemá vůbec žádného potomka.

5.6.6 Záporné pseudotřídy

:not(x) – jak již název napovídá, tato pseudotřída reprezentuje element, který není popsán žádným argumentem.

```
button:not([DISABLED])
```

Selektor se vztahuje na všechna tlačítka button v HTML dokumentu, která nejsou zakázaná.

```
html|*:not(:link):not(:visited)
```

Výše uvedená skupina selektorů se vztahuje na všechny prvky HTML kromě odkazů.

5.7 Pseudoelementy

Pseudoelementy tvoří abstrakci nad jazykem dokumentu. Např. jazyk nemá nástroj, jak přistoupit k prvnímu písmenu v textu elementu. Pseudoelementy umožňují autorovi přístup k jinak nedosažitelným informacím. Pseudoelementy by také měli autorovi způsob, jak se dotazovat na obsah, který ve zdrojovém souboru není (např. pseudoelementy `::before` a `::after` umožňují přístup ke generovanému obsahu).

V CSS3 je zavedena nová syntaxe v uvádění pseudoelementů a to, že jsou předcházeny dvojicí dvojteček ":", oproti předchozím definicím v kaskádních stylech. Je to z důvodů přehlednosti, aby se viditelněji odlišily od pseudotříd.

`::first-line` – celý první řádek bude zformátován tak, jak si určíme.

`::first-letter` – první znak bude zformátován tak, jak si určíme.

Příklad:

```
p::first-line { text-transform: uppercase }
```

první řádka bude psána kapitálkami

```
p::first-letter { font-size: 200%; font-style: italic;
                  font-weight:bold; }
```

první písmeno bude dvakrát větší, bude psáno kurzívou a tučně

`::selection` – text, který je označen uživatelem.

`::menu` – pomocí tohoto pseudoelementu můžeme specifikovat styl a polohu generovaného menu.

`::before` – může být použit pro popis generovaného obsahu před elementem.

`::after` – může být použit pro popis generovaného obsahu za elementem.

název	FireFox 2.0	Opera 9.2	Internet Explorer 7.0
<code>::first-line</code>	ano	ano	ne
<code>::first-letter</code>	ano	ano	ne
<code>::selection</code>	ne	ne	ne
<code>::before</code>	ano	ne	ne
<code>::after</code>	ano	ano	ne

Tabulka 1 - Funkčnost pseudoelementů

5.8 Gramatika

Gramatika definuje syntaxi selektorů.

- *: žádný nebo více výskytů
- +: jeden nebo více výskytů
- ?: žádný nebo jeden výskyt
- |: oddělené alternativy
- []: seskupování

6 Kaskádovost a dědičnost

Jeden ze základních funkčních návrhů kaskádových stylů, je povolit více šablon, eventuálně pro různé zdroje, aby ovlivnily tvorbu dokumentu. Tento modul popisuje, jak vybrat správnou deklaraci. Mechanismus, který je pro tento výběr použit se nazývá kaskádování. Pokud pro element vlastnost není přímo deklarována, pak se použije vlastnost zděděná od rodičovského elementu nebo se nastaví na implicitní hodnotu.

Vstupem kaskádování a dědičnosti jsou:

- Nastavení deklarace, které mohou být aplikovány na prvky, či vlastnosti v kombinaci s otázkou. Předpokládá se, že deklarace není používána a není zahrnuta v nastavení.
- Zděděně hodnoty vlastností.
- Počáteční hodnoty vlastností.

Výstupem výběru vlastnosti pomocí kaskádovosti a dědičnosti je jediná hodnota, která se nazývá *specifikovaná hodnota*. Tuto hodnotu prohlížeč vypočítá pomocí následujících pravidel:

1. Pokud procesem vznikne vítězná deklarace a její hodnota bude jiná než *initial* nebo *inherit*, pak je specifikovanou hodnotou.
2. Pokud je vítěznou hodnotou *inherit*, pak se specifikovanou hodnotou stává zděděná hodnota.
3. Pokud je vítěznou hodnotou *initial*, pak se specifikovanou hodnotou stává implicitní hodnota.
4. Pokud je hodnota zděděná, pak se stává specifikovanou hodnotou.
5. Pokud je implicitní, pak se stává specifikovanou hodnotou.

6.1 Pravidlo `@import`

Toto pravidlo uživateli umožňuje importovat pravidla z jiných šablon. Pravidlo `@import` musí respektovat pravidla `@charset` a být nadřazeno všem ostatním pravidlům. Klíčové slovo `@import` musí být ihned následováno URI šablony. Je povolen i řetězec, který bude interpretován, jako by byl hodnotou pro funkci `url()`.

Aby prohlížeč nepoužil zdroje pro nepodporovaný typ medií, může autor specifikovat, pro která media chce `@import` pravidlo použít. Tyto podmínky jsou umístěny za URI a jsou odděleny čárkou. Další hodnotou může být řetězec, který určuje název šablony.

Příklad:

```
@import "mystyle.css";
@import url("mystyle.css");
@import url("fineprint.css") print;
@import url("bluish.css") projection, tv;
@import url("narrow.css") handheld and (max-width:
400px);
@import url(style4.css) screen "Four-columns and dark";
```

6.2 Pravidlo `@media`

Pravidlo `@media` specifikuje cíl typů medií (oddělených čárkami) jako sadu pravidel (uvnitř složených závorek). Pravidlo umožňuje vytvořit šablony pro různá média v jedné. Volitelnou hodnotou je název pravidla.

```
@media all "Ultra blue" {
  body { background: blue }
  h1 { color: cyan }
}
@media screen, print "Simple and light" {
  body { background: white; color: #333 }
  h2 { font-size: bigger }
}
```

6.3 Kaskádovost

Úkolem kaskádovosti je najít vítěznou deklaraci a aplikovat ji na dané elementy. Prohlížeč musí třídit deklarace podle následujících kritérií, v pořadí důležitosti:

- Důležitost. V CSS3, váha deklarace je založena na původu deklarace a je označena levellem důležitosti. Přečtěte si následující sekce, abyste věděli jak vypočítat důležitost. Deklarace s největší váhou vyhrává.
- Specifikace. Selektory popisují jak vypočítat specifikaci. Deklarace s nejvyšší specifikací vyhrává.
- Pořadí vzhledu. Poslední deklarace vyhrává.

Řadící proces pokračuje, dokud není nalezena první vyhrávající deklarace.

6.4 Dědičnost

Dědičnost je způsob rozšiřování vlastností od rodičovských prvků potomkům. Dědičnost znamená, že specifikovaná hodnota elementu je získána z hodnoty rodičovského elementu. Kořenový element, který nemá žádného předka, používá implicitní hodnotu.

Některá vlastnosti jsou dědičné. To znamená, že i když elementu nespecifikujeme žádnou hodnotu, tak je mu přiřazena zděděná hodnota.

Všechny vlastnosti přijímají klíčové slovo *inherit*, které explicitně specifikuje, že hodnoty budou přeneseny od předků.

6.5 Počáteční hodnota

Každá vlastnost má svojí počáteční hodnotu, která je použita, pokud nemá prvek specifikovanou hodnotu pomocí kaskádovosti a dědičnosti. Může být také explicitně specifikována pomocí klíčového slova *initial*, které je možno použít pro všechny vlastnosti.

6.6 Vypočtení váhy

V pořadí třídění deklarace v procesu kaskádování, musí být známa váha deklarace. V CSS3 je váha deklarací založena na původu deklarace, a je označena úrovní důležitosti. Kaskádové styly mohou mít tři rozdílné zdroje: autora, uživatele a prohlížeč.

- Autor specifikuje kaskádní styly pro zdroj dokumentu podle konvencí jazyka dokumentů. Například, v HTML, mohou být kaskádní styly obsaženy v dokumentu nebo provázány zvenku.
- Uživatel může specifikovat informace stylů pro jednotlivé dokumenty. Například uživatel může specifikovat soubor, který obsahuje kaskádní styly nebo prohlížeč může poskytnout rozhraní, které generuje uživatelův stylový předpis.
- Prohlížeč musí používat standardní stylový předpis, dříve než všechny ostatní stylové předpisy dokumentu.

Každá deklarace CSS může mít dvě různé úrovně důležitosti: normální (který je standardní) a důležitá (která musí být označeny). Syntaxe CSS3 popisuje, jak jsou deklarace označovány jako důležité. Váha důležitosti kaskádních stylů z různých předpisů, vzestupně je:

1. Kaskádní předpis prohlížeče
2. Normální kaskádní předpis uživatele
3. Normální předpis autora
4. Důležitý předpis autora
5. Důležitý předpis uživatele

Standardně tato strategie dává větší váhu autorovi deklarace, než uživateli.

7 Jmenné prostory

Tato specifikace definuje syntaxi pro používání jmenných prostorů v CSS. Uvádí pravidlo `@namespace` pro deklaraci výchozího jmenného prostoru a pro svázání jmenného prostoru s příslušným prefixem. Tato specifikace také definuje syntaxi pro užívání těchto prefixů v názvech jmenných prostorů, ale neříká, kde jsou tato jmenná validní nebo co znamenají.

Mělo by být známo, že CSS klient, který nepodporuje tento modul, by měl všechna `@namespace` pravidla ignorovat. Syntaxe je navržena tak, že je upřednostněna ignorace pravidel před špatným vyložením.

7.1 Deklarace `@namespace`

Pravidlo `@namespace` deklaruje prefix jmenného prostoru a asociuje ho s daným jmenným prostorem. Tento prefix může být použit ve jménech uvozených s prefixem.

Příklad:

```
@namespace "http://www.w3.org/1999/xhtml";
```

Pravidlo nastavuje jako `http://www.w3.org/1999/xhtml` jako výchozí jmenný prostor.

```
@namespace svg "http://www.w3.org/2000/svg";
```

Pravidlo nastavuje prefix jmenného prostoru `svg`. Tam kde bude uveden prefix `svg` bude použit jmenný prostor `http://www.w3.org/2000/svg`.

7.1.1 Syntaxe

Syntaxe pro `@namespace` pravidla je následující:

```
namespace
:NAMESPACE_SYM S* [namespace_prefix S*]? [STRING|URI] S*
';' S*;
namespace_prefix
:IDENT;
@namespace vrací NAMESPACE_SYM
```

Každé @namespace pravidlo musí dodržovat všechna @charset a @import pravidla a být nadřazena všem ostatním pravidlům.

7.1.2 Pole působnosti

Jmenný prefix je deklarován pouze pro šablonu, uvnitř šablony které se @namespace pravidlo vyskytuje. Neplatí ani pro šablony, které jsou touto šablonou importovány, ani pro šablonu, které tuto šablonu importují.

7.1.3 Deklarace prefixů

Jmenný prefix reprezentuje jmenný prostor, pro který byl deklarován a můžeme ho použít pro označení jmenného prostoru při deklaraci prvků.

Pokud je při deklaraci prefix vynechán, pak je jmenný prostor brán jako výchozí. Výchozí jmenný prostor je použit u všech jmen, která nejsou uvozena prefixem. Je na modulech, aby zajistily, ve kterém kontextu lze jmenné prostory použít.

Jmenné prefixy jsou stejně jako jiné vlastnosti case-insensitive.

Pokud je jmenný prefix nebo výchozí jmenný prostor deklarovaný vícekrát, pak je použit jeho poslední výskyt.

7.2 CSS Qualified Names

Kvalifikovaná jména jsou jména, která jsou explicitně umístěna ve jmenném prostoru. K vytvoření kvalifikovaného jména v CSS syntaxi je třeba uvést lokální jméno uvozené prefixem, odděleným "|". Prefix reprezentuje jmenný prostor, pro který je jméno deklarováno. Pokud je prefix vynechán, pak jméno nepatří do žádného jmenného prostoru. Jako speciální znak lze použít znak "*", který značí příslušnost ke kterémukoliv jmennému prostoru.

Příklad:

```
@namespace toto "http://toto.example.org";  
@namespace "http://example.com/foo";
```

toto | A

A patří do jmenného prostoru <http://toto.example.org>

| B

B nepatří do žádného jmenného prostoru

* | C

C patří do každého jmenného prostoru

D

D patří do výchozího jmenného prostoru, tedy do <http://example.com/foo>

8 Hodnoty a jednotky

Tento modul popisuje možné hodnoty a jednotky, které můžeme v CSS používat. Hlavním účelem je definovat běžné hodnoty a jednotky, na které se pak budou ostatní moduly odkazovat.

8.1 Klíčová slova

V oficiálních definicích hodnot vlastností, jsou klíčová slova zapisována doslovně. Zde jsou pro příklad uvedeny hodnoty definice pro vlastnost *border-collapse* a jejich následné využití:

Příklad:

```
Value: collapse | separate | inherit  
table { border-collapse: separate }
```

Pro všechny vlastnosti CSS3 lze použít klíčového slova *inherit* a *initial*. Klíčová slova nesmí být psány v uvozovkách ani v apostrofech.

8.2 Čísla

V CSS3 používáme celých i reálných čísel. V případě celých čísel jsou značeny jako `<integer>`, v případě reálných jako `<number>`. Čísla jsou psána v desítkové soustavě. Znak pro desetinnou čárku je ".". Mohou nabývat kladných i záporných hodnot, kde znaménko je ihned následováno hodnotou. Pro kladné hodnoty je znaménko volitelné.

8.3 Kladná čísla

Kladná čísla mohou být z oboru celých i reálných čísel. V případě celých čísel jsou značeny jako `<non-negative integer>`, v případě reálných jako `<non-negative number>`. Čísla jsou psána v desítkové soustavě.

8.4 Čísla s jednotkou

8.4.1 Délkové jednotky

Mnoho parametrů nastavovaných v CSS je hodnota udávající míru nebo vzdálenost. Tyto hodnoty se zapisují jako čísla, ve které je vzdálenost zadána, bezprostředně následována jednotkou. Jestliže je číslice uvedena bez své jednotky, je jí implicitně přiřazena hodnota pixelů. Pokud je hodnota rovna nule, pak je jednotka volitelná.

Používané jednotky se dělí na absolutní a relativní.

8.4.2 Relativní délkové jednotky

Relativní jednotky jsou takové, jejichž hodnota není přesně definována a závisí na jiné hodnotě. U šablon, které využívají relativních jednotek, je jednodušší měnit měřítko (např. z displeje na laserovou tiskárnu).

Jednotka	vztahuje se k
em	velikosti elementu (velikosti rodičovskému elementu, vlastnosti font-size)
ex	x-výška fontu elementu
px	zobrazovacímu zařízení
gd	ke gridu definovanému layout-grid
rem	velikost písma kořenového elementu
vw	šířka zobrazení, platí viewport width = 100 vw
vh	výška zobrazení, platí viewport height = 100 vh
vm	menší z výšky a šířky zobrazení
ch	šířce přeškrtnutí nuly

Tabulka 2 - relativní délkové jednotky

Příklad:

```
h1 { line-height: 1.2em }
```

výška řádky bude o 20% větší než velikost elementu

```
h1 { font-size: 1.2em }
```

velikost písma bude o 20% větší než zděděná

```
p { layout-grid: strict both 20pt 15pt; margin: 1gd 3gd 1gd 2gd }
```

p element bude mít 15pt top margin, 60pt right margin, 15pt bottom margin a 40pt left margin

```
h1 { font-size: 8vw }
```

Poznámka. Pokud je šířka viewportu 200mm, pak je velikost elementu h1 16mm ((8x200)/100). Pokud se změní šířka viewportu (např. maximalizace prohlížeče), pak se příslušně zvětší i hodnota vw.

8.4.3 Absolutní délkové jednotky

Absolutní jednotky jsou takové, jejichž hodnota je přesně definována a je odvoditelná od základních jednotek tabulky SI. Jednotky je vhodné použít, pokud jsou známé fyzické vlastnosti na výstupu.

Jednotka	Definice
in	palce 1 in = 2.54cm
cm	centimetry
mm	milimetry
pt	points 1pt = 1/72in
pc	picas 1pc = 12pt

Tabulka 3 - absolutní délkové jednotky

Příklad:

```
h1 { margin: 0.5in }
h2 { line-height: 3cm }
h3 { word-spacing: 4mm }
h4 { font-size: 12pt }
h4 { font-size: 1pc }
```

8.4.4 Procenta

Procenta se udávají ve tvaru číslo, ihned následováno znakem "%". Každá vlastnost, která procenta podporuje, také definuje, ke které hodnotě se vztahuje.

Příklad:

```
h1 { font-size: +120%; font-weight: bold; text-align:
center }
smaller { font-size: -20% }
```

8.4.5 Úhly

Hodnoty úhlů jsou používány aurálními styly. Jsou ve formátu čísla, ihned následovaného jednotkou. Hodnoty úhlů by měly být normalizovány prohlížečem na interval 0-360 stupňů, přičemž platí: $-10\text{deg} = 350\text{deg}$.

- deg: stupně
- grad: gradiány
- rad: radiány
- turns: otočky

Příklad

$90\text{deg} = 100\text{grad} = 1.570796326794897\text{rad}$

Všechny tři hodnoty vyjadřují pravý úhel.

8.4.6 Čas

Časové hodnoty jsou používány aurálními styly. Jsou ve formátu kladného čísla, ihned následovaného jednotkou.

- ms: milisekundy
- s: sekundy

8.4.7 Frekvence

Hodnoty frekvence jsou používány aurálními styly. Jsou ve formátu kladného čísla, ihned následovaného jednotkou.

- Hz: hertze
- kHz: kilohertze

Poznámka: 200Hz je nízký tón, 6kHz je vysoký tón.

8.5 Řetězce

Řetězce mohou být psány uvnitř uvozovek nebo apostrofů. Uvnitř uvozovek (apostrofů) nemohou být uvozovky (apostrofy). Pro vložení speciálních znaků (uvozovky, apostrofy, nový řádek, tabulátor) se používají tzv. escape sekvence.

Příklad:

```
"toto je 'string'"
"toto je \"string\""
'toto je "string"'
'toto je \'string\''
```

8.6 Funkce

8.6.1 Funkce attr

V CSS2.1 vrací funkce attr string. V CSS3 může vrátit různé typy. Syntaxe je: `'attr(' ident [',' <type> [',' <value>]?]? ')'`.

První argument je jméno atributu. Může obsahovat prefix oddělený "|". Mezi separátorem nesmí být mezery. Druhý argument je typ, který říká prohlížeči, jak interpretovat hodnotu atributu. Je volitelný, ale musí být, pokud bude použit i třetí argument. Třetí je CSS hodnota, která musí být validní s umístěním výrazu. Pokud není, pak není validní celý výraz. Pokud chybí první atribut, pak nemůže být výraz zpracován a bude vrácena hodnota třetího argumentu, nebo defaultní hodnota typu.

Vhodné typy pro druhý argument

- string – Hodnota atributu bude interpretována jako CSS *string*. Defaultní hodnota je prázdný řetězec.
- color – Hodnota atributu bude interpretována jako CSS *color*. Defaultní hodnota závisí na prohlížeči, ale musí být shodná s jako výchozí nastavení.
- url – Hodnota atributu bude interpretována jako výraz *url()*. Defaultní hodnota závisí na prohlížeči, směřuje na neexistující stránku s vygenerovaným error textem (např. by to neměla být FTP URI, kterou způsobí DNS error, nebo HTTP URI, která vyvolá error 404. Měla by to být nepopsaná error stránka).

- `integer`: – Hodnota atributu bude interpretována jako CSS *integer*. Defaultní hodnota je 0. Měla by být také použita, pokud vyjde hodnota mimo rozsah.
- `number` – Hodnota atributu bude interpretována jako CSS *number*. Defaultní hodnota je 0. Měla by být také použita, pokud vyjde hodnota mimo rozsah.
- `length`, `angle`, `time`, `frequency` – Hodnota atributu bude interpretována jako CSS *délka*, *úhel*, *čas*, *frekvence* a příslušná jednotka. Defaultní hodnota je 0. Měla by být také použita, pokud vyjde hodnota mimo rozsah.
- `em`, `ex`, `px`, `gd`, `rem`, `vw`, `vh`, `vm`, `mm`, `cm`, `in`, `pt`, `pc`, `deg`, `grad`, `rad`, `ms`, `s`, `Hz`, `kHz`, `%` – Hodnota atributu bude interpretována jako reálné číslo s příslušnou hodnotou. Defaultní hodnota je 0 v příslušných jednotkách.

Příklad:

```
<stock>
<wood length="12"/>
<wood length="5"/>
<metal length="19"/>
<wood length="4"/>
</stock>
stock::before {
display: block;
content: "To scale, the lengths of materials in stock
are:";
}
stock >
* { display: block; width: attr(length, em); /* default 0
*/
height: 1em;
border: solid thin;
margin: 0.5em;
}
wood {
background: orange url(wood.png);
}
metal {
background: silver url(metal.png);
}
```

toto také používá možné rozšíření vlastností 'obsahu' pro zvládnutí nahrazeného obsahu a alternativ nedostupného, poškozeného nebo nepodporovaného obsahu.

```
img {
content: replaced attr(src, url), attr(alt, string, none);
height: attr(height, px, auto);
width: attr(width, px, auto);
}
```

8.6.2 Funkce url

Url poskytuje adresu zdroje na síti. Obecnější název je URI. Z historických důvodů je jméno URI funkce *url*. *Url* funkce má URI jako argument. URI může být uzavřena apostrofem nebo uvozovkami. Pokud se v url vyskytují závorky, středníky, uvozovky, apod., musí být nahrazeny escape sekvencemi.

Příklad:

```
body { background: url("grafika/pozadi.gif") }
li { list-style: url(grafika/disk.png) disc }
```

8.6.3 Funkce counter

Counter jsou označeny identifikátory *counter-increment* a *counter-reset*. K použití hodnoty jsou použity *counter(<identifier>)* nebo *counter(<identifier>, <list-style-type>)*. Výchozí tvar je *decimal*.

Pro odkaz na sekvenci vnořených počítadel stejného jména, se používá zápis *counters(<identifier>, <string>)* nebo *counters(<identifier>, <string>, <list-style-type>)*. V CSS2.1 mohou být hodnoty čítače změněny pouze z vlastnosti *content*.

Příklad:

```
p { counter-increment: par-num }
h1 { counter-reset: par-num }
p:before { content: counter(par-num, upper-roman) ". " }
```

Odstavce jsou číslovány velkými římskými čísly. Po každém H1 nadpisu je čítač vynulován.

8.6.4 Funkce calc

Funkce *calc(výraz)* vyhodnotí výraz uvedený v závorce. Může být použita tam, kde se mohou použít délkové hodnoty. Jako operátory je možno použít: +, -, *, /, mod.

Příklad:

```
section {
  float: left;
  margin: 1em; border: solid 1px;
  width: calc(100%/3 - 2*1em - 2*1px);
}
p {
  margin: calc(1rem - 2px) calc(1rem - 1px);
  border: solid transparent; border-width: 2px 1px;
}
p:hover { border-color: yellow; }
```

8.7 Speciální případy

8.7.1 Rodiny písem

font and *font-family* přijímá seznam písem oddělený čárkami. Může to být název písma nebo jeden z pěti generických typů *serif*, *sans-serif*, *cursive*, *fantasy* a *monospace*. Generické typy nesmějí být v uvozovkách.

Příklad:

```
body { font-family: "Helvetica", "Univers", "Arial",
  sans-serif }
```

8.7.2 Colors

Hodnota *color* může být klíčové slovo nebo numerický zápis barev pro systémy RGB, RGBA, HSL, HSLA.

Příklad:

```
em { color: #F00 }
```

9 Barva

Tento modul popisuje vlastnosti CSS, které autorovi umožní specifikovat barvu a průhlednost elementů.

9.1 Vlastnost "color"

- Vlastnost nastavuje barvu textu.
- Vlastnost lze použít na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: `<color>` | `"inherit"` | `attr(<identifier>,color)`.
- Implicitní hodnota: *záleží na prohlížeči*.
- Funkčnost: Firefox 2.0
 Opera 9.2
 Internet Explorer 7.0

9.2 Vlastnost "opacity"

- Vlastnost nastavuje průhlednost barvy
- Vlastnost lze použít na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: `<alphavalue>` (hodnota v intervalu `<0,0;1,0>`, kde 0 je plně průhledná, 1 je plně neprůhledná) | `"inherit"`.
- Implicitní hodnota: *1*.
- Funkčnost: Firefox 2.0
 Opera 9.2

9.3 Vlastnost "color-profile"

- Vlastnost specifikuje zdroj (jiného než základního) barevného profilu.
- Vlastnost lze použít na všechny elementy.

- Vlastnost je dědičná.
- Hodnoty: *"auto"* (všechny barva jsou předdefinovány v RGB barvách, bez obsahu dat. Pro obrázky se nastavuje profil na data toho profilu, který je právě používán) | *"sRGB"* (zdrojový profil je RGB) | *<name>* (jméno barevného profilu, který podporuje prohlížeč. Prohlížeč hledá takový barevný profil, který odpovídá hodnotě v tomto atributu. Jestliže není žádný vhodný nalezen, pak je použit profil vložený uvnitř obrázku.) | *<uri>* (adresa standardního ICC profilu) | *"inherit"*.
- Implicitní hodnota: *auto*.

Příklad:

```
IMG { color-profile: url("http://priklady.com/barvy.icm")
}
```

9.4 Vlastnost "rendering-intent"

- Vlastnost povoluje specifikaci barevného profilu zobrazujícího nestandardní vykreslování. Chování jiných hodnot, než automatických je definována International Color Consorciem (ICC).
- Vlastnost lze použít na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"auto"* (prohlížeč určuje nejlepší záměr, který je založen na typu obsahu) | *"perceptual"* | *"relative-colorimetric"* | *"saturation"* | *"absolute-colorimetric"* | *"inherit"*.
- Implicitní hodnota: *auto*.

9.5 Pravidlo "@color-profile"

SVG 1.0 představuje `@color-profile` jako metodu pro seskupování vlastností *color-profile* a *rendering-intent*.

Toto pravidlo může být použito ke specifikaci popisu barevného profilu. Všeobecný formát je:

```
@color-profile {<color-profile-description>}
/* kde <color-profile-description> má tuto formu: */
deskriptor: hodnota
[...]
deskriptor: hodnota
```

Každé pravidlo `@color-profile` upřesňuje hodnotu pro každý barevný profil deskriptoru, jak pro implicitní, tak i pro explicitní hodnoty.

9.6 Barevné jednotky - `<color>`

9.6.1 HTML4 klíčová slova

Black = #000000	Green = #008000
Silver = #C0C0C0	Lime = #00FF00
Gray = #808080	Olive = #808000
White = #FFFFFF	Yellow = #FFFF00
Maroon = #800000	Navy = #000080
Red = #FF0000	Blue = #0000FF
Purple = #800080	Teal = #008080
Fuchsia = #FF00FF	Aqua = #00FFFF

Příklad:

```
body {color: black; background: white }
h1 { color: maroon }
h2 { color: olive }
```

9.6.2 Numerické hodnoty

9.6.2.1 RGB model

RGB formát je ve tvaru `"#"` ihned následováno třemi nebo šesti hodnotami v šestnáctkové soustavě.

RGB formát je možné vyjádřit také funkcí ve tvaru `rgb (r,g,b)`, kde `r,g,b` jsou hodnoty z intervalu `<0;255>` nebo `<0%;100%>`.

Funkčnost: FireFox 2.0
 Opera 9.2
 Internet Explorer 7.0

Příklad:

```
em { color: #f00 }  
em { color: #ff0000 }  
em { color: rgb(255,0,0) }  
em { color: rgb(100%, 0%, 0%) }
```

Všechny čtyři hodnoty vyjadřují stejnou barvu.

9.6.2.2 RGBA model

RGBA formát je možné vyjádřit také funkcí ve tvaru *rgb(r,g,b,a)*, kde r,g,b jsou hodnoty z intervalu <0;255> nebo <0%;100%>. *a* je reálné číslo z intervalu <0,1>, které dovoluje specifikovat průhlednost barvy.

Funkčnost: FireFox 2.0
 Opera 9.2
 Internet Explorer 7.0

Příklad:

```
em { color: rgb(255,0,0) }  
em { color: rgba(255,0,0,1) }  
em { color: rgb(100%,0%,0%) }  
em { color: rgba(100%,0%,0%,1) }
```

Všechny čtyři hodnoty vyjadřují stejnou barvu.

9.6.2.3 Klíčové slovo "transparent"

CSS1 uvedlo klíčové slovo *transparent* pro vlastnost *background-color*. CSS2 dovoluje použít toto klíčové slovo také pro vlastnost *border-color*. CSS3 dále rozšiřuje použití tohoto klíčového slova na všechny vlastnosti, které akceptují hodnotu *color*. Tímto se zjednoduší definice těchto vlastností v CSS3.

9.6.2.4 HSL model

HSL jsou zakódovány ve tvaru *hsl(jasnost, sytost, odstín)*. Jasnost je vyjádřena jako určitý úhel v barevném kruhu (představme si to jako duhu stočenou v kruhu). Podle definice červené = 360 a ostatní barvy jsou rozprostřeny okolo tohoto kruhu. Tak například zelená = 120, modrá = 240, atd. Sytost a odstín jsou vyjádřena procentuálně. 100% je plná sytost, zatímco 0% je už odstín šedé. 0% odstín je černá, 100% odstín je naopak bílá a 50% odstín je normální.

Funkčnost: FireFox 2.0

Příklad:

```
{ color: hsl(0, 100%, 50%) }
```

červená

```
{ color: hsl(120, 100%, 50%) }
```

zelená

9.6.2.5 HSLA model

K HSL modelu je přidána čtvrtá složka, která vyjadřuje průhlednost.

Příklad:

```
p { color: hsla(240, 100%, 50%, 0.5) }
```

poloprůhledná oranžová

9.6.3 SVG klíčová slova

Seznam barev X11, jsou takové, které jsou podporovány oblíbenými prohlížeči.

Příklad:

```
darkkhaki #bdb76b rgb (189,183,107)
```

```
deeppink #ff1493 rbg (255,20,147)
```

9.6.4 Klíčové slovo "currentColor"

CSS1 a CSS2 definují implicitní hodnotu pro vlastnost *border-color*, ale nemá pro něj odpovídající klíčové slovo. To bylo zjištěno v SVG a proto zavádí klíčové slovo *currentColor*, pro *fill*, *stop-color*, *flood-color*, *lifting-color* vlastnosti. CSS3 rozšiřuje hodnotu *color*, aby zahrnula klíčové slovo *currentColor*, které je možno použít na místo jakékoliv *<color>* hodnoty.

currentColor – vypočtená hodnota klíčového slova je vypočtená hodnota vlastnosti *color*. Pokud je hodnota nastavena na samotnou vlastnost *color*, je s ní při překladu zacházeno jako s hodnotou *inherit*.

9.6.5 CSS Systémové barvy

Systémové barvy dle specifikace CSS2 jsou považovány za překonané ve prospěch vlastnosti CSS3 UI *appearance*, která umožňuje nastavit kompletní vzhled uživatelského rozhraní.

10 Řeč

Zvuková prezentace dokumentu se už mezi slepými běžně používá. Tato hlasová prezentace nabízí možnost převést dokument do čistého textu a následně je obsah přečten pomocí softwarového nebo hardwarového nástroje. Tento způsob ovšem není tak efektivní, jako kdybychom si ponechali strukturu dokumentu. Vlastnosti stylů pro řeč by mohly být použity společně s obrazovými styly nebo jako alternativa k hlasové prezentaci.

10.1 Vlastnost "voice-volume"

- Vlastnost nastavující hlasitost přehrávání zvukového stylu. Při použití zvukových stylů by mělo být dbáno na to, aby alespoň tuto vlastnost mohl uživatel ovlivnit (sám si ji optimálně nastavit), protože u této hodnoty nelze nastavit optimální hranici. Každý audio systém je trochu jiný a každý uživatel má trochu jiné požadavky.
- Vlastnost lze použít na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *<number>* (celé nebo desetinné číslo v intervalu *<0;100>*, kde 0 je ticho a 100 je maximální hlasitost) | *<percentage>* (procenta počítaná vůči zděděné hodnotě jsou oříznuta na interval *<0;100>*) | *"silent"* (0) | *"x-soft"* | *"soft"* | *"medium"* | *"loud"* | *"x-loud"* (100) | *"inherit"*.
- Implicitní hodnota: *medium*.

Příklad:

```
h1: { voice-volume: loud; }
```

10.2 Vlastnost "voice-balance"

- Vlastnost nastavuje poměr mezi levým a pravým zvukovým kanálem.
- Vlastnost lze použít na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *<number>* (celé nebo desetinné číslo v intervalu *<-100;100>*, kde -100 znamená zvuk pouze z levého kanálu, 100 pouze z pravého. Hodnota 0 znamená, že oba kanály hrají stejně hlasitě.) | *"left"* (-100) | *"center"* (0) | *"right"* (100) | *"leftwards"* (od zděděné hodnoty odečte 20, a ořízne na interval *<-100;100>*) | *"rightwards"* (od zděděné hodnoty přičte 20, a ořízne na interval *<-100;100>*) | *"inherit"*.
- Implicitní hodnota: *center*.

Příklad:

```
p.left:    { voice-balance: left; }  
p.right:   { voice-balance: right; }
```

10.3 Vlastnost "speak"

- Vlastnost nastavuje způsob hlasové interpretace textů.
- Vlastnost lze použít na všechny dokumenty.
- Vlastnost je dědičná.
- Hodnoty: *"none"* (zakáže čtení) | *"normal"* (je použita výslovnost pro daný jazyk) | *"spell-out"* (text je hláskován) | *"digits"* (čísla jsou hláskována – tzn., že číslo 18 bude přečteno jako jedna, osm.) | *"literal-punctuation"* (jsou čteny i oddělovače) | *"no-punctuation"* (oddělovače jsou ignorovány i v mluvené formě – tzn., že není generována ani pauza.) | *"inherit"*.
- Implicitní hodnota: *normal*.

Poznámka: speak-volume: silent čte dokument s nulovou hlasitostí, speak: none dokument nečte vůbec.

10.4 Vlastnost "pause-before"

- Vlastnost nastavuje velikost zvukové pomlky před hlasovým překladem elementu.

Pokud je specifikovaná vlastnost *cue-before* nebo *cue-after*, pak platí před touto hodnotou.

- Vlastnost lze použít na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *<time>* (kladná hodnota v s nebo ms) | *"none"* | *"x-weak"* | *"weak"* | *"medium"* | *"strong"* | *"x-strong"* | *"inherit"*.
- Implicitní hodnota: záleží na nastavení prohlížeče.

10.5 Vlastnost "pause-after"

- Vlastnost nastavuje velikost zvukové pomlky za hlasovým překladem elementu.

Pokud je specifikovaná vlastnost *cue-before* nebo *cue-after*, pak platí po této hodnotě.

- Vlastnost lze použít na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *<time>* (kladná hodnota v s nebo ms) | *"none"* | *"x-weak"* | *"weak"* | *"medium"* | *"strong"* | *"x-strong"* | *"inherit"*.
- Implicitní hodnota: záleží na nastavení prohlížeče.

10.6 Vlastnost "pause"

- Vlastnost nastavuje velikost zvukové pomlky před a za hlasovým překladem elementu. Je kombinací *pause-before* a *pause-after*.

Pokud jsou zadány dvě hodnoty, pak první je hodnotou *pause-before* a druhý je hodnotou *pause-after*. Pokud je zadána jedna hodnota, pak je pro použita pro obě vlastnosti.

- Vlastnost lze použít na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: [*<pause-before>* | *<pause-after>*] | "inherit".
- Implicitní hodnota: záleží na nastavení prohlížeče.

Příklad:

```
h1 { pause: 20ms }
p { pause: 30ms 50ms }
h2 { pause-before: 25ms }
```

10.7 Vlastnost "rest-before"

- Vlastnost prozodickou pauzu před elementem.
- Vlastnost lze použít na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *<time>* (kladná hodnota v *s* nebo *ms*) | "none" | "x-weak" | "weak" | "medium" | "strong" | "x-strong" | "inherit".
- Implicitní hodnota: záleží na nastavení prohlížeče.

10.8 Vlastnost "rest-after"

- Vlastnost prozodickou pauzu po elementu.
- Vlastnost lze použít na všechny elementy.
- Vlastnost není dědičná.

- Hodnoty: *<time>* (kladná hodnota v *s* nebo *ms*) | *"none"* | *"x-weak"* | *"weak"* | *"medium"* | *"strong"* | *"x-strong"* | *"inherit"*.
- Implicitní hodnota: záleží na nastavení prohlížeče.

10.9 Vlastnost "rest"

- Vlastnost prozodickou pauzu mezi elementy. Pokud jsou zadány dvě hodnoty, pak první je hodnotou *rest-before* a druhý je hodnotou *rest-after*. Pokud je zadána jedna hodnota, pak je pro použita pro obě vlastnosti.
- Vlastnost lze použít na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: [*<rest-before>* | *<rest-after>*] | *"inherit"*.
- Implicitní hodnota: záleží na nastavení prohlížeče.

10.10 Vlastnost "cue-before"

- Vlastnost nastavuje zvukovou ikonu, která bude spuštěna před hlasovým překladem vztažného elementu.
- Vlastnost lze použít na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *<uri>* (cesta ke zvukové ikoně) | [*<number>* (celé nebo desetinné číslo v intervalu *<0;100>*, kde 0 je ticho a 100 je maximální hlasitost) | *<percentage>* (procenta počítaná vůči zděděné hodnotě *voice-volume* jsou oříznuta na interval *<0;100>*) | *"silent"* | *"x-soft"* | *"soft"* | *"medium"* | *"loud"* | *"x-loud"*] | *"none"* | *"inherit"*.
- Implicitní hodnota: *none*.

10.11 Vlastnost "cue-after"

- Vlastnost nastavuje zvukovou ikonu, která bude spuštěna za hlasovým překladem vztažného elementu.
- Vlastnost lze použít na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: `<uri>` (cesta ke zvukové ikoně) | [`<number>` (celé nebo desetinné číslo v intervalu `<0;100>`, kde 0 je ticho a 100 je maximální hlasitost) | `<percentage>` (procenta počítaná vůči zděděné hodnotě `voice-volume` jsou oříznuta na interval `<0;100>`) | `"silent"` | `"x-soft"` | `"soft"` | `"medium"` | `"loud"` | `"x-loud"`] | `"none"` | `"inherit"`.
- Implicitní hodnota: `none`.

Příklad:

```
a { cue-before: url (bell.aiff); cue-after:
url (dog.wav) }
```

10.12 Vlastnost "cue"

- Vlastnost nastavuje zvukové ikony, která budou spuštěny před a za hlasovým překladem vztažného elementu.
Pokud jsou zadány dvě hodnoty, pak první je hodnotou `cue-before` a druhý je hodnotou `cue-after`. Pokud je zadána jedna hodnota, pak je pro použita pro obě vlastnosti.
- Vlastnost lze použít na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: [`<cue-before>` | `<cue-after>`] | `"inherit"`.
- Implicitní hodnota není definovaná.

Příklad:

```
a { cue: url (bell.aiff) url (dog.wav) }
```

10.13 Vlastnost "mark-before"

- Vlastnost, která nastavuje vkládání pojmenovaných značek před element.
Značky na zvuk nemají žádný vliv.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: `<string>` (jméno značky).
- Implicitní hodnota: *none*.

10.14 Vlastnost "mark-after"

- Vlastnost, která nastavuje vkládání pojmenovaných značek za element.
Značky na zvuk nemají žádný vliv.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: `<string>` (jméno značky).
- Implicitní hodnota: *none*.

10.15 Vlastnost "mark"

- Vlastnost, která nastavuje vkládání pojmenovaných značek před a za element.
Značky na zvuk nemají žádný vliv.
Pokud jsou zadány dvě hodnoty, pak první je hodnotou *mark-before* a druhý je hodnotou *mark-after*. Pokud je zadána jedna hodnota, pak je pro použita pro obě vlastnosti.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: `<mark-before> | <mark-after>`.
- Implicitní hodnota: *none*.

Poznámka: Vlastnost lze vkládat také pomocí funkce attr().

Příklad:

```
div { mark-before: "zacatek"; mark-after: "konec" }  
div { mark: "zacatek" "konec" }
```

10.16 Vlastnost "voice-family"

- Vlastnost nastavuje rodinu hlasů (resp. použitý hlas či skupinu možných použitých hlasů). Tato vlastnost je obdobou výběru správného písma. Pokud tuto vlastnost prohlížeč nepodporuje, nebo není v systému žádné generické zvukové schéma, pak nelze používat zvukový překlad.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: [[<specific-voice> (konkrétní hlas který má být použit k překladu, hlas musí být importován v prohlížeči) | [<age> ("child", "young", "old")] <generic-voice> (stejně jako font-family i této vlastnost obsahuje tzv. generická jména, tedy základní specifikace hlasu – tato vlastnost obsahuje tři hodnoty - "male" – charakterizuje libovolný mužský hlas, "female" – libovolný ženský hlas, "neutral" – neutrální hlas)] [<number> (pořadí další možnosti)],]* [<specific-voice> | [<age>] <generic-voice>] [<number>] | "inherit".
- Implicitní hodnota: záleží na nastavení prohlížeče.

Příklad:

```
h1 {voice-family: announcer, old male}  
p.part.romeo { voice-family: romeo, young male }  
p.part.juliet { voice-family: juliet, female }  
p.part.mercutio { voice-family: male 2 }  
p.part.tybalt { voice-family: male 3 }  
p.part.nurse { voice-family: child female }
```

10.17 Vlastnost "voice-rate"

- Vlastnost nastavuje rychlost hlasového překladu.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *<percentage>* (procentuální rychlost vůči výchozí rychlosti pro danou řeč) | *x-slow* (hodnota odpovídá 80 slovům za minutu) | *"slow"* (hodnota odpovídá 120 slovům za minutu) | *"medium"* (hodnota odpovídá 180 – 200 slovům za minutu) | *"fast"* (hodnota odpovídá 300 slovům za minutu) | *"x-fast"* (hodnota odpovídá 500 slovům za minutu) | *"inherit"*.
- Implicitní hodnota: záleží na nastavení prohlížeče.

Příklad:

```
h1 {speech-rate: fast}
p {speech-rate: 150}
spell {speak: spell-out; speech-rate: x-slow}
```

10.18 Vlastnost "voice-pitch"

- Vlastnost nastavuje průměrnou výšku hlasu. Tato vlastnost závisí na výběru hlasu vlastností *voice-family*.
- Vlastnost lze použít na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *<number>* (kladné celé nebo desetinné číslo vyjadřující frekvenci v Hz) | *<percentage>* (vyjadřuje relativní změnu frekvence vůči zděděnému hlasu) | *"x-low"* | *"low"* | *"medium"* | *"high"* | *"x-high"* | *"inherit"*.
- Implicitní hodnota: *medium*.

10.19 Vlastnost "voice-pitch-range"

- Vlastnost nastavuje rozsah hlasu.
- Vlastnost lze použít na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: `<number>` (kladné celé nebo desetinné číslo vyjadřující rozsah v Hz, nižší hodnota vytvoří monotónní hlas, vyšší bude hlas živější) | `<percentage>` (vyjadřuje relativní změnu frekvence vůči zděděnému hlasu) | `"x-low"` | `"low"` | `"medium"` | `"high"` | `"x-high"` | `"inherit"`.
- Implicitní hodnota: záleží na nastavení prohlížeče.

10.20 Vlastnost "voice-stress"

- Vlastnost, která nastavuje intenzitu důrazu dle národních zvyklostí.
- Vlastnost lze použít na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: `"strong"` (silný) | `"moderate"` (mírný) | `"none"` (žádný) | `"reduced"` (zkracuje sousloví, např. "going to" na "gonna") | `"inherit"`.
- Implicitní hodnota: `moderate`.

10.21 Vlastnost "voice-duration"

- Vlastnost, která specifikuje dobu pro provedení obsahu elementu. Tato vlastnost je nadřazená vlastnosti `voice-rate`.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: `<time>` (kladná hodnota v s nebo ms, která vyjadřuje, kolik času je potřeba pro přečtení obsahu elementu).
- Implicitní hodnota: záleží na nastavení prohlížeče.

10.22 Vlastnost "phonemes"

- Vlastnost, která specifikuje fonetickou výslovnost pro text elementu.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: `<string>`
- Implicitní hodnota: záleží na nastavení prohlížeče.

10.23 @phonetic-alphabet

- Výchozí abeceda pro výslovnost je Internal Phonetic Alphabet – "ipa", která reprezentuje znaky UNICODE.
- Explicitně lze abecedu zadat pomocí pravidla @phonetic-alphabet.

Příklad:

```
@phonetic-alphabet "ipa";  
#tomato { phonemes: "t\0252 m\0251 to\028a " }
```

Syntetizátor řeči nahradí výchozí výslovnost uvedenými znaky.

11 Hypertextové odkazy

Tento modul se zabývá hypertextovými odkazy. Vlastnosti dovolují nastavit cíl odkazu a jeho pozici.

11.1 Vlastnost "target-name"

- Vlastnost nastavuje cíl odkazu.
- Vlastnost lze aplikovat na odkazy.
- Vlastnost není dědičná.
- Hodnoty: *"current"* (aktuální frame, záložka nebo okno) | *"root"* (aktuální záložka nebo okno) | *"parent"* (rodičovský frame) | *"new"* (nový cíl specifikovaný vlastností *target-new*) | *"modal"* (dočasné modální okno) | *<string>* (jméno cílového okna, pokud takové není, pak je vytvořeno).
- Implicitní nastavení: *current*.

11.2 Vlastnost "target-new"

- Vlastnost nastavuje typ, který bude vytvořen.
- Vlastnost lze aplikovat na odkazy.
- Vlastnost není dědičná.
- Hodnoty: *"window"* (nové okno) | *"tab"* (nová záložka) | *"none"* (odkaz nebude zobrazen).
- Implicitní hodnota: *window*.

Příklad: Pokud dá uživatel přednost otvírání odkazů do nového okna před otevřením do nové záložky, pak pomůže následující kód.

```
{ target-new: tab ! important }
```

11.3 Vlastnost "target-position"

- Vlastnost nastavuje pozici nově vytvořeného odkazu.
- Vlastnost lze aplikovat na odkazy.
- Vlastnost není dědičná.
- Hodnoty: *"above"* (cíl odkazu je nad zdrojovou stránkou) | *"behind"* (cíl odkazu je pod zdrojovou stránkou) | *"front"* (cíl odkazu je nad všemi stránkami) | *"back"* (cíl odkazu je pod všemi stránkami).
- Implicitní hodnota: *above*.

Poznámka: Vlastnosti target-position a target-new jsou závislé na hodnotě target-name.

11.4 Vlastnost "target"

- Vlastnost, která nastavuje komplexní vlastnosti pro odkazy.
- Vlastnosti jsou nastaveny na implicitní hodnoty a pak jsou přepsány explicitně zadanými.
- Vlastnost lze aplikovat na odkazy.
- Vlastnost není dědičná.
- Hodnoty: *<target-name>* | *<target-new>* | *<target-position>*
- Implicitní hodnota: *dle jednotlivých vlastností*.

12 Seznamy

Tento modul se zabývá seznamy. Jsou zde uvedeny vlastnosti pro vzhled a umístění odrážek pro číselované i nečíselované seznamy.

12.1 Vlastnost "list-style-type"

- Vlastnost nastavuje výchozí vzhled odrážek a číslování.
- Vlastnost lze použít na všechny elementy, které mají nastaveno *display: list-item*.
- Vlastnost je dědičná.
- Hodnoty:
 - `<glyph>` - jednotný vzhled odrážek
 - `"box"` (čtverec bez výplně) | `"check"` (zaškrtnutí) | `"circle"` (kruh bez výplně) | `"diamond"` (kosočtverec) | `"disc"` (kruh) | `"hyphen"` (pomlčka) | `"square"` (čtverec s výplní).
 - `<algorithmic>` - číslování dle národních zvyklostí
 - `"armenian"` | `"cjk-ideographic"` | `"ethiopic-numeric"` | `"georgian"` | `"hebrew"` | `"japanese-formal"` | `"japanese-informal"` | `"lower-armenian"` | `"lower-roman"` | `"simp-chinese-formal"` | `"simp-chinese-informal"` | `"syriac"` | `"tamil"` | `"trad-chinese-formal"` | `"trad-chinese-informal"` | `"upper-armenian"` | `"upper-roman"`.
 - `<numeric>` - číslování pomocí čísel národních abeced
 - `"arabic-indic"` | `"binary"` | `"bengali"` | `"cambodian"` | `"decimal"` | `"decimal-leading-zero"` | `"devanagari"` | `"gujarati"` | `"gurmukhi"` | `"kannada"` | `"khmer"` | `"lao"` | `"lower-hexadecimal"` | `"malayalam"` | `"mongolian"` | `"myanmar"` |

"octal" | "oriya" | "persian" | "telugu" | "tibetan" | "thai" | "upper-hexadecimal" | "urdu".

- <alphanumeric> - číslování pomocí písmen národních abeced
 - "afar" | "amharic" | "amharic-abegede" | "cjk-earthly-branch" | "cjk-heavenly-stem" | "ethiopic" | "ethiopic-abegede" | "ethiopic-abegede-am-et" | "ethiopic-abegede-gez" | "ethiopic-abegede-ti-et" | "ethiopic-halehame-aa-er" | "ethiopic-halehame-aa-et" | "ethiopic-halehame-am-et" | "ethiopic-halehame-gez" | "ethiopic-halehame-om-et" | "ethiopic-halehame-sid-et" | "ethiopic-halehame-so-et" | "ethiopic-halehame-ti-et" | "ethiopic-halehame-ti-et" | "ethiopic-halehame-tig" | "hangul" | "hangul-consonant" | "hiragana" | "hiragana-iroha" | "katakana" | "katakana-iroha" | "lower-alpha" | "lower-greek" | "lower-norwegian" | "lower-latin" | "oromo" | "sidama" | "somal" | "tigre" | "tigrinya-er" | "tigrinya-er-abegede" | "tigrinya-et" | "tigrinya-et-abegede" | "upper-alpha" | "upper-greek" | "upper-norwegian" | "upper-latin".
- <symbolic> - opakování daného symbolu
 - asteriks (*, **, ***, ...) | footnotes (*, ②, †, ‡, **, ②②, ††, ‡‡, ***, ②②②, †††, ‡‡‡, ****, ②②②②, ...).
- <non-repeating> - číslování využívá speciální symboly omezeného rozsahu. Pokud se dostanete mimo interval, pak je symbol vygenerován prohlížečem nebo se přejde do desítkové soustavy.
 - circled-decimal ("0-20" v kroužku) | circled-lower-latin ("a-z" v kroužku) | circled-upper-latin ("A-Z" v kroužku) | dotted-decimal ("1.-20.") | double-circled-decimal ("(1) – (10)") | filled-circled-decimal ("(11) – (20)") | parenthesised-decimal ("(1) – (20)") | parenthesised-lower-latin ("(a) – (z)").

- *"normal"* (odrážky jsou nastaveny vlastností *@counter*).
- *"none"* (žádná odrážka).
- Implicitní hodnota: *disc*.
- Funkčnost: FireFox 2.0
 Opera 9.2

12.2 Vlastnost **"list-style-image"**

- Vlastnost nastavuje obrázek, který bude použit jako odrážka. Pokud obrázek není k dispozici, je použit druh odrážky definovaný vlastností *list-style-type*.
- Vlastnost lze použít na všechny elementy, které mají nastaveno *display: list-item*.
- Vlastnost je dědičná.
- Hodnoty: *<uri>* | *"none"*.
- Implicitní hodnota: *none*.
- Funkčnost: FireFox 2.0
 Opera 9.2
 Internet Explorer 7.0

Příklad:

```
li { list-style-image: url(grafika/oval.gif) }
```

12.3 Vlastnost **"list-style-position"**

- Vlastnost určuje pozici odrážky seznamu.
- Vlastnost lze použít na všechny elementy, které mají nastaveno *display: list-item*.
- Vlastnost je dědičná.
- Hodnoty: *"inside"* (odrážka umístěna v textu, text se zalamuje k začátku odrážky) | *"outside"* (odrážka umístěna před textem, text se zalamuje k začátku textu prvního řádku).

- Implicitní hodnota: *outside*.
- Funkčnost: FireFox 2.0
Opera 9.2

Příklad:

```
UL { list-style: outside }  
UL.compact { list-style: inside }
```

```
<UL>  
  <LI>toto je první položka</LI>  
  <LI>toto je druhá položka</LI>  
</UL>  
<UL class="compact">  
  <LI>toto je první položka</LI>  
  <LI>toto je druhá položka</LI>  
</UL>
```

Výsledek by měl vypadat asi takto:

- toto je první položka
- toto je druhá položka

- toto je první položka
- toto je druhá položka

12.4 Vlastnost "list-style"

- Vlastnost, která nastavuje komplexní vlastnosti odrážek a číslování.
- Vlastnost lze použít na všechny elementy, které mají nastaveno display: *list-item*.
- Hodnoty: *<list-style-type>* | *<list-style-position>* | *<list-style-image>*.
- Funkčnost: FireFox 2.0
Opera 9.2

Příklad:

```
UL { list-style: upper-roman inside }
UL > UL { list-style: circle outside }
vnořený UL
```

12.5 ::marker pseudoelement

Značky jsou vytvořeny nastavením prvku vlastnosti *display* do položky seznamu. Položky seznamu titulkového písma je v každém druhém respektu, totožném s blokem titulkového písma. Lépe to možná pochopíme pomocí příkladů.

Příklad: Vycentrováný obsah uvnitř odrážek v boxu s pevnou šíří.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE>Content alignment in the marker box</TITLE>
<STYLE type="text/css">
LI::marker {
content: "(" counter(counter) ")";
width: 6em;
text-align: center;
}
LI {
display: list-item;
counter-increment: counter;
}
</STYLE>
</HEAD>
<BODY>
<OL>
<LI> Toto je první položka. </LI>
<LI> Toto je druhá položka. </LI>
<LI> Toto je třetí položka. </LI>
</OL>
</BODY>
</HTML>
```

Výsledek může vypadat nějak takto.

- (1) Toto je první položka.
- (2) Toto je druhá položka.
- (3) Toto je

třetí položka.

Příklad: Následující příklad používá značky k očíslování značek poznámek (odstavců).

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE>Značky k vytvoření poznámek</TITLE>
<STYLE type="text/css">
P { margin-left: 12 em; }
P.note::marker {
content: url("poznamka.gif") "poznamka " counter(note-
counter)
": ";
text-align: left;
width: 10em;
}
P.Note {
display: list-item;
counter-increment: note-counter;
}
</STYLE>
</HEAD>
<BODY>
<P>Toto je první odstavec v našem dokumentu.</P>
<P CLASS="Note">Toto je velmi krátký dokument.</P>
<P>Toto je konec.</P>
</BODY>
</HTML>
```

může zobrazit něco takového.

Toto je první odstavec
v našem dokumentu.
Poznámka 1: Toto je velmi krátký
dokument.
Toto je konec.

Příklad: Následující příklad ilustruje, jak značky mohou být vyrovnány ze svých prvků.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transiti-
onal//EN">
<HTML>
<HEAD>
<TITLE>Marker example 5</TITLE>
<STYLE type="text/css">
```

```
P { margin-left: 8em } /* Vytvoří mezeru pro čítače */
LI::marker { margin: 0 3em 0 0; content: counter(list-
item,
lower-roman) "."; }
LI { display: list-item }
</STYLE>
</HEAD>
<BODY>
<P> Toto je dlouhý předešlý odstavec ...</P>
<OL>
<LI> Toto je první položka
<LI> Toto je druhá položka
<LI> Toto je třetí položka
</OL>
<P> Toto je dlouhý následující odstavec ...</P>
</BODY>
</HTML>
```

může zobrazit třeba takto:

Toto je dlouhý předcházející
odstavec ...
i. Toto je první položka
ii. Toto je druhá položka.
iii. Toto je třetí položka.
Toto je následující
odstavec ...

Poznámka: použití implicitního čítače je inkrementováno.

13 Písma

Tento modul přináší soubor vlastností, které umožňují nové vlastnosti pro úpravu písma jako různé efekty, zdůraznění, vyhlazování, apod.

13.1 Vlastnost "font-family"

- Vlastnost nastavuje rodinu písem, tedy seznam písem, která se mají použít pro zobrazení textu.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: jméno písma či seznam několika písem (dříve napsané má vyšší prioritu). Pokud používáte méně rozšířený typ písma, je vhodné vkládat seznam písem, protože ne vždy je zaručeno, že klient disponuje vámi zvoleným písmem. Z tohoto důvodu se také doporučuje uvést na konec seznamu jedno z tzv. generických jmen, kterými CSS disponuje. Tato generická jména popisují základní skupiny písem, kterými by měl disponovat každý operační systém.
 - *serif*
 - klasické patkové písmo (př. Times New Roman, Garamond)
 - *sans-serif*
 - bezpatkové písmo (př. Verdana, Arial, Helvetica, Tahoma)
 - *cursive*
 - kurzívní písmo (př. Zapf-Chancery, Caflisch Script)
 - *fantasy*
 - ozdobné písmo (př. Western, **Comic Sans MS**)
 - *monospace*
 - neproporcionální písmo (př. Courier)
- Implicitní hodnota je dána přednastavením prohlížeče.
- Funkčnost: Firefox 2.0
Opera 9.2
Internet Explorer 7.0

Příklad:

```
p {color: #000; font-family: Times New Roman CE, serif;}  
div {color: #000; font-family: Arial, sans-serif;}
```

13.2 Vlastnost "font-style"

- Vlastnost nastavuje styl písma (normální, kurzívu či sklonění).
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"normal"* | *"italic"* (písmo je psáno kurzívou) | *"oblique"* (písmo je skloněné, u některých písem stejné jako *"italic"*)
- Implicitní hodnota: *normal*.
- Funkčnost: Firefox 2.0
Opera 9.2
Internet Explorer 7.0

Příklad:

```
p {color: #000; font-family: serif; font-style: oblique}  
div {color: #000; font-style: italic; font-family: arial,  
sans-serif}
```

13.3 Vlastnost "font-variant"

- Vlastnost nastavuje variantu písma (normálně či kapitálky).
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"normal"* | *"small-caps"* (písmo je psané kapitálkami – malá písmena jsou zapsána jako zmenšená velká písmena, někdy nemusí být zmenšená)
- Implicitní hodnota: *normal*.
- Funkčnost: Firefox 2.0
Opera 9.2
Internet Explorer 7.0

Příklad:

```
div {color: #000; font-family: serif;
     font-variant: small-caps}
p   {color: #000; font-variant: normal;
     font-family: arial, sans-serif}
```

13.4 Vlastnost "font-weight"

- Vlastnost nastavuje tzv. duktus (tloušťku) písma.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"normal"* (400) | *"bold"* (700) | *"bolder"* (širší než rodičovské písmo) | *"lighter"* (užší než rodičovské písmo) | *"100"* | *"200"* | *"300"* | *"400"* | *"500"* | *"600"* | *"700"* | *"800"* | *"900"*
- Implicitní hodnota: *normal*.
- Funkčnost: Firefox 2.0
Opera 9.2
Internet Explorer 7.0

Příklad:

```
div {color: #000; font-family: serif;
     font-variant: small-caps;
     font-size: 15pt; font-weight: bold}
p   {color: #000; font-variant: normal; font-size: xx-
large;
font-family: arial, sans-serif; font-weight: 200}
```

13.5 Vlastnost "font-stretch"

- Vlastnost nastavuje šířku písma.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"ultra-condensed"* | *"extra-condensed"* | *"condensed"* | *"semi-condensed"* | *"normal"* | *"semi-expanded"* | *"expanded"* | *"extra-expanded"* | *"ul-*

tra-expanded" | *wider*" (širší než rodičovské písmo) | *narrower*" (užší než rodičovské písmo)

- Implicitní hodnota: *normal*.

Příklad:

```
div {color: #000; font-family: serif;
     font-variant: small-caps;
font-size: 15pt; font-weight: bold}
p   {color: #000; font-variant: normal; font-size: xx-
large;
font-family: arial, sans-serif; font-weight: 200}
```

13.6 Vlastnost "font-size"

- Vlastnost nastavuje velikost písma.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *xx-small*" | *x-small*" | *small*" | *medium*" | *large*" | *x-large*" | *xx-large*" | *larger*" (větší než rodičovské písmo) | *smaller*" (menší než rodičovské písmo) | *<percentage>* (procentuální hodnota, vztahující se k rodičovské velikost, popř. k nastavení prohlížeče) | *<length>* (číselná hodnota s jednotkami).
- Implicitní hodnota: *medium* resp. původní rodičovská hodnota či implicitní hodnota prohlížeče.
- Funkčnost: Firefox 2.0
Opera 9.2
Internet Explorer 7.0

Hodnoty	xx-small	x-small	small	medium	large	x-large	xx-large	
Měřítko	3/5	³ / ₄	8/9	1	6/5	3/2	2/1	3/1
Nadpisy	h6		h5	h4	h3	h2	h1	
font size	1		2	3	4	5	6	7

Tabulka 4 - velikost písma

Příklad:

```
div {color: #000; font-family: serif;
     font-variant: small-caps;
font-size: 50px}
```

```
p {color: #000; font-variant: normal;
  font-size: xx-large;
font-family: arial, sans-serif}
```

13.7 Vlastnost "font-size-adjust"

- Vlastnost upravuje velikost písma.
Subjektivní velikost písma je vyjádřena poměrnou hodnotou zvanou *aspect value* a její hodnota je rovna poměru výšky-x a velikosti písma.
Aspect value = x-height/font-size
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: `<number>` | `"none"`
 - `<number>` je hodnota *a*, přičemž pro změnu měřítka platí: $c = y \cdot (a/a')$, kde
 - *y* = font-size
 - *a* = aspect value specifikována hodnotou font-size-adjust
 - *a'* = aspect value aktuálního fontu
 - *c* = výsledný font-size
- Implicitní hodnota: *none*.

Příklad:

Pokud není možné použít Verdanu s font-size: 14px (aspect value = 0.58) a dostupný font má aspect value 0.46, pak bude výsledná velikost přepočítána:

$14 \cdot (0.58/0.46) = 17.65\text{px}$.

Písmo	aspect value	font-size-adjust
Verdana	0.58	1
Comic Sans MS	0.54	1.07
Trebuchet MS	0.53	1.09
Georgia	0.5	1.16
Myriad Web	0.48	1.2
Minion Web	0.47	1.23
Times New Roman	0.46	1.26
Gill Sans	0.46	1.26
Bernhard Modern	0.4	1.45
Caflich Script Web	0.37	1.57
Flemish Script	0.28	2.07

Tabulka 5 - Font-size-adjust pro font Verdana

13.8 Vlastnost "font"

- Parametr pro komplexní nastavení vlastností písma. Při použití jsou všechny vlastnosti nastaveny na implicitní hodnoty.

Vlastnosti nastavujeme v pořadí: *font-style*, *font-variant*, *font-weight*, *font-size*, *line-height*, *font-family*.

Kterákoliv položka může být vypuštěna.

Atributy *font-stretch* a *font-size-adjust* se vzhledem ke zpětné kompatibilitě musí zadávat zvlášť.

- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty pro nastavení vlastností jsou totožné s hodnotami daných vlastností.
- Implicitní hodnota: *shodná s předchozími popisy*.
- Funkčnost: FireFox 2.0

 Opera 9.2

 Internet Explorer 7.0

Příklad:

```
div { font: italic small-caps bold 15pt serif}
p   { font: 200 xx-large "Arial CE", sans-serif}
p   { font: bold italic large Palatino, serif }
p   { font: 600 9pt Charcoal }
```

je stejné jako

```
p   { font-style: normal; font-variant: normal;
font-weight: 600; font-size: 9pt;
line-height: normal; font-family: Charcoal }
```

13.9 Vlastnost "font-effect"

- Vlastnost, která písmu nastaví speciální efekt.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"none"* | *"emboss"* (reliéf) | *"engrave"* (rytina) | *"outline"* (vnější obrys).
- Implicitní hodnota: *none*.

Příklad:

```
h2#relief { font-effect: emboss }
```

13.10 Vlastnost "font-smooth"

- Vlastnost, která umožní při vykreslování použít anti-aliasing.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"auto"* (závisí na prohlížeči) | *"never"* | *"always"* | *<absolute-size>* | *<length>* (Pokud je aktuální hodnota *font-size* větší nebo rovna než specifikovaná hodnota, pak je použito vyhlazování při vykreslení textu).
- Implicitní hodnota: *auto*.

Příklad:

```
body { font-smooth: always }
```

13.11 Vlastnost "font-emphasize-style"

- Vlastnost, která nastaví zvýraznění textu. Východoasijské dokumenty používají tyto symboly nad každým znakem, aby zdůraznili směr textu.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"none"* | *"accent"* | *"dot"* | *"circle"* | *"disc"*.
- Implicitní hodnota: *none*.

13.12 Vlastnost "font-emphasize-position"

- Vlastnost, která nastaví pozici zvýrazňujících symbolů.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"before"* (nad textem) | *"after"* (vlevo od sloupce).
- Implicitní hodnota: *before*.

Jazyk	Preferovaná pozice zvýraznění
Japanese	before
Chinese (Traditional)	before
Chinese (Simplified)	after

Tabulka 6 - hodnoty font-emphasize-position pro asijské jazyky

13.13 Vlastnost "font-emphasize"

- Parametr pro komplexní nastavení vlastností zvýraznění
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *<font-emphasize-style>* | *<font-emphasize-position>*.
- Implicitní hodnota: *shodná s předchozími popisy*.

14 Písma pro web

V CSS1 bylo vyžadováno, aby byla všechna písma nainstalována na klientově počítači a byla identifikována pouze jmény. V CSS2 přišla změna. Umožňuje autorovi popsat font, který chce použít. Pokud písmo uživatel nemá, pak ho prohlížeč stáhne z internetu. A právě o to se starají "písma pro web".

14.1 Výběr písma

Výběr písma je prováděn dle následujících pravidel:

- *font name matching* – V tomto případě použije prohlížeč existující font, který je ze stejné rodiny písem jako zadané písmo.
- *intelligent font matching* – V tomto případě použije prohlížeč existující font, který je zadanému písmu nejvíce podobný.
- *font synthesis* – V tomto případě prohlížeč nejpodobnější písmo ještě upraví, aby výsledné zobrazení co nejvíce odpovídalo vlastnostem původně požadovaného fontu.
- *font download* – V tomto případě je písmo staženo z internetu.
- *progressive rendering* – Toto je kombinace stažení písma a některé z předchozích metod.

14.2 @font-face

Pomocí tohoto parametru definujeme vlastnosti fontu, dle kterých je při "intelligentním hledání" vybírán podobný font popřípadě, při "metodě syntézy" i upravován náhradní font. V tomto parametru taktéž definujeme umístění tohoto fontu na Internetu pro načtení písma.

Parametr má syntaxi:

```
@font-face { <font-description> }
```

<font-description> je ve tvaru:

```
descriptor: value;  
descriptor: value;  
[...]  
descriptor: value;
```

14.3 Deskriptory pro výběr písma - font-family, font-style, font-variant, font-weight, font-stretch a font-size

Všechny tyto vlastnosti mají stejná jména jako klasické v modulu FONT a mají i stejné hodnoty.

14.4 Deskriptory pro kvalifikaci dat fontu: unicode-range

Vlastnost popisuje hodnoty variant Unicode, které má obsahovat písmo hledané pro shodu s původním fontem. Používá se v kontextu *@font-face*.

Implicitní hodnota: *U+0-7FFFFFFF*.

14.5 Deskriptor pro numerické hodnoty: units-per-em

Vlastnost popisuje délkovou jednotku písma pro použití v dalších parametrech, konkrétně sděluje, kolik jednotek se vejde do čtverčiku em. Používá se v kontextu *@font-face* a hodnotou je číslice bez jednotek.

14.6 Deskriptor pro odkazy: src

Odkaz (či seznam odkazů) na soubor obsahující definici požadovaného písma. Vlastnost lze použít v kontextu *@font-face*. Hodnoty: *<uri>* (odkaz na soubor písma), *format(<string>)* (specifikace druhu písem a platformy).

Příklad:

```
src: url("../fonts/bar") format("truedoc-pfr")
```

14.7 Deskriptor pro párování: *panose-1*, *stemv*, *stemh*, *slope*, *cap-height*, *x-height*, *ascent*, *descent*

Vlastnost *panose-1*, popisuje požadovaný typ písma podle standardu Panose, dnes zřejmě nepoužívanějšího standardu pro písma typu latinka.

Vlastnost *stemv* popisuje výšku písma.

Vlastnost *stemh* popisuje šířku písma.

Vlastnost *slope*, popisuje základní úhel sklonu písma od svislé osy.

Vlastnost *cap-height*, popisuje výšku velkých písmen u původního písma.

Vlastnost *x-height*, popisuje výšku malých písmen u původního písma.

Vlastnost *ascent*, popisuje maximální výšku znaku bez diakritiky.

Vlastnost *descent*, popisuje maximální výšku písma pod základnou, bez diakritiky. Některé jazyky totiž používají diakritiku i pod písmenem např. Arabština.

14.8 Deskriptor pro syntézu

Vlastnost *widths*, určuje šířku znaků, hodnotou je seznam hodnot *range* oddělených čárkou, který je následován jednou nebo více číselnými hodnotami. Jestliže použijeme tento deskriptor, musíme použít i deskriptor *units-per-em*, viz výše.

Vlastnost *bbox*, určuje maximální ohraničení boxů, fontu. Hodnotou jsou čtyři čísla, v pořadí: levá x-ová, levá y-ová, pravá x-ová a pravá y-ová, ohraničujícího boxu.

Vlastnost *definition-src*, může být buď uvnitř fontu jako popis ve formátovací sadě, nebo může být poskytnutý uvnitř odděleného fontu definice zdroje definovaného pomocí URI. Pozdější přístup může snížit zatížení sítě, pokud se více šablon odkazuje na stejné fonty.

14.9 Deskriptory na zarovnání: *baseline*, *centerline*, *mathline*, *topline*

Deskriptor *baseline*, je pro použití pro všechny základny fontů. A jestliže je tento nastaven na jinou hodnotu, než implicitní (nula), musí být použit také deskriptor *units-per-em*.

Deskriptor *centerline*, tento je použit pro použití střední základny písma. Jestliže není nadefinována žádná hodnota, prohlížeč může použít různou heuristiku, tak jako střed úsečky výstupu a sestupu hodnoty.

Deskriptor *mathline*, se používá pro základnu u matematických výrazů.

Deskriptor *topline*, se používá pro nejvyšší základnu písma.

Pro lepší porozumění výše uvedených vlastností, je potřeba znát typografické výrazy a základy profesionální sazby.

Příklad:

```
Swiss 721 medium      medium & medium italic
@font-face {
  font-family: "Swiss 721";
  src: url("swiss721md.pfr"); /* Swiss 721 medium */
  font-style: normal, italic;
  font-weight: 500; }
```

15 Text

Tento modul definuje vlastnosti pro práci s textem. Zabývá se například zalomením řádků, zarovnáním, prací s bílými znaky, dekorací a převodem textu.

15.1 Vlastnost "text-wrap"

- Vlastnost, která nastavuje zalomení textu.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"normal"* (zalomení na povolených místech) | *"unrestricted"* (text může být zalomen mezi dvěma) | *"none"* (text nebude zalomen) | *"suppress"* (zalomení textu je zakázáno).
- Implicitní hodnota: *normal*.

15.2 Vlastnost "white-space-collapse"

- Vlastnost nastavuje, jak bude zacházeno s mezerami uvnitř elementu.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"preserve"* (budou zobrazeny všechny mezery i přechody na nový řádek) | *"collapse"* (více po sobě jdoucích mezer bude reprezentováno jako jedna) | *"preserve-breaks"* (více po sobě jdoucích mezer bude reprezentováno jako jedna, přechody na nový řádek budou zobrazeny) | *"discard"* (všechny bílé znaky budou odstraněny).
- Implicitní hodnota: *collapse*.

15.3 Vlastnost "white-space"

- Vlastnost, která je kombinací *white-space-collapse* a *text-wrap*.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"normal"* | *"pre"* | *"nowrap"* | *"pre-wrap"* | *"pre-line"*.
- Implicitní hodnota: *normal*.
- Funkčnost: FireFox 2.0
 Opera 9.2
 Internet Explorer 7.0

vlastnost	nové řádky	mezery a tabulátory	zalomení
normal	pohlčení	pohlčení	zalomení
pre	zachovává	zachovává	nezalomení
nowrap	pohlčení	pohlčení	nezalomení
pre-wrap	zachovává	zachovává	zalomení
pre-line	zachovává	pohlčení	zalomení

Tabulka 7 - hodnoty vlastností white-space

15.4 Vlastnost "word-wrap"

- Vlastnost, která nastavuje, jak může prohlížeč dělit slova. Funguje to pouze za předpokladu, že je zalamování textu nastaveno na *normal* nebo *suppress*.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"normal"* (zalomení na povolených místech) | *"break-word"* (nedělitelné slovo může být zalomeno na jakémkoli místě, pokud v řádce není jiné vhodnější místo).
- Implicitní hodnota: *normal*.

15.5 Vlastnost "word-break"

- Vlastnost nastavuje, jak budou zalamována slova v elementu.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"normal"* | *"keep-all"* | *"loose"* | *"break-strict"* | *"break-all"*.
- Implicitní hodnota: *normal*.

15.6 Vlastnost "text-align"

- Vlastnost nastavuje především odstavcové zarovnání textu (k levému či pravému okraji, na střed nebo do bloku).
- Vlastnost lze aplikovat pouze na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"start"* (obsah je zarovnán se začátkem bloku) | *"end"* (obsah je zarovnán s koncem bloku) | *"left"* (text zarovnán k levému okraji) | *"center"* (text zarovnán na střed) | *"right"* (text zarovnán k pravému okraji) | *"justify"* (text je zarovnán do bloku pomocí zvětšení mezislovních mezer) | *<string>* (specifikuje řetězec, ve které buňce bude řetězec zarovnán).
- Implicitní hodnota: *start*.
- Funkčnost: FireFox 2.0
 Opera 9.2
 Internet Explorer 7.0

*Poznámka: Pro text, který je zleva-doprava je hodnota "start" rovna hodnotě "left".
Pro text, který je zprava-doleva je hodnota "start" rovna hodnotě "right".*

15.7 Vlastnost "text-align-last"

- Vlastnost nastavuje odstavcové zarovnání textu posledního řádku pro blok, který má *text-align: justify*.
- Vlastnost lze aplikovat pouze na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: "start" (obsah je zarovnán se začátkem bloku) | "end" (obsah je zarovnán s koncem bloku) | "left" (text zarovnán k levému okraji) | "center" (text zarovnán na střed) | "right" (text zarovnán k pravému okraji) | "justify" (text je zarovnán do bloku pomocí zvětšení mezislovních mezer)
- Implicitní hodnota: *start*.
- Funkčnost: Internet Explorer 7.0

15.8 Vlastnost "text-justify"

- Vlastnost nastavuje, jaká metoda bude použita pro vlastnost *text-align: justify*.
- Vlastnost lze aplikovat pouze na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: "auto" (algoritmus volí prohlížeč) | "inter-word" (mění se především mezery u oddělovačů) | "inter-ideograph" | "inter-cluster" | "distribute" | "kashida" | "tibetan".
- Implicitní hodnota: *auto*.
- Funkčnost: Internet Explorer 7.0

Poznámka: V další verzi dokumentu bude nejspíš odebrána hodnota "tibetan".

15.9 Vlastnost "word-spacing"

- Vlastnost nastavuje velikost mezislovní mezery v textu.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: "normal" | *<length>* {1,3}.

- Vlastnost přijímá 1-3 hodnoty - minimum, optimum, maximum. Pokud je použita jedna hodnota, pak vyjadřuje optimální hodnotu, minimum a maximum jsou *normal*. Pokud jsou použity dvě hodnoty, pak první vyjadřuje optimální a minimální hodnotu a druhý vyjadřuje maximum. Pokud jsou použity tři hodnoty, pak reprezentují postupně optimum, minimum a maximum.
- Implicitní hodnota: *normal*.
- Funkčnost: FireFox 2.0
 Opera 9.2
 Internet Explorer 7.0

Příklad:

```
h1 { font: small-caps 15pt serif;  
      word-spacing: 15px }  
p   { word-spacing: normal }
```

15.10 Vlastnost "letter-spacing"

- Vlastnost nastavuje velikost mezery mezi jednotlivými písmeny ve slově.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: "*normal*" | *<length>* {1,3}.

Vlastnost přijímá 1-3 hodnoty. Minimum, optimum, maximum. Pokud je použita jedna hodnota, pak vyjadřuje optimální hodnotu, minimum a maximum jsou *normal*. Pokud jsou použity dvě hodnoty, pak první vyjadřuje optimální a minimální hodnotu a druhý vyjadřuje maximum. Pokud jsou použity tři hodnoty, pak reprezentují postupně optimum, minimum a maximum.

- Implicitní hodnota: *normal*.
- Funkčnost: FireFox 2.0
 Opera 9.2
 Internet Explorer 7.0

Poznámka: Při použití této vlastnosti musíte počítat se zvětšením mezislovních mezer.

Příklad:

```
h1 { font: small-caps 15pt serif;
      word-spacing: 15px}
p   { word-spacing: normal;
      letter-spacing: 1mm }
```

15.11 Vlastnost "punctuation-trim"

- Vlastnost nastavuje, jak mají být oříznuty znaky, které vyplňují pouze část boxu pro znak.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"none"* | *"start"* (ořízne prázdnou část otvíracích závorek na počátku každé řády) | *"end"* (ořízne prázdnou část uzavíracích závorek na konci každé řády) | *"adjacent."*
- Implicitní hodnota: *none*.

Demonstration of 'adjacent' punctuation trimming

Combination	Sample Pair	Looks Like
Opening—Opening	[+ ([(
Middle Dot—Opening	. + (.(
Closing—Opening] + (] (
Ideographic Space—Opening	+ ([(
Closing—Closing) +])]
Closing—Middle Dot) + .) .
Closing—Ideographic Space) +)

Obrázek 1 - zobrazení punctuation-trim: adjacent

15.12 Vlastnost "text-line-decoration"

- Vlastnost nastavuje pozici umístění čáry pro zvýraznění.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *"none"* (bez dekorace) | *"underline"* (text bude podtržený plnou vodorovnou čárou) | *"overline"* (čára bude nad textem) | *"line-through"* (text bude přeškrtnut).
- Implicitní hodnota: *none*.

15.13 Vlastnost "text-line-color"

- Vlastnost nastavuje barvu čáry pro zvýraznění.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *<color>*.
- Implicitní hodnota: *currentcolor*.

15.14 Vlastnost "text-line-style"

- Vlastnost nastavuje styl čáry pro zvýraznění.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"solid"* (plná čára) | *"double"* (plná dvojitá čára) | *"dotted"* (tečkovaná čára) | *"dashed"* (čárkovaná čára) | *"dot-dash"* (čerchovaná čára) | *"dot-dot-dash"* (čerchovaná čára s dvěma tečkami) | *"wave"* (vlnka) | *"thick"* (tlustá plná čára).
- Implicitní hodnota: *solid*.

15.15 Vlastnost "text-line-skip"

- Vlastnost nastavuje styl ohraničení písma.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"none"* | *"images"* | *"spaces"* | *"ink"*.
- Implicitní hodnota: *images*.

15.16 Vlastnost "text-decoration"

- Parametr pro komplexní nastavení dekorace písma.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *<text-line-decoration>* | *<text-line-color>* | *<text-line-style>* | *"blink"* (text bude blikat).
- Implicitní hodnota: shodná s předchozími popisy.

15.17 Vlastnost "text-underline-position"

- Vlastnost nastavuje umístění podtržení při nastaveném podtržení.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"auto"* | *"before-edge"* (zarovnání před pozicí čáry určené podle okrajové čáry v boxu) | *"alphabetic"* (pod textem) | *"after-edge"* (za pozicí čáry určené podle okrajové čáry v boxu)
- Implicitní hodnota: *auto*.

15.18 Vlastnost "text-emphasis"

- Vlastnost nastavuje východním jazykům zvýraznění, které zvýrazní směr toku textu.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"none"* | *"accent"* (čárkou) | *"dot"* (tečkou) | *"circle"* (kroužkem) | *"disc"* (kolečkem) | *"before"* (vpravo nahoře) | *"after"* (vlevo dole).
- Implicitní hodnota: *none*.

15.19 Vlastnost "text-shadow"

- Vlastnost nastavující u textu trojrozměrný vzhled pomocí stínování.
- Vlastnost je dědičná.
- Hodnoty: *"none"* | *<shadow>*.
<shadow> je reprezentován jako *<color> <length>* (posun v ose x) *<length>* (posun v ose y) *<length>* (volitelný poloměr rozmazání).
Vlastnost může obsahovat více hodnot, oddělených čárkami.
- Implicitní hodnota: *none*.

Příklad:

```
h1 {text-shadow: 10px 2px yellow}
```

15.20 Vlastnost "text-outline"

- Vlastnost nastavující obrys textu. Podobná vlastnosti text-shadow.
- Vlastnost je dědičná.
- Hodnoty: *"none"* | *<color> <length>* (tloušťka čáry) *<length>* (volitelný poloměr rozmazání).
Posunutí splňuje rovnici $x^2 + y^2 = \text{tloušťka}^2$.
- Implicitní hodnota: *none*.

15.21 Vlastnost "text-indent"

- Vlastnost nastavuje velikost odstavcové odrážky (o kolik bude odsazen první řádek v odstavci).
- Vlastnost lze aplikovat na blokové elementy a buňky tabulky.
- Vlastnost je dědičná.
- Hodnoty: `<length>` | `<percentage>`. | `"hanging"` (odsazeny všechny řádky, kromě první).
- Implicitní hodnota: `0`.
Odsazení je aplikováno pouze na blokové elementy, které obsahují více než jeden řádek.
- Funkčnost: FireFox 2.0
 Opera 9.2
 Internet Explorer 7.0

Příklad:

```
p { word-spacing: normal;
    letter-spacing: 1mm;
    text-align: justify; text-indent: 4em}
```

15.22 Vlastnost "hanging -punctuation"

- Vlastnost nastavuje, kde bude umístěna interpunkční značka.
- Vlastnost lze aplikovat na blokové elementy a buňky tabulky.
- Vlastnost je dědičná.
- Hodnoty: `"none"` | `"start"` (znaménko bude na začátku textu, ke kterému znaménko patří) | `"end"` (znaménko bude na konci textu, ke kterému znaménko patří) | `"end-edge"` (může přesahovat přes kteroukoli řádku).
- Implicitní hodnota: `none`.

Poznámka: Odsazení je aplikováno pouze na blokové elementy, které obsahují více než jeden řádek.

15.23 Vlastnost "text-transform"

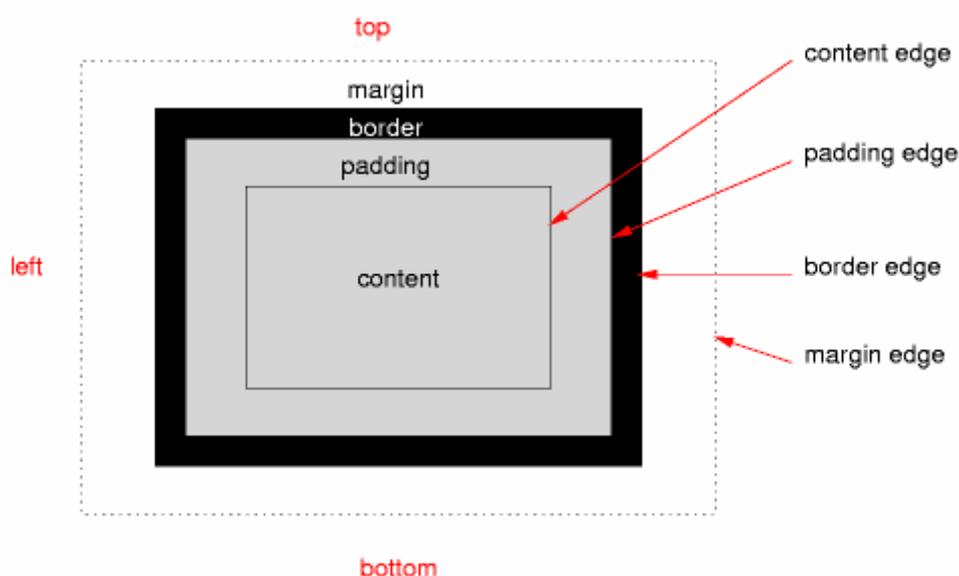
- Vlastnost převede text na daný formát (kapitálky a velká nebo malá písmena).
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"none"* (bez transformace) | *"capitalize"* (převod textu na kapitálky) | *"uppercase"* (transformace na velká písmena) | *"lowercase"* (převod na malá písmena)
- Implicitní hodnota: *none*.
- Funkčnost: FireFox 2.0
 Opera 9.2
 Internet Explorer 7.0

Příklad:

```
h1 { font: small-caps 15pt serif;  
      word-spacing: 15px;  
      text-transform: capitalize}
```

16 Box model

Formátovací model CSS popisuje obdélníkové rámy, které klient generuje pro prvky dokumentu. Základem každého rámu je oblast obsahu (text, obrázek atd.). Kolem ní mohou být volitelně další oblasti: výplňová oblast (padding), oblast rámečku (border) a oblast okraje (margin). Jejich rozměry popisují příslušné vlastnosti CSS a také odpovídající typ formátování. Pro rozměry všech oblastí kolem obsahu, existují samostatné vlastnosti, také jejich rozměr je možné určit na každé straně jiný.



Obrázek 2 - popis box modelu

16.1 Vlastnost "display-model"

- Vlastnost nastavuje uspořádání vnořených elementů.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *"inline-inside"* (uvnitř řádky) | *"block-inside"* (uvnitř bloku) | *"table"* | *"ruby"*.
- Implicitní hodnota: *text*.

16.2 Vlastnost "display-role"

- Vlastnost nastavuje, jakou funkci mají prvky v rodičovských elementech.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *"none"* (element není vykreslen)| *"block"* (element je vykreslen jako blok) | *"inline"* (element je vykreslen uvnitř řádkového boxu) | *"list-item"* (*"block"*, na jehož začátek je přidána odrážka)| *"run-in"* (tento efekt záleží na tom, jaký bude následovat prvek.) | *"compact"* (kompaktní) | *"table-row"* (řádka tabulky) | *"table-cell"* (buňka tabulky) | *"table-row-group"* (skupina řádek tabulky) | *"table-header-group"* (záhlaví tabulky) | *"table-footer-group"* (zápatí tabulky) | *"table-column"* (sloupec tabulky) | *"table-column-group"* (skupina sloupců tabulky) | *"table-caption"* (titulek tabulky) | *"ruby-text"* | *"ruby-base"* | *"ruby-base-group"* | *"ruby-text-group"*.
- Implicitní hodnota: *inline*.

16.3 Vlastnost "display"

- Vlastnost komplexně nastavuje vlastnosti zobrazení.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *"inline"* | *"block"* | *"inline-block"* | *"list-item"* | *"run-in"* | *"compact"* | *"table"* | *"inline-table"* | *"table-row-group"* | *"table-header-group"* | *"table-footer-group"* | *"table-row"* | *"table-column-group"* | *"table-column"* | *"table-cell"* | *"table-caption"* | *"ruby"* | *"ruby-base"* | *"ruby-text"* | *"ruby-base-group"* | *"ruby-text-group"* | *"none"*.
- Implicitní hodnota: *inline*.

16.4 Vlastnost "padding-top"

- Vlastnost nastavuje velikost horního vnitřního okraje.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *<length>* | *<percentage>*.
- Implicitní hodnota: *0*.

16.5 Vlastnost "padding-right"

- Vlastnost nastavuje velikost pravého vnitřního okraje.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: číselná *<length>* | *<percentage>*.
- Implicitní hodnota: *0*.

16.6 Vlastnost "padding-bottom"

- Vlastnost nastavuje velikost spodního vnitřního okraje.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *<length>* | *<percentage>*.
- Implicitní hodnota: *0*.

16.7 Vlastnost "padding-left"

- Vlastnost nastavuje velikost levého vnitřního okraje.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: číselná *<length>* | *<percentage>*.
- Implicitní hodnota: *0*.

16.8 Vlastnost "padding"

- Vlastnost komplexně nastavuje vnitřní okraje bloku.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: Tato vlastnost má buď čtyři, tři, dva nebo jeden parametr. V případě, že má parametry čtyři, nastavujeme okraje v pořadí nahoře, vpravo, dole, vlevo. V případě tří parametrů nastavujeme v pořadí nahoře, vpravo a zároveň vlevo, dole. V případě dvou parametrů nahoře a zároveň dole, vlevo a zároveň vpravo. A v případě jednoho parametru jsou všechny okraje stejné. Hodnoty parametrů jsou shodné s předchozími vlastnostmi.
- Implicitní hodnota: *0*.
- Funkčnost: FireFox 2.0
 Opera 9.2
 Internet Explorer 7.0

16.9 Vlastnost "margin-top"

- Vlastnost nastavuje velikost horního okraje.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *<length>* | *<percentage>* | *"auto"*.
- Implicitní hodnota: *0*.

16.10 Vlastnost "margin-right"

- Vlastnost nastavuje velikost pravého okraje.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *<length>* | *<percentage>* | *"auto"*.
- Implicitní hodnota: *0*.

16.11 Vlastnost "margin-bottom"

- Vlastnost nastavuje velikost spodního okraje.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *<length>* | *<percentage>* | *"auto"*.
- Implicitní hodnota: *0*.

16.12 Vlastnost "margin-left"

- Vlastnost nastavuje velikost levého okraje.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *<length>* | *<percentage>* | *"auto"*.
- Implicitní hodnota: *0*.

16.13 Vlastnost "margin"

- Vlastnost komplexně nastavuje okraje kolem bloku.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: Tato vlastnost má buď čtyři, tři, dva nebo jeden parametr. V případě, že má parametry čtyři, nastavujeme okraje v pořadí nahoře, vpravo, dole, vlevo. V případě tří parametrů nastavujeme v pořadí nahoře, vpravo a zároveň vlevo, dole. V případě dvou parametrů nahoře a zároveň dole, vlevo a zároveň vpravo. A v případě jednoho parametru jsou všechny okraje stejné. Hodnoty parametrů jsou shodné s předchozími vlastnostmi.
- Implicitní hodnota: *0*.
- Funkčnost: FireFox 2.0
 Opera 9.2
 Internet Explorer 7.0

16.14 Vlastnost "width"

- Vlastnost nastavuje šířku elementu.
- Vlastnost lze aplikovat na všechny elementy, kromě inline-elementů s vlastností *display-model: inline-inside*.
- Vlastnost není dědičná.
- Hodnoty: *<length>* | *<percentage>* (relativní změna vůči rodičovskému bloku) | *<number>* (nastavuje velikost vůči šířce vnitřního rámečku) | "auto".
- Implicitní hodnota: *auto*.
- Funkčnost: Firefox 2.0
Opera 9.2
Internet Explorer 7.0

Poznámka: width of containing block = margin-left + border-left + padding-left + width + padding-right + border-right + margin-right.

16.15 Vlastnost "height"

- Vlastnost nastavuje výšku elementu
- Vlastnost lze aplikovat na všechny elementy, kromě inline-elementů s vlastností *display-model: inline-inside*.
- Vlastnost není dědičná.
- Hodnoty: *<length>* | *<percentage>* (relativní změna vůči rodičovskému bloku) | *<number>* (nastavuje velikost vůči výšce vnitřního rámečku) | "auto".
- Implicitní hodnota: *auto*.
- Funkčnost: Firefox 2.0
Opera 9.2
Internet Explorer 7.0

16.16 Vlastnost "box-width"

- Vlastnost nastavuje šířku elementu boxu elementu.
- Vlastnost lze aplikovat na všechny elementy, kromě inline-elementů s vlastností *display-model: inline-inside*.
- Vlastnost není dědičná.
- Hodnoty: *<length>* | *<percentage>* (relativní změna vůči rodičovskému bloku) | *<number>* (nastavuje velikost vůči šířce vnitřního rámečku) | *"auto"*.
- Implicitní hodnota: *auto*.

16.17 Vlastnost "box-height"

- Vlastnost nastavuje výšku elementu boxu elementu.
- Vlastnost lze aplikovat na všechny elementy, kromě inline-elementů s vlastností *display-model: inline-inside*.
- Vlastnost není dědičná.
- Hodnoty: *<length>* | *<percentage>* (relativní změna vůči rodičovskému bloku) | *<number>* (nastavuje velikost vůči výšce vnitřního rámečku) | *"auto"*.
- Implicitní hodnota: *auto*.

16.18 Vlastnost "box-sizing"

- Vlastnost řeší problém s různou interpretací vlastnosti *width*.
- Vlastnost lze aplikovat na všechny elementy, kromě inline-elementů s vlastností *display-model: inline-inside*.
- Vlastnost není dědičná.
- Hodnoty: *"content-box"* (nastavení výšky a šířky boxu s obsahem) | *"border-box"* (nastavení výšky a šířky vnějších okrajů boxu).
- Implicitní hodnota: *content-box*.

Poznámka: Vlastnost bude nejspíše nahrazena vlastnostmi box-width, box-height.

16.19 Vlastnost "min-width"

- Vlastnost nastavuje minimální hodnotu šířky obsahu elementu.
- Vlastnost lze aplikovat na všechny elementy, kromě inline-elementů s vlastností *display-model: inline-inside*.
- Vlastnost není dědičná.
- Hodnoty: *<length>* | *<percentage>*.
- Implicitní hodnota: *0*.
- Funkčnost: FireFox 2.0
 Opera 9.2
 Internet Explorer 7.0

16.20 Vlastnost "min-height"

- Vlastnost nastavuje minimální hodnotu výšky obsahu elementu.
- Vlastnost lze aplikovat na všechny elementy, kromě inline-elementů s vlastností *display-model: inline-inside*.
- Vlastnost není dědičná.
- Hodnoty: *<length>* | *<percentage>*.
- Implicitní hodnota: *0*.
- Funkčnost: FireFox 2.0
 Opera 9.2
 Internet Explorer 7.0

16.21 Vlastnost "max-width"

- Vlastnost nastavuje maximální hodnotu šířky elementu.
- Vlastnost lze aplikovat na všechny elementy, kromě inline-elementů s vlastností *display-model: inline-inside*.
- Vlastnost není dědičná.
- Hodnoty: *<length>* | *<percentage>* | "auto" | "inherit".
- Implicitní hodnota: *auto*.
- Funkčnost: FireFox 2.0
Opera 9.2
Internet Explorer 7.0

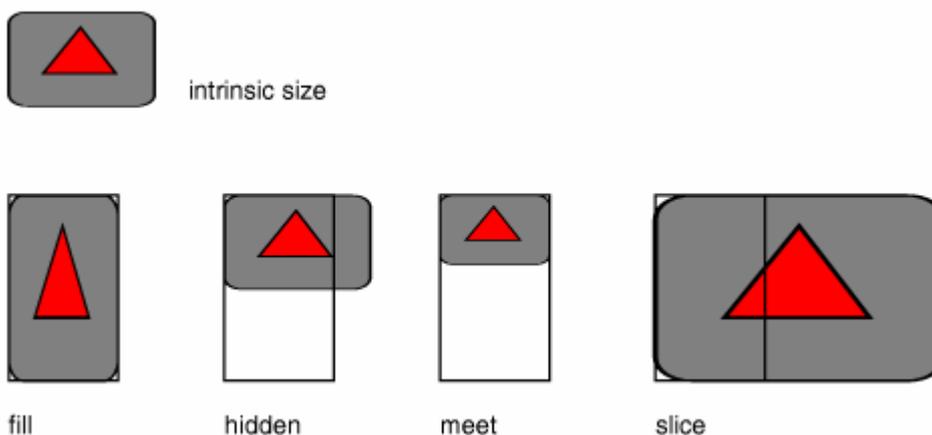
16.22 Vlastnost "max-height"

- Vlastnost nastavuje maximální hodnotu výšky elementu.
- Vlastnost lze aplikovat na všechny elementy, kromě inline-elementů s vlastností *display-model: inline-inside*.
- Vlastnost není dědičná.
- Hodnoty: *<length>* | *<percentage>* | "auto" | "inherit".
- Implicitní hodnota: *auto*.
- Funkčnost: FireFox 2.0
Opera 9.2
Internet Explorer 7.0

Poznámka: min-width je nadřazená vlastnosti max-width. Obě jsou nadřazené vlastnosti width, pokud jsou se hodnoty v rozporu. Analogie platí pro vlastnosti min-height, max-height and height

16.23 Vlastnost "fit"

- Vlastnost nastavuje, jak změnit velikost elementu, pokud hodnota width a height není nastavena na auto.
- Vlastnost lze aplikovat na replaced elementy.
- Vlastnost je dědičná.
- Hodnoty: *"fill"* (změní velikost objektu tak, aby vyplnil celou plochu hranici) | *none* (nemění velikost objektu) | *"meet"* (změní velikost na maximální možnou, zachovává poměr stran) | *"slice"* (změní velikost na minimální možnou, zachovává)
- Implicitní hodnota: *fill*.



16.24 Vlastnost "fit-position"

- Vlastnost nastavuje zarovnání objektu uvnitř elementu.
- Vlastnost lze aplikovat na replaced elementy.
- Vlastnost je dědičná.
- Hodnoty: [[<percentage> | <length>]{1,2} | ["top" | "center" | "bottom"] || ["left" | "center" | "right"]] | "auto" (zarovnání dle vlastnosti writing-mode).
- Hodnoty mají stejný význam jako u *background-position*.
- Implicitní hodnota: *0% 0%*.

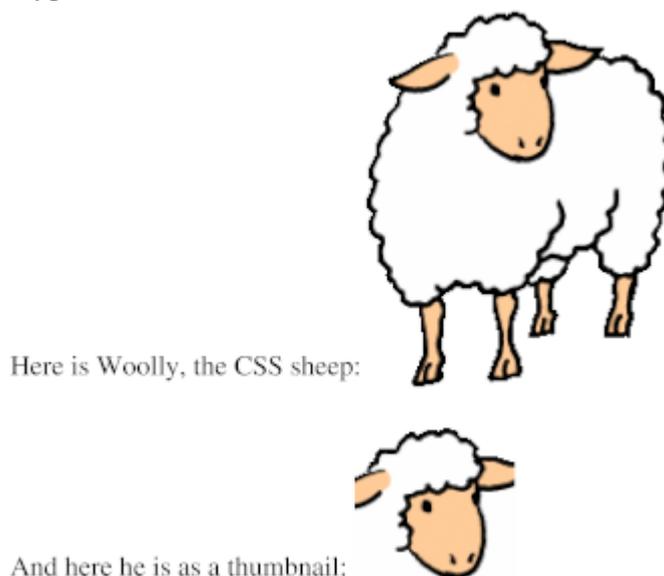
16.25 Vlastnost "crop"

- Vlastnost umožní vyříznout část elementu.
- Vlastnost lze aplikovat na nahrazené elementy.
- Vlastnost není dědičná.
- Hodnoty: `<shape> (rect(top, right, bottom, left) | inset-rect(top, right, bottom, left)) | "none"`.
- Implicitní hodnota: `auto` (vypočtená hodnota je stejná jako aktuální hodnota).

Příklad:

```
<p>Tady máme ovečku </p>  
<p>A tady máme miniaturu ovečky 
```

Výsledek by měl vypadat asi takto:



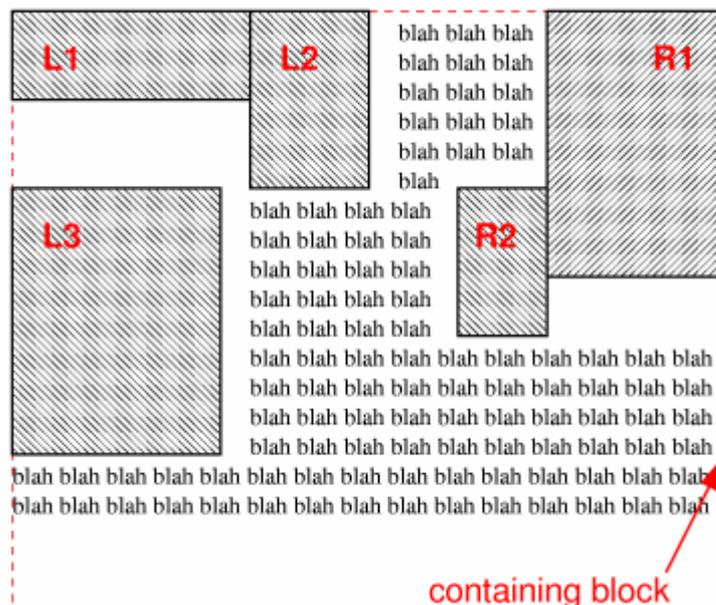
Obrázek 3 - použití vlastnosti crop

16.26 Vlastnost "float"

- Vlastnost nastavuje, jakým způsobem bude daný element obtékán.
- Vlastnost lze aplikovat na všechny elementy, kromě absolutně pozicovaných.
- Vlastnost není dědičná.
- Hodnoty: *"left"* (element je umístěn vlevo, je obtékán zprava) | *"right"* (element je umístěn vpravo, je obtékán zleva) | *"top"* (element je umístěn nahoře, je obtékán zdola) | *"bottom"* (element je umístěn dole, je obtékán shora) | *"inside"* (stejný efekt jako u *left* nebo *right*, u nestránkovaných dokumentů je to *left*) | *"outside"* (stejný jako u *right* nebo *left* u nestránkových dokumentů je to *right*) | *"start"* (stejný efekt jako *left*, *right*, *top* nebo *bottom*, záleží zde na směru – např. pokud bude směr toku obsahu *ltr* pak efekt bude *left*, *top*, apod.) | *"end"* (obdobně jako *start*) | *"none"* (obtěkání není povoleno).
- Implicitní hodnota: *none*.
- Funkčnost: FireFox 2.0
 Opera 9.2
 Internet Explorer 7.0

Příklad:

```
<STYLE TYPE="text/css">
P { width: 24em }
#L1 { float: left; width: 8em; height: 3em }
#L2 { float: left; width: 4em; height: 6em }
#R1 { float: right; width: 6em; height: 9em }
#L3 { float: left; width: 7em; height: 9em }
#R2 { float: right; width: 3em; height: 5em }
</STYLE>
<P>
<IMG ID=L1 SRC="L1.png" ALT="L1">
<IMG ID=L2 SRC="L2.png" ALT="L2">
<IMG ID=R1 SRC="R1.png" ALT="R1">
<IMG ID=L3 SRC="L3.png" ALT="L3">
<IMG ID=R2 SRC="R2.png" ALT="R2">
blah bla blah bla blah...
</P>
```



Obrázek 4 - použití vlastnosti float

16.27 Vlastnost "clear"

- Vlastnost nastavuje, jakým způsobem bude element obtékat jiný element.
- Vlastnost lze aplikovat na blokové elementy.
- Vlastnost není dědičná.
- Hodnoty: *"none"* (element může obtékat z obou stran) | *"left"* (element nesmí obtékat zleva) | *"right"* (element nesmí obtékat zprava) | *"top"* (element nesmí obtékat shora) | *"bottom"* (element nesmí obtékat zdola) | *"inside"* (stejně jako *left* nebo *right*, u nestránkovaných dokumentů je to *left*) | *"outside"* (stejně jako *right* nebo *left*, u stránkovaných dokumentů je to *right*) | *"start"* (stejný efekt jako *left*, *right*, *top* nebo *bottom*, záleží zde na směru – např. pokud bude směr toku obsahu *ltr* pak efekt bude *left*, *top*, apod.) | *"end"* (obdobně jako *start*) | *"both"* (znemožňuje obtékání).
- Implicitní hodnota: *none*.
- Funkčnost:
 - Firefox 2.0
 - Opera 9.2
 - Internet Explorer 7.0

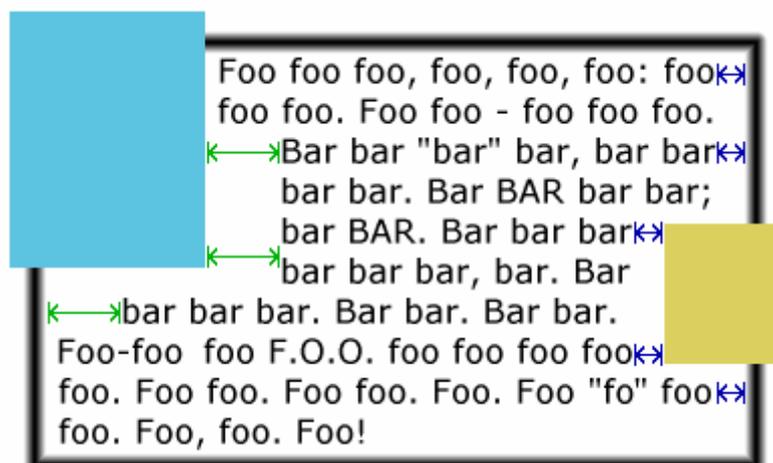
16.28 Vlastnost "clear-after"

- Vlastnost nastavuje odsazení od spodního okraje boxu.
- Vlastnost lze aplikovat na blokové elementy.
- Vlastnost není dědičná.
- Hodnoty: *"none"* (žádný efekt) | *"left"* (bude bráno v úvahu pouze obtékání zleva) | *"right"* (bude bráno v úvahu pouze obtékání zprava) | *"top"* (bude bráno v úvahu pouze levé obtékání shora) | *"bottom"* (bude bráno v úvahu pouze levé obtékání zdola) | *"inside"* (stejný efekt jako u *left* nebo *right*, u nestránkovaných dokumentů je to *left*) | *"outside"* (stejný efekt jako u *right* nebo *left* u nestránkových dokumentů je to *right*.) | *"start"* (je stejný jako *left*, *right*, *top* nebo *bottom*, záleží zde na řízení – např. řízení bude *ltr* pak efekt bude *left*, *top* apod.) | *"end"* (obdobně jako *start*) | *"both"* (znemožňuje obtékání).
- Implicitní hodnota: *none*.

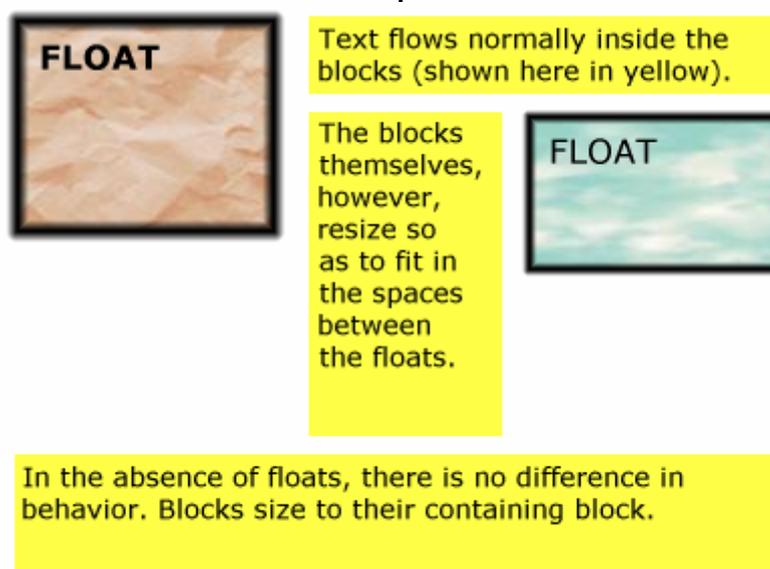
Poznámka: Je možné, že bude vlastnost clear-after zrušena a místo ní bude k vlastnosti clear přidána hodnota "after".

16.29 Vlastnost "float-displace"

- Vlastnost nastavuje, jak budou dělena slova při zapnutém obtékání.
- Vlastnost lze aplikovat na blokové elementy.
- Vlastnost je dědičná.
- Hodnoty: *"line"* (řádkové boxy budou zkráceny a přesunuty tak, aby utvořili obtékání) | *"indent"* (zajišťuje udržení relativního odsazení, i přes obtékání) | *"block"* (obsahující blokovou šíři užívanou při vodorovném formátovacím modelu) | *"block-within-page"* (stejně jako hodnota *block*, ale bez stanovení, které ruší přizpůsobení šíře bloku, to je doděláno samostatně pro každou stránku, na které se blok objeví).
- Implicitní hodnota: *line*.



Obrázek 5 - float-displace: indent



Obrázek 6 - float-displace: block

16.30 Vlastnost "indent-edge-reset"

- Tato vlastnost určuje, který okraj se použije jako výchozí pro výpočet velikosti odsazení, které se má zachovat.
- Vlastnost lze aplikovat na blokové elementy s uvnitř formátovaným obsahem.
- Vlastnost není dědičná.
- Hodnoty: *"none"* (blok se neodkazuje na žádnou novou vodicí hranu) | *"margin-edge"* | *"border-edge"* | *"padding-edge"* | *"content-edge"* | *"inherit"*.
- Implicitní hodnota: *none*.

16.31 Vlastnost "overflow-x"

- Vlastnost nastavuje způsob práce, v horizontálním směru, s elementy, které se nevejdou do nadefinovaného prostoru.
- Vlastnost lze aplikovat na blokové nebo nahrazované elementy.
- Vlastnost není dědičná.
- Hodnoty: *"visible"* (elementu je povoleno libovolně přetékat stanovený rámeček) | *"hidden"* (element je oříznut na velikost rámečku) | *"scroll"* (element je možno v menším rámečku skrolovat a tím prohlížet celý) | *"auto"* (typ zobrazení se vybere dle možností prohlížeče) | *"inherit"*.
- Implicitní hodnota: *visible*.

16.32 Vlastnost "overflow-y"

- Vlastnost nastavuje způsob práce, ve vertikálním směru, s elementy, které se nevejdou do nadefinovaného prostoru.
- Vlastnost lze aplikovat na blokové nebo nahrazované elementy.
- Vlastnost není dědičná.
- Hodnoty: *"visible"* (elementu je povoleno libovolně přetékat stanovený rámeček) | *"hidden"* (element je oříznut na velikost rámečku) | *"scroll"* (element je možno v menším rámečku skrolovat a tím prohlížet celý) | *"auto"* (typ zobrazení se vybere dle možností prohlížeče) | *"inherit"*.
- Implicitní hodnota: *visible*.

16.33 Vlastnost "overflow"

- Vlastnost nastavuje způsob práce s elementy, které se nevejdou do nadefinovaného prostoru.
- Vlastnost lze aplikovat na blokové nebo nahrazované elementy.
- Vlastnost není dědičná.
- Hodnoty: *"visible"* (elementu je povoleno libovolně přetékat stanovený rámeček) | *"hidden"* (element je oříznut na velikost rámečku) | *"scroll"* (element je

možno v menším rámečku skrolovat a tím prohlížet celý) | *"auto"* (typ zobrazení se vybere dle možností prohlížeče) | *"inherit"*.

- Implicitní hodnota: *visible*.
- Funkčnost: FireFox 2.0
 Opera 9.2
 Internet Explorer 7.0

16.34 Vlastnost **"marquee-style"**

- Vlastnost jakým způsobem se bude obsah pohybovat.
- Vlastnost lze aplikovat na blokové nebo nahrazované elementy.
- Vlastnost je dědičná.
- Hodnoty: *"none"* (nebude se pohybovat) | *"slide"* (obsah začíná skrolovat vně obsahu boxu) | *"scroll"* (obsah začíná skrolovat vně obsahu boxu a skroluje ho až do posledního řádku) | *"alternate"* (obsah začíná jako viditelný, s jedním okrajem proti okraji boxu, pak skroluje do té doby, než protější okraj obsahu je proti protějšímu okraji boxu).
- Implicitní hodnota: *alternate*.

16.35 Vlastnost **"marquee-direction"**

- Vlastnost nastavuje, kterým směrem se bude obsah pohybovat.
- Vlastnost lze aplikovat na blokové nebo nahrazované elementy.
- Vlastnost je dědičná.
- Hodnoty: *"forwards"* (ve směru psaní) | *"backwards"* (proti směru psaní) | *"ahead"* (ve směru toku textu) | *"reverse"* (proti směru toku textu) | *"left"* (doleva) | *"right"* (doprava) | *"up"* (nahoru) | *"down"* (dolů) | *"auto"* (nastavuje automatické rolování obsahu).
- Implicitní hodnota: *auto*.

16.36 Vlastnost "marquee-speed"

- Vlastnost nastavuje rychlost pohybu.
- Vlastnost lze aplikovat na blokové nebo nahrazované elementy.
- Vlastnost je dědičná.
- Hodnoty: *"slow"* (pomalu) | *"normal"* (normálně) | *"fast"* (rychle) | *<length>* (vzdálenost za sekundu).
- Implicitní hodnota: *normal*.

Poznámka: Možná bude přidána hodnota <length>/<time>, která bude vyjadřovat posun za jednotku času.

16.37 Vlastnost "marquee-repetition"

- Vlastnost nastavuje počet opakování.
- Vlastnost lze aplikovat na blokové nebo nahrazované elementy.
- Vlastnost je dědičná.
- Hodnoty: *<integer>* (počet opakování) | *"infinite"* (nekonečná smyčka).
- Implicitní hodnota: *infinite*.

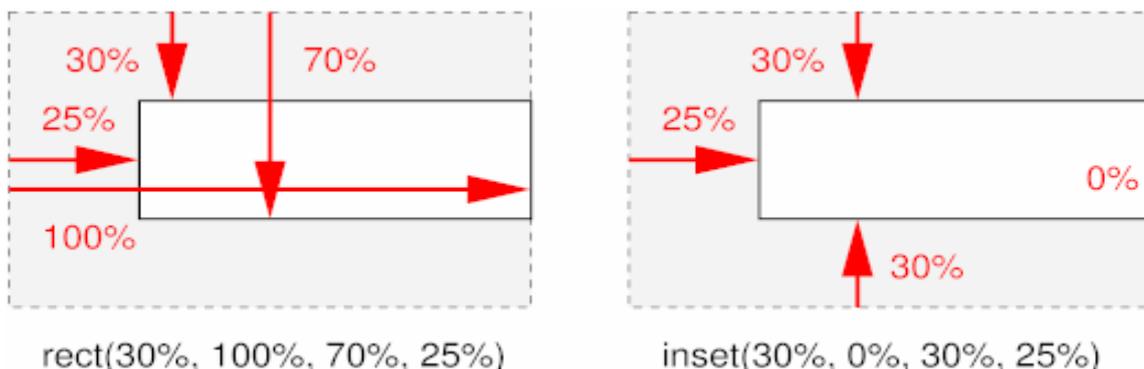
16.38 Vlastnost "marquee"

- Souhrnná vlastnost marquee.
- Vlastnost lze aplikovat na blokové nebo nahrazované elementy.
- Vlastnost je dědičná.
- Hodnoty: *<marquee-style>* || *<marquee-direction>* || *<marquee-speed>* || *<marquee-repetition>*.
- Implicitní hodnota: *dle jednotlivých vlastností*.

16.39 Vlastnost "overflow-clip"

- Vlastnost určuje, která část obsahu bude viditelná při přetečení.
- Vlastnost lze aplikovat na blokové nebo nahrazované elementy.
- Vlastnost není dědičná.

- Hodnoty: *"auto"* (nastaveno automaticky) | *"rect(T, R, B, L)"* (obsah je zkrácený vzhledem k obdélníku daného čtyřmi vyrovnáními z levého rohu obsahu plochy (nebo dalšího rohu, v závislosti na *writing-mode*) | *"inset-rect(t, r, b, l)"* (zkracuje vzhledem k obdélníku danému čtyřmi hodnotami a je vyrovnaný ke čtveřici okrajů obsahu oblasti).
- Implicitní hodnota: *auto*.



Obrázek 7 - overflow-clip: `rect(30%,100%,70%,25%)`, overflow-clip: `inset(30%,0%,30%,25%)`

16.40 Vlastnost "visibility"

- Vlastnost nastavuje viditelnost elementu.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"visible"* (element je vidět) | *"hidden"* (element není vidět, resp. je plně transparentní) | *"collapse"* (vlastnost umožňuje v tabulce vypouštět řádky, resp. sloupce, pokud je vlastnost použita jinde, má stejnou vlastnost jako *hidden*).
- Implicitní hodnota: *visible*.
- Funkčnost:
 - FireFox 2.0
 - Opera 9.2
 - Internet Explorer 7.0

17 Pozadí a okraje

Tento modul obsahuje funkce, které popisují vzhled a práci s okraji a pozadím. Obsahuje i možnosti vytvořit okraje pomocí obrázků nebo pozadí s více obrázky.

17.1 Vlastnost "background-color"

- Vlastnost nastavuje barvu rámečku.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *<color>*.
- Implicitní hodnota: *transparent* (průhledné pozadí).
- Funkčnost: Firefox 2.0
 Opera 9.2
 Internet Explorer 7.0

Příklad:

```
h1 { background-color: #FFF }
```

17.2 Vlastnost "background-image"

- Vlastnost nastavuje jako pozadí obrázek.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *<uri>* (umístění obrázku) | *"none"* (žádný obrázek).
- Implicitní hodnota: *none*.
- Funkčnost: Firefox 2.0
 Opera 9.2
 Internet Explorer 7.0

Poznámka: Je možné zadat seznam <uri> oddělený čárkami. Obrázky pak budou vrstveny dle pořadí, první v seznamu bude úplně nahoře.

Příklad: dvě různé deklarace, které mají stejný význam.

```
background-image: url(a), url(b);  
background-position: top, right, bottom, left, center;  
background-repeat: no-repeat, no-repeat;
```

```
background-image: url(a), url(b), url(a), url(b), url(a);  
background-position: top, right, bottom, left, center;  
background-repeat: no-repeat, no-repeat, no-repeat,  
no-repeat, no-repeat;
```

17.3 Vlastnost "background-repeat"

- Obrázek může být na pozadí umístěn pouze jednou, nebo se může dlaždicově opakovat ve svislém, vodorovném či obou směrech. V místech, kde obrázek není a pod jeho průhlednými oblastmi je zobrazena plocha daná hodnotou *background-color* (případně nic, pokud je hodnota *transparent*).
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: "*repeat-x*" (opakování obrázku v ose x) | "*repeat-y*" (opakování obrázku v ose y) | "*repeat*" (opakování obrázku v obou osách) | "*space*" (obrázek je vkládán tak, aby se na pozadí vešly celé násobky obrázku) | "*no-repeat*" (ve směru je zakázáno opakování).
- Implicitní hodnota: *repeat*.

Poznámka: repeat-x = repeat no-repeat, repeat-y = no-repeat repeat

17.4 Vlastnost " background-attachment

- Vlastnost určuje, zda dochází k pohybu pozadí s elementem nebo zda je pozadí fixováno a pohybuje se pouze element po pozadí.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: "*scroll*" (pozadí skroluje společně s elementem) | "*fixed*" (pozadí je fixováno) | "*local*" (pokud má element definováno skrolování, pak pozadí skroluje společně s textem).

- Implicitní hodnota: *scroll*.
- Funkčnost: FireFox 2.0
Opera 9.2
Internet Explorer 7.0

17.5 Vlastnost "background-position"

- Vlastnost určuje výchozí pozici pro vkládání obrázku do pozadí.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: délkové jednotky (X Y), *<percentage>* (vztahují se k velikosti vlastního elementu), pro X pozici "left", "center", "right", pro Y pozici "top", "center", "bottom".
- Implicitní hodnota: 0% 0%.
- Funkčnost: FireFox 2.0
Opera 9.2
Internet Explorer 7.0

Poznámka: Klíčová slova se nesmí kombinovat s procenty.

Příklad:

```
body { background: url("banner.jpeg") right top }  
/* 100% 0% */  
body { background: url("banner.jpeg") top center }  
/* 50% 0% */  
body { background: url("banner.jpeg") center }  
/* 50% 50% */  
body { background: url("banner.jpeg") bottom }  
/* 50% 100% */
```

17.6 Vlastnost "background-clip"

- Vlastnost určuje, jestli bude pozadí dokumentu rozšířeno i za okraje dokumentu.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *"border"* (pozadí není roztáhnuto za okraje) | *"padding"* (pozadí je roztáhnuto a je transparentní).
- Implicitní hodnota: *border*.

17.7 Vlastnost "background-origin"

- Vlastnost určuje, jak bude vlastnost vypočítána *background-position*.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *"border"* (pozice je relativní vůči ohraničení elementu) | *"padding"* (pozice je relativní vůči levému hornímu a spodnímu pravému rohu odsazení) | *"content"* (pozice je relativní vůči obsahu).
- Implicitní hodnota: *padding*.

17.8 Vlastnost "background-size"

- Vlastnost nastavuje velikost obrázku, který je použit v pozadí.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *<length>* | *<percentage>* | *"auto"* (obrázek si zachovává původní rozlišení) | *"round"* (volitelná hodnota pro zaokrouhlení vypočtených hodnot).

Vlastnost má dvě hodnoty. První argument vyjadřuje šířku, druhý výšku. Pokud je použita pouze jedna, pak je druhá hodnota brána jako *auto*.

- Implicitní hodnota: *auto*.

17.9 Vlastnost "background-break"

- Vlastnost nastavuje chování inline elementů nebo blokových elementů, které jsou rozděleny do více boxů (např. přes více stránek). Pro jednotlivé typy boxů se chová různě.
- Vlastnost lze aplikovat na inline elementy a blokové elementy.
- Vlastnost není dědičná.
- Hodnoty: *"bounding-box"* (uzavře všechny inline boxy elementu do co nejmenšího obdélníku) | *"each-box"* | *"continuous"*.
- Implicitní hodnota: *continuous*.

17.10 Vlastnost "background"

- Souhrnná vlastnost pro nastavení pozadí.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *<background-image>* || (*<background-size>*) || *<background-repeat>* || *<background-position>* || *<background-attachment>* || *<background-clip>* *<background-origin>* || *<background-color>*.
- Implicitní hodnota: *dle jednotlivých vlastností*.

Příklad:

```
body { background: red }
p { background: url("chess.png") (10em round) gray 40%
  repeat
    fixed border border }
```

První pravidlo je ekvivalentní:

```
body {
  background-color: red;
  background-position: 0% 0%;
  background-size: 30% 30%;
  background-repeat: repeat repeat;
  background-clip: border;
  background-origin: padding;
  background-attachment: scroll;
  background-image: none }
```

Druhé pravidlo je ekvivalentní:

```
p {  
    background-color: gray;  
    background-position: 40% 50%;  
    background-size: 10em 10em round;  
    background-repeat: repeat repeat;  
    background-clip: border;  
    background-origin: border;  
    background-attachment: fixed;  
    background-image: url(chess.png) }
```

17.11 Vlastnost "border-top-color"

- Vlastnost nastavuje barvu horní části rámečku.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *<color>*.
- Implicitní hodnota: *currentcolor*.
- Funkčnost: FireFox 2.0
 Opera 9.2
 Internet Explorer 7.0

17.12 Vlastnost "border-right-color"

- Vlastnost nastavuje barvu pravé části rámečku.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *<color>*.
- Implicitní hodnota: *currentcolor*.

17.13 Vlastnost "border-bottom-color"

- Vlastnost nastavuje barvu dolní části rámečku.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *<color>*.
- Implicitní hodnota: *currentcolor*.

17.14 Vlastnost "border-left-color"

- Vlastnost nastavuje barvu levé části rámečku.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *<color>*.
- Implicitní hodnota: *currentcolor*.

17.15 Vlastnost "border-color"

- Souhrnná vlastnost nastavuje barvu rámečku.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *<color>* Vlastnost má buď čtyři, tři, dva nebo jeden parametr. V případě, že má parametry čtyři, nastavujeme barvu rámečků v pořadí nahoře, vpravo, dole, vlevo. V případě tří parametrů nastavujeme v pořadí nahoře, vpravo a zároveň vlevo, dole. V případě dvou parametrů nahoře a zároveň dole, vlevo a zároveň vpravo. A v případě jednoho parametru jsou všechny rámečky stejné.
- Implicitní hodnota: *currentcolor*.
- Funkčnost: FireFox 2.0
 Opera 9.2
 Internet Explorer 7.0

17.16 Vlastnost "border-top-style"

- Vlastnost nastavuje styl horní části rámečku.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *"none"* (bez linky) | *"hidden"* (skrytá) | *"dotted"* (tečkovaná) | *"dashed"* (čárkovaná) | *"solid"* (plná) | *"double"* (dvojitá plná) | *"groove"* (rytina) | *"ridge"* (reliéf) | *"inset"* (zapuštěný vzhled) | *"outset"* (vystouplý vzhled).
- Implicitní hodnota: *none*.
- Funkčnost: FireFox 2.0
 Opera 9.2
 Internet Explorer 7.0

17.17 Vlastnost "border-right-style"

- Vlastnost nastavuje styl pravé části rámečku.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *"none"* (bez linky) | *"hidden"* (skrytá) | *"dotted"* (tečkovaná) | *"dashed"* (čárkovaná) | *"solid"* (plná) | *"double"* (dvojitá plná) | *"groove"* (rytina) | *"ridge"* (reliéf) | *"inset"* (zapuštěný vzhled) | *"outset"* (vystouplý vzhled).
- Implicitní hodnota: *none*.

17.18 Vlastnost "border-down-style"

- Vlastnost nastavuje styl dolní části rámečku.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *"none"* (bez linky) | *"hidden"* (skrytá) | *"dotted"* (tečkovaná) | *"dashed"* (čárkovaná) | *"solid"* (plná) | *"double"* (dvojitá plná) | *"groove"* (rytina) | *"ridge"* (reliéf) | *"inset"* (zapuštěný vzhled) | *"outset"* (vystouplý vzhled).
- Implicitní hodnota: *none*.

17.19 Vlastnost "border-left-style"

- Vlastnost nastavuje styl horní části rámečku.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *"none"* (bez linky) | *"hidden"* (skrytá) | *"dotted"* (tečkovaná) | *"dashed"* (čárkovaná) | *"solid"* (plná) | *"double"* (dvojitá plná) | *"groove"* (rytina) | *"ridge"* (reliéf) | *"inset"* (zapuštěný vzhled) | *"outset"* (vystouplý vzhled).
- Implicitní hodnota: *none*.

17.20 Vlastnost "border-style"

- Vlastnost nastavuje styl horní části rámečku.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: `<border-style>` Vlastnost má buď čtyři, tři, dva nebo jeden parametr. V případě, že má parametry čtyři, nastavujeme styl rámečků v pořadí nahoře, vpravo, dole, vlevo. V případě tří parametrů nastavujeme v pořadí nahoře, vpravo a zároveň vlevo, dole. V případě dvou parametrů nahoře a zároveň dole, vlevo a zároveň vpravo. A v případě jednoho parametru jsou všechny rámečky stejné.
- Implicitní hodnota: *none*.
- Funkčnost: FireFox 2.0
 Opera 9.2
 Internet Explorer 7.0

17.21 Vlastnost "border-top-width"

- Vlastnost nastavuje šířku rámečku nad elementem.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: `<length>` | `<percentage>` | `"thin"` (tenký rámeček) | `"medium"` (středně široký rámeček) | `"thick"` (široký rámeček).
- Implicitní hodnota: `medium`.
- Funkčnost: FireFox 2.0
 Opera 9.2
 Internet Explorer 7.0

17.22 Vlastnost "border-right-width"

- Vlastnost nastavuje šířku rámečku za elementem
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: `<length>` | `<percentage>` | `"thin"` (tenký rámeček) | `"medium"` (středně široký rámeček) | `"thick"` (široký rámeček).
- Implicitní hodnota: `medium`.

17.23 Vlastnost "border-bottom-width"

- Vlastnost nastavuje šířku rámečku pod elementem.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: `<length>` | `<percentage>` | `"thin"` (tenký rámeček) | `"medium"` (středně široký rámeček) | `"thick"` (široký rámeček).
- Implicitní hodnota: `medium`.

17.24 Vlastnost "border-left-width"

- Vlastnost nastavuje šířku rámečku před elementem.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *<length>* | *<percentage>* | "thin" (tenký rámeček) | "medium" (středně široký rámeček) | "thick" (široký rámeček).
- Implicitní hodnota: *medium*.

17.25 Vlastnost "border-width"

- Vlastnost komplexně nastavuje šířku rámečku kolem celého elementu. Je kombinací vlastností *border-top-width*, *border-right-width*, *border-bottom-width*, *border-left-width*.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: Tato vlastnost má buď čtyři, tři, dva nebo jeden parametr. V případě, že má parametry čtyři, nastavujeme šířku rámečků v pořadí nahoře, vpravo, dole, vlevo. V případě tří parametrů nastavujeme v pořadí nahoře, vpravo a zároveň vlevo, dole. V případě dvou parametrů nahoře a zároveň dole, vlevo a zároveň vpravo. A v případě jednoho parametru jsou všechny rámečky stejné. Hodnoty parametrů jsou shodné s předchozími vlastnostmi.
- Implicitní hodnota: *medium*.
- Funkčnost: FireFox 2.0
 Opera 9.2
 Internet Explorer 7.0

Poznámka: Implicitní hodnota šířky medium, implicitní hodnota border-style je none, takže výsledná šířka bude 0.

17.26 Vlastnost "border-image"

- Vlastnost nastavuje obrázek, který bude použit místo vlastnosti *border-style*.
- Vlastnost lze aplikovat na všechny elementy až na table element, který má vlastnost
border-collapse: collapse.
- Vlastnost není dědičná.
- Hodnoty: *"none"* | *<uri>* [*<number>* | *<percentage>*]{4} (čtyři čísla nebo procenta rozdělí obrázek na 9 částí, první hodnota je velikost horní částí, druhá hodnota je velikost pravé části, třetí je velikost dolní části, čtvrtý je velikost levé části) [/ *<border-width>*{1,4} (nastaví šířku rámečku)] [*"stretch"* | *"repeat"* | *"round"*]{0,2} (nastavení měřítko).
- Implicitní hodnota: *none*.

Příklad:

Obrázek má velikost 81 krát 81 pixelů.

```
DIV {
  ..width: 12em; height: 5em;
  border-image: url("border.png") 27 27 27 27 round
  stretch;
  border: double orange 1em }
```

Výsledek by měl vypadat asi takto:



Obrázek 8 - výsledek

17.27 Vlastnost "border-top-right-radius"

- Vlastnost nastavuje zaoblení pravého horního rohu.
- Vlastnost lze aplikovat na všechny elementy až na table element, který má vlastnost *border-collapse: collapse*.
- Vlastnost není dědičná.
- Hodnoty: *<length> <length> {1,2}* (délka hlavní a vedlejší poloosy).
- Implicitní hodnota: *0*.

17.28 Vlastnost "border-bottom-right-radius"

- Vlastnost nastavuje zaoblení pravého dolního rohu.
- Vlastnost lze aplikovat na všechny elementy až na table element, který má vlastnost *border-collapse: collapse*.
- Vlastnost není dědičná.
- Hodnoty: *<length> <length> {1,2}* (délka hlavní a vedlejší poloosy).
- Implicitní hodnota: *0*.

17.29 Vlastnost "border-bottom-left-radius"

- Vlastnost nastavuje zaoblení levého dolního rohu.
- Vlastnost lze aplikovat na všechny elementy až na table element, který má vlastnost *border-collapse: collapse*.
- Vlastnost není dědičná.
- Hodnoty: *<length> <length> {1,2}* (délka hlavní a vedlejší poloosy).
- Implicitní hodnota: *0*.

17.30 Vlastnost "border-top-left-radius"

- Vlastnost nastavuje zaoblení levého horního rohu.
- Vlastnost lze aplikovat na všechny elementy až na table element, který má vlastnost *border-collapse: collapse*.
- Vlastnost není dědičná.
- Hodnoty: *<length> <length> {1,2}* (délka hlavní a vedlejší poloosy).
- Implicitní hodnota: *0*.

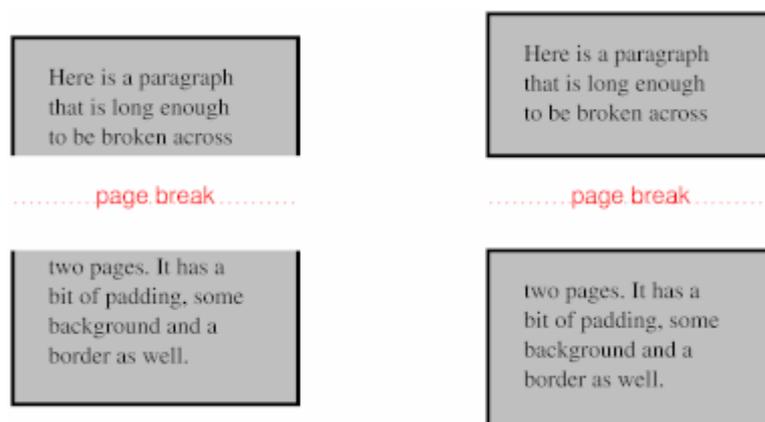
17.31 Vlastnost "border-radius"

- Souhrnná vlastnost nastavuje zaoblení rohů.
- Vlastnost lze aplikovat na všechny elementy až na table element, který má vlastnost *border-collapse: collapse*.
- Vlastnost není dědičná.
- Hodnoty: *<length> <length> {1,2}* (délka hlavní a vedlejší poloosy).
- Implicitní hodnota: *0*.

17.32 Vlastnost "border-break"

- Vlastnost nastavuje vzhled rámečku při na novou stránku.
- Vlastnost lze aplikovat na všechny elementy, které mají okraj.
- Vlastnost je dědičná.
- Hodnoty: *<border-width> || <border-style> || <color>*
- Implicitní hodnota: *none*.

Příklad:



Obrázek 9 - První má nastaveno none, druhý solid.

17.33 Vlastnost "border-top"

- Souhrnná vlastnost pro nastavení horní části rámečku.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: `<border-width> || <border-style> || <color>`.
- Implicitní hodnota: *dle jednotlivých vlastností*.
- Funkčnost:
 - Firefox 2.0
 - Opera 9.2
 - Internet Explorer 7.0

17.34 Vlastnost "border-right"

- Souhrnná vlastnost pro nastavení pravé části rámečku.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: `<border-width> || <border-style> || <color>`.
- Implicitní hodnota: *dle jednotlivých vlastností*.

17.35 Vlastnost "border-down"

- Souhrnná vlastnost pro nastavení levé části rámečku.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: `<border-width> || <border-style> || <color>`.
- Implicitní hodnota: *dle jednotlivých vlastností*.

17.36 Vlastnost "border-left"

- Souhrnná vlastnost pro nastavení levé části rámečku.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: `<border-width> || <border-style> || <color>`.
- Implicitní hodnota: *dle jednotlivých vlastností*.

17.37 Vlastnost "border"

- Souhrnná vlastnost pro nastavení rámečku.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: `<border-width> || <border-style> || <color>`.
- Implicitní hodnota: *dle jednotlivých vlastností*.
- Funkčnost: FireFox 2.0
 Opera 9.2
 Internet Explorer 7.0

Poznámka: Na rozdíl od vlastností jako margin, padding nastaví tato vlastnost všechny okraje stejně.

Příklad:

```
p { border: solid red }
p {
  border-top: solid red;
  border-right: solid red;
  border-bottom: solid red;
  border-left: solid red;
}
```

17.38 Vlastnost "box-shadow"

- Vlastnost nastavuje stín okolo boxu.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *"none"* | [*<length>* *<length>* *<length>* (horizontální posun, vertikální posun, poloměr rozmazání) || *<color>*].
- Implicitní hodnota: *none*.

18 Pokročilé rozmístění prvků

Vzhled webové stránky by šel zhruba rozdělit na dvě části: první popisuje rozložení jednotlivých prvků na stránce, druhé specifikuje písma, odsazení, barvy, atd. Tedy rozložení jednotlivých prvků se při změně stránky mění, kdežto písmo, barvy, apod. zůstávají.

Tento modul umožňuje nadefinovat neviditelnou síť, do které budou prvky umístěny podle různých pravidel. Např. můžeme elementy poskládat do gridu podobnému balíčku karet, kdy bude viditelný v jeden okamžik pouze jeden element.

Typické použití těchto vlastností bude v následujících případech:

- Komplexní webové stránky s několika navigačními pásy, které mají pevnou pozici, prostory pro reklamy, atd.
- Komplexní formuláře, které je snadnější vyrovnat pomocí tohoto modulu než pomocí tabulek a okrajů.
- Grafická uživatelská rozhraní s tlačítky, nástrojovými lištami, ikonami, atd., které jsou různě umístěny a musí tak zůstat, i když se okno zvětší.
- Překrývající se elementy, kdy je viditelný pouze horní element.

18.1 Pozicování pomocí šablon

Pozicování pomocí šablon je alternativou k absolutnímu pozicování, které je užitečné k umístění elementů, které nemají jednoduché vztahy se zdrojem. Na rozdíl od absolutního pozicování nejsou elementy umístěny pomocí souřadnic, ale pomocí přiřazení jednotlivých prvků do slotů podobně jako u tabulek. Relativní velikost a zarovnání elementu bude záležet na šabloně. Prvky nesmějí přecházet, ale umí se lépe vypořádat s různými šířkami.

Přiřazení je prováděno pomocí vlastnosti *position*, která specifikuje, do kterého slotu bude element přiřazen. Samotná šablona je specifikována pomocí řetězce ve vlastnosti *display-model* nebo *display*.

Příklad: V tomto příkladě máme čtyři elementy přiřazeny do čtyř slotů (nazvaných a, b, c, d).

A	B
C	D

Každý element je v jednom slotu, všechny sloty mají stejnou velikost.

```
<style type="text/css">
  dl { display: "ab"
        "cd" }
  #sym1 { position: a }
  #lab1 { position: b }
  #sym2 { position: c }
  #lab2 { position: d }
</style>
<dl>
  <dt id=sym1>A
  <dd id=lab1>B
  <dt id=sym2>C
  <dd id=lab2>D
</dl>
```

18.1.1 Vlastnost "display-model"

- Element s tímto *display-modelem* je podobný tabulkovému elementu. Jeho obsah je rozčleněn do řádků a sloupců. Dva hlavní rozdíly jsou: počet řádků a sloupců nezávisí na obsahu, ale je pevně dán hodnotou vlastnosti, pořadí ve zdrojovém dokumentu se může lišit od zobrazení, které bude interpretováno pomocí šablony.
- Hodnoty: `<string>` `<row-height>` `<col-width>`.
 - `<string>` se skládá z jednoho nebo více písmen, at značek ("@") a opakování ("."). Každý string reprezentuje jednu řádku, každý znak reprezentuje jeden sloupec v řádku.
 - `<row-height>` nastavuje výšku řádky. Hodnoty: `<length>` (explicitní výška řádky), `"intrinsic"` (výška je daná obsahem), `"*"` (takto označené řádky budou mít shodnou výšku). Implicitní hodnota: `*`.

- `<col-width>` nastavuje šířku sloupce. Hodnoty: `<length>` (explicitní šířka sloupce), `"intrinsic"` (šířka je daná obsahem), `"*"` (takto označené sloupce budou mít shodnou šířku). Implicitní hodnota: `*`.
- Orientace šablony závisí na toku textu.

18.1.2 Vlastnost "position"

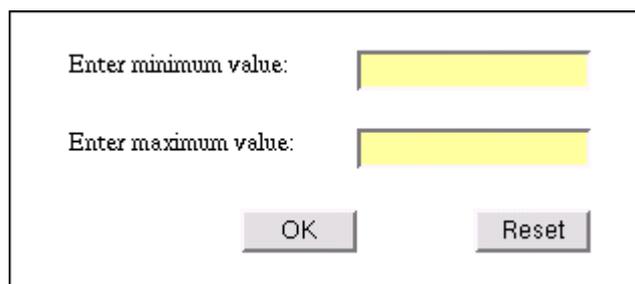
- Vlastnost nastavuje, do které části bude element umístěn.
- Hodnoty: `<letter>` (element je vyjmut a vložen do specifikovaného místa) | `"same"` (pozice je definována rodičovskou šablonou).

Příklad: Máme formulář, který je postaven na gridu se dvěma labely, dvěma input boxy, potvrzujícím a reset tlačítkem:

```
form {
  border: thin solid;
  display: "aaaa.bbbb"
         "....."
         "cccc.dddd"
         "....."
         "...ee..ff" }
label[for=minv] { position: a }
input#minv { position: b; display: block }
label[for=maxv] { position: c }
input#maxv { position: d; display: block }
input[type=submit] { position: e; display: block }
input[type=reset] { position: f; display: block }
```

Kód aplikovaný v html:

```
<form action="./">
  <label for=minv>Enter minimum value:</label>
  <input id=minv name=minv>
  <label for=maxv>Enter maximum value:</label>
  <input id=maxv name=maxv>
  <input type=submit value="OK">
  <input type=reset value="Reset">
</form>
```

A rectangular form with a black border. Inside, there are two rows of text and input fields. The first row contains the text "Enter minimum value:" followed by a yellow rectangular input field. The second row contains the text "Enter maximum value:" followed by another yellow rectangular input field. Below these fields, there are two gray rectangular buttons: "OK" on the left and "Reset" on the right.

Obrázek 10 - Možné vykreslení formuláře

18.1.3 Vertikální zarovnání

Vlastnost *vertical-align* umožňuje nastavení vertikálního umístění uvnitř slotu. Lze ji aplikovat pouze na první blokový element, a který neobtéká. Hodnoty: "*bottom*" (zarovnání do spodní části slotu) | "*middle*" (element je vycentrován) | "*baseline*" (zarovnání účaří s ostatními).

18.1.4 Přetékání

Obsah slotu je zobrazen, i když obsah přetéká. Je to stejný efekt, jako by byla nastavena vlastnost *overflow: visible*.

18.1.5 Šablony ve stránkovaných médiích

Obsah slotu je zobrazen, i když obsah přetéká. Je to stejný efekt, jako by byla nastavena vlastnost *overflow: visible*.

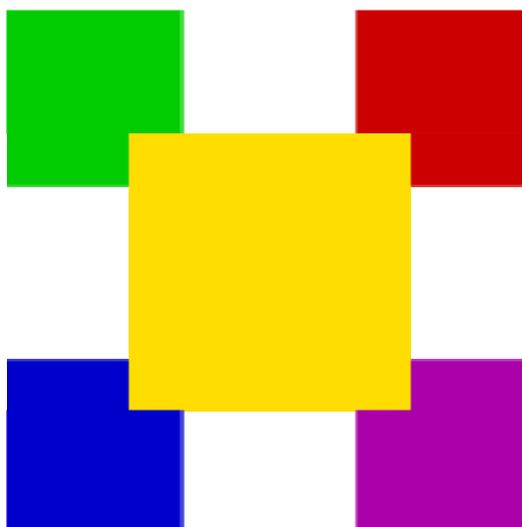
18.1.6 Pořadí

Element s vypočtenou hodnotou pro pozici je umístěn na základě vlastnosti *z-index*.

Příklad:

```
body { display: "a.b"  
        ".c."  
        "d.e";  
        height: 6cm;  
        width: 6cm }
```

```
#a { background: #0C0; position: a }  
#b { background: #C00; position: b }  
#c { background: #FD0; position: c; margin: -1em; z-  
index: 1 }  
#d { background: #00C; position: d }  
#e { background: #A0A; position: e }
```



Obrázek 11 - Možné vykreslení příkladu

18.1.7 Plovoucí elementy uvnitř šablony

Element může být uvnitř šablony pozicován a zároveň být obtékán. Musí být rozlišeny následující případy:

- Page-based floats – Pro stránkovaná média platí, pokud je specifikována vlastnost *float*, pak je vlastnost upřednostněna na úkor vlastnosti *position*.
- Normal folats – v ostatních případech je element normálně obtékán uvnitř příslušného slotu.

18.2 Záložkové zobrazení – tabbed display

display-model: "stack"

display-role: "card" | "tab"

display: "stack" | "card" | "tab"

tab-side: "top" | "bottom" | "left" | "right"

Element, který má *display-model: stack* zobrazuje pouze jedno ze svých potomků (kteří mají *display-role: card*). Uživatel může interaktivně měnit, který potomek bude zobrazen. Implicitně to bude první. Potomci, kteří mají *display-role: tab* budou zobrazení podél horní části (nebo po straně).

Interaktivní prohlížeč by měl uživateli dovolit vybrat jakoukoli kartu (např. po klepáním). Také by měl být schopen umožnit uživateli vybrat kartu, která nemá záložku, ale to není nutné.

Následující dokument zobrazuje tři záložky a k nim tři odpovídající karty. Uživatel si může vybrat, kterou chce ukázat:

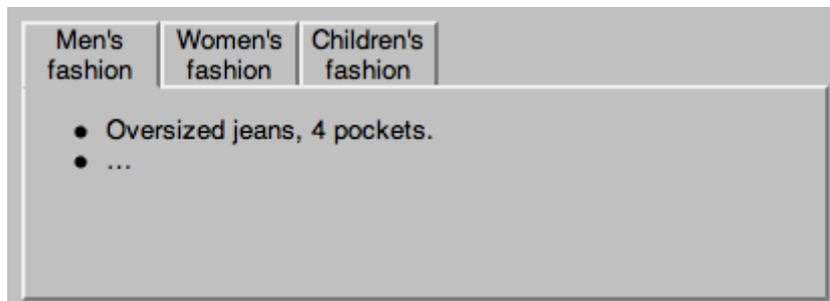
```
<style type="text/css">
  body {background: silver; color: black}
  div.records {display: stack; border: outset}
  div.record {display: card}
  h2 {display: tab; width: 5em; border: outset; text-align: center}
  h2:current {border-bottom: solid silver}
</style>
<div class=records>
  <div class=record>
    <h2>Men's fashion</h2>
    <ul>
      <li>Oversized jeans, 4 pockets.
      <li>...
    </ul>
  </div>
  <div class=record>
    <h2>Women's fashion</h2>
    ...
  </div>
  <div class=record>
    <h2>Children's fashion</h2>
```

```

...
</div>
</div>

```

Výsledek by měl vypadat asi takto:



Obrázek 12 - Možné vykreslení příkladu

Display: stack je zkrácený zápis pro *display-role: block* a *display-model: stack*.

Display: card je zkrácený zápis pro *display-role: card* a *display-model: block-inside*.

Display: tab je zkrácený zápis pro *display-role: tab* a *display-model: block-inside*.

Element, který má *display: stack* má dvě části, které jsou vrstveny jako dva boxy s *display-role: block* uvnitř boxu, která má *display-model: block-inside*. První část má *display-model: inline-inside*, druhý má *display-model: block-inside*. Druhá část zajišťuje řazení a formátování kontextu pro potomky elementů. Druhá část má výšku a šířku rovnu maximu všech ze všech potomků. Do tohoto bloku jsou umístěni všichni potomci

Všechny asociované záložkové elementy budou umístěny do první části jako by měly *display-role: inline*. Jejich pořadí je shodné s pořadím v dokumentu.

18.2.1 Anonymní karty a zásobníky

Pokud má nějaký potomek *display-role* jinou hodnotu než *none* nebo *card*, pak je zabalen do anonymního boxu s vlastností *display-role: card*. Dalšími vlastnostmi anonymních boxů jsou dědičnost, nebo že mají implicitní hodnoty.

Pokud nějaká sekvence jednoho nebo více sourozenců elementů s vlastností *display-role: card* nebo *display-role: none* má rodiče s jiným *display-modelem* než

stack, pak je tato sekvence zabalena do anonymního boxu. Tento anonymní box má *display-model: stack*.

18.2.2 Zálůžky

Pokud je element první záložkou uvnitř karty (ve zdroji dokumentu), pak je element *associated tab* svého nejbližšího předka, který je typu *card*. *Associated tab* element má vypočtenou hodnotu pro vlastnost *display-role: tab*. Pokud není záložka přiřazena k žádné kartě, pak je vypočtená hodnota *display-role: inline*.

Vlastnost *appearance* má také hodnotu *tab*. Můžeme použít obě dvě vlastnosti najednou (*display: tab, appearance: tab*), abychom zajistili, že element, který slouží jako záložka také jako záložka vypadá. Vlastnosti jsou nezávislé a mohou se použít každá zvlášť.

18.2.3 :active a :current záložky

Pokud uživatel aktivuje element, aby byl zobrazen, bude vybrána záložka shodná s pseudotřídou *:active*. Tento stav trvá tak dlouho, dokud uživatel drží tlačítko myši nebo klávesu.

Associated tab právě zobrazené záložky je element, který se shoduje s pseudotřídou *:current*.

18.3 Řádkové a sloupcové rozvržení stránky

Tento layout je alternativou k předchozímu. Nejspíše oba nejsou potřeba. Možná budou nejlepší části zkombinovány na layout, který je nezávislý na struktuře dokumentu a na layout, který se bude měnit při změně rozlišení.

18.3.1 Vlastnost "box-orient"

- Vlastnost nastavuje šířku sloupce.
- Vlastnost lze aplikovat na box elementy.
- Vlastnost není dědičná.
- Hodnoty: *"horizontal"* | *"vertical"* | *"inline-axis"* | *"block-axis"*.
- Implicitní hodnota: *inline-axis*.

18.3.2 Vlastnost "box-direction"

- Vlastnost nastavuje šířku sloupce.
- Vlastnost lze aplikovat na box elementy.
- Vlastnost je dědičná.
- Hodnoty: *"normal"* | *"reverse"*.
- Implicitní hodnota: *normal*.

Příklad:

```
<box>  
  <button style="width: 200px">Child 1</button>  
  <button style="width: 100px">Child 2</button>  
</box>
```

Tento kód vytvoří dvě tlačítka v jedné řádce dle dané šířky.

18.3.3 Vlastnost "box-sizing"

- Vlastnost nastavuje šířku sloupce.
- Vlastnost lze aplikovat na box a blokové elementy.
- Vlastnost není dědičná.
- Hodnoty: *"content-box"* | *"padding-box"* | *"border-box"* | *"margin-box"*.
- Implicitní hodnota: *content-box*.

18.3.4 Vlastnost "box-align"

- Vlastnost nastavuje šířku sloupce.
- Vlastnost lze aplikovat na box elementy.
- Vlastnost není dědičná.
- Hodnoty: *"start"* | *"end"* | *"center"* | *"baseline"* | *"stretch"*.
- Implicitní hodnota: *stretch*.

18.3.5 Vlastnost "box-flex"

- Vlastnost nastavuje šířku sloupce.
- Vlastnost lze aplikovat na potomky box elementů.
- Vlastnost není dědičná.
- Hodnoty: *<number>*.
- Implicitní hodnota: *0.0*.

18.3.6 Vlastnost "box-pack"

- Vlastnost nastavuje šířku sloupce.
- Vlastnost lze aplikovat na box elementy.
- Vlastnost není dědičná.
- Hodnoty: *"start"* | *"end"* | *"center"* | *"justify"*.
- Implicitní hodnota: *start*.

Příklad:

```
<box>
  <button>1. dítě</button>
  <button>2. dítě</button>
  <button>3. dítě</button>
</box>
box {display: box; box-pack: center}
```

Tři tlačítka, která budou uprostřed, těsně vedle sebe.

```
box {display: ".abc." * intrinsic intrinsic intrinsic *}
button {position: a}
```

```
button + button {position: b}
button + button + button {position: c}
```

Tři tlačítka pomocí šablony. Budou uprostřed, těsně vedle sebe s prostorem před a za tlačítka.

Příklad: Za pomoci vlastnosti `box-flex` lze vytvořit boxy, které budou růst různě rychle.

```
<box>
  <button id=b1>Cat</button>
  <button id=b2>Piranha</button>
  <button id=b3>Canary</button>
</box>
with style:
box {display: box; box-orient: vertical}
#b1 {box-flex: 1}
#b2 {box-flex: 2}
#b3 {box-flex: 3}
```

18.3.7 Vlastnost "box-flex-group"

- Vlastnost nastavuje šířku sloupce.
- Vlastnost lze aplikovat na potomky box elementů.
- Vlastnost není dědičná.
- Hodnoty: *<integer>*.
- Implicitní hodnota: *1*.

18.3.8 Vlastnost "box-lines"

- Vlastnost nastavuje šířku sloupce.
- Vlastnost lze aplikovat na potomky box elementů.
- Vlastnost není dědičná.
- Hodnoty: *<integer>*.
- Implicitní hodnota: *1*.

19 Základní uživatelské rozhraní

Tento modul popisuje selektory a vlastnosti, které umožní autorovi upravovat uživatelské rozhraní dle států, základních prvků, vlastností a hodnot.

19.1 Pseudotřídy

Pseudotřídy *:active*, *:hover*, *:focus*, *:enabled*, *:disabled*, *:checked*, *:intermediate* jsou definovány v sekci selektory.

:default – je aplikováno na jeden nebo více elementů, který je výchozí pro sadu podobných elementů. Typicky je aplikováno na menu, tlačítka, seznamy, apod.

:valid, *:invalid* – je aplikováno vzhledem k sémantice dat.

:in-range, *:out-of-range* – je aplikováno na data, která mají omezený rozsah hodnot.

:required, *:optional* – je aplikováno na elementy formuláře, pokud jsou elementy povinné nebo volitelné při potvrzení formuláře.

:read-only – je aplikováno na element, který není dovoleno upravit.

:read-write – je aplikováno na element, který je povoleno upravovat.

19.2 Pseudoelementy

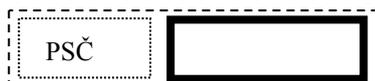
::value – prvek formuláře může obsahovat jak popis, tak hodnotu. Tento pseudoelement reprezentuje hodnotu.

```
<input>
  <label>PSČ<label>
  <input::value/>
</input>
```

pseudo značkovací jazyk

```
input { border:dashed }
label { border:dotted }
input::value { border:solid }
```

CSS



::choices – prvek formuláře, který reprezentuje seznam možností. Seznam radio butonů může být vybrán pseudoelementem **::choices**, zvolená možnost pak pomocí pseudoelementu **::value**.

::repeat-item – reprezentuje jeden prvek z opakující se sekvence.

::repeat-index – reprezentuje aktuálně vybraná prvek z opakující se sekvence.

19.3 Systémový vzhled

CSS2 představil koncept systémových barev, ve kterých může autor pomocí klíčových slov přizpůsobit barvy uživatelského rozhraní. Nicméně barva není jedinou vlastností, která má nativně výchozí hodnoty.

icon – ikona.

window – okno.

desktop, workspace, document, tooltip, dialog.

button – Tlačítko.

push-button, hyperlink, radio-button, checkbox, menu-item, tab.

menu – skupina možností, ze kterých si uživatel může vybrat.

menubar, pull-down-menu, pop-up-menu, list-menu, radio-group, checkbox-group, outline-tree, range

field – pole, které může uživatel měnit.

combo-box, signature, password

19.3.1 Vlastnost "appearance"

- Souhrnná vlastnost pro vlastnosti *appearance*, *color*, *font* a *cursor*.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *"normal"* | *<appearance>* (některá hodnota z výše uvedených) | *"inherit"*.
- Implicitní hodnota: *normal*.

19.4 Vlastnost "icon"

- Vlastnost umožňuje autorovi upravit jakýkoliv element podobný ikoně.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *"auto"* (prohlížeč použije výchozí ikonu) | *<uri>* (odkaz na ikonu) | *"inherit"*
- Implicitní hodnota: *auto*.

Příklad:

```
img,object { content:icon }  
img { icon:url(imgicon.png); }  
object { icon:url(objicon.png); }
```

19.5 Vlastnost "outline-width"

- Vlastnost nastavuje šířku zvýraznění.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *<border-width>* | *"inherit"*.
- Implicitní hodnota: *medium*.
- Funkčnost: FireFox 2.0
 Opera 9.2

19.6 Vlastnost "outline-style"

- Vlastnost nastavuje styl zvýraznění.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *"auto"* | *<border-style>* | *"inherit"*.
- Implicitní hodnota: *none*.
- Funkčnost: FireFox 2.0
 Opera 9.2

19.7 Vlastnost "outline-color"

- Vlastnost nastavuje barvu zvýraznění.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *<color>* | *"invert"* (opačná barva než má element) | *"inherit"*.
- Implicitní hodnota: *invert*.
- Funkčnost: FireFox 2.0
 Opera 9.2

19.8 Vlastnost "outline"

- Vlastnost umožňuje autorovi zvýraznit element orámováním. Na rozdíl od rámečku nezabírá žádný prostor a nemusí být obdélníkový.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *<outline-color>* || *<outline-style>* || *<outline-width>* | *"inherit"*.
- Implicitní hodnota: *dle jednotlivých vlastností*.
- Funkčnost: FireFox 2.0
 Opera 9.2

Příklad:

```
:active { outline: thick solid red }
```

Když je element aktivován, pak je kolem něj vykresleno tlusté, plné, červené zvýraznění.

19.9 Vlastnost "outline-offset"

- Vlastnost nastavuje posunutí zvýraznění za okraj rámečku.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *<length>* | "inherit".
- Implicitní hodnota: 0.
- Funkčnost: FireFox 2.0

Příklad:

```
:focus, :active { outline-offset: 2px }
```

19.10 Vlastnost "resize"

- Vlastnost nastavuje, zda a jak půjde změnit velikost elementu.
- Vlastnost lze aplikovat na elementy, kromě elementů s vlastností *overflow: visible*.
- Vlastnost není dědičná.
- Hodnoty: "none" | "both" | "horizontal" | "vertical" | "inherit".
- Implicitní hodnota: 0.

19.11 Vlastnost "cursor"

- Vlastnost nastavuje vzhled kurzoru.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: `<uri> [<x> <y>] | "auto" | "default" | "none" | "context-menu" | "help" | "pointer" | "progress" | "wait" | "cell" | "crosshair" | "text" | "vertical-text" | "alias" | "copy" | "move" | "no-drop" | "not-allowed" | "e-resize" | "n-resize" | "ne-resize" | "nw-resize" | "s-resize" | "se-resize" | "sw-resize" | "w-resize" | "ew-resize" | "ns-resize" | "nesw-resize" | "nwse-resize" | "col-resize" | "row-resize" | "all-scroll" | "inherit".`
- Implicitní hodnota: *auto*.
- Funkčnost: Firefox 2.0
Opera 9.2
Internet Explorer 7.0

19.12 Vlastnost "nav-index"

- Vlastnost nastavuje pořadí při přechodu na další prvek pomocí tabulátoru.
- Vlastnost lze aplikovat na všechny aktivované elementy.
- Vlastnost není dědičná.
- Hodnoty: `"auto" | <number> (nezáporné číslo určující pořadí) | "inherit"`
- Implicitní hodnota: *auto*.

19.13 Vlastnost "nav-up"

- Vlastnost nastavuje prvek při stisku směrového tlačítka nahoru.
- Vlastnost lze aplikovat na všechny aktivované elementy.
- Vlastnost není dědičná.
- Hodnoty: `"auto" | <id> ["current" | "root" | <target-name>] | "inherit"`
- Implicitní hodnota: *auto*.

19.14 Vlastnost "nav-right"

- Vlastnost nastavuje prvek při stisku směrového tlačítka doprava.
- Vlastnost lze aplikovat na všechny aktivované elementy.
- Vlastnost není dědičná.
- Hodnoty: *"auto"* | *<id>* [*"current"* | *"root"* | *<target-name>*] | *"inherit"*
- Implicitní hodnota: *auto*.

19.15 Vlastnost "nav-down"

- Vlastnost nastavuje prvek při stisku směrového tlačítka dolů.
- Vlastnost lze aplikovat na všechny aktivované elementy.
- Vlastnost není dědičná.
- Hodnoty: *"auto"* | *<id>* [*"current"* | *"root"* | *<target-name>*] | *"inherit"*
- Implicitní hodnota: *auto*.

19.16 Vlastnost "nav-left"

- Vlastnost nastavuje prvek při stisku směrového tlačítka doleva.
- Vlastnost lze aplikovat na všechny aktivované elementy.
- Vlastnost není dědičná.
- Hodnoty: *"auto"* | *<id>* [*"current"* | *"root"* | *<target-name>*] | *"inherit"*
- Implicitní hodnota: *auto*.

Příklad:

```
button { position:absolute }
button#b1 {
    top:0; left:50%;
    nav-index:1;
    nav-right:#b2; nav-left:#b4;
    nav-down:#b2; nav-up:#b4;
}
```

```
button#b2 {
    top:50%; left:100%;
    nav-index:2;
    nav-right:#b3; nav-left:#b1;
    nav-down:#b3; nav-up:#b1;
}
button#b3 {
    top:100%; left:50%;
    nav-index:3;
    nav-right:#b4; nav-left:#b2;
    nav-down:#b4; nav-up:#b2;
}
button#b4 {
    top:50%; left:0;
    nav-index:4;
    nav-right:#b1; nav-left:#b3;
    nav-down:#b1; nav-up:#b3;
}
```

Tlačítka ve tvaru diamantu. Lze se po nich pohybovat proti směru hodinových ručiček nebo přímo pomocí navigačních tlačítek.

20 Vícesloupcové uspořádání

Tento modul popisuje, jak vytvořit pomocí CSS vytvořit vícesloupcové uspořádání. Hlavním přínosem tohoto modulu je jeho flexibilita. Obsah může přetékat z jednoho sloupce do jiného a počet sloupců se může lišit v závislosti rozlišení obrazovky.

20.1 Vlastnost "column-width"

- Vlastnost nastavuje šířku sloupce.
- Vlastnost lze aplikovat na blokové elementy.
- Vlastnost není dědičná.
- Hodnoty: <length> (šířka sloupce) | "auto".
- Implicitní hodnota: *auto*.

Poznámka: column-width definuje ideální šířku, tzn. že hodnota může být změněna, aby vyplnila zadaný prostor.

Příklad:

```
div { width: 100px; column-width: 45px }
```

Máme nevyplněný prostor, proto bude *column-width* nastaveno na 50px.

```
div { width: 40px; column-width: 45px }
```

Prostor je menší, proto bude *column-width* nastaveno na 40px.

20.2 Vlastnost "column-count"

- Vlastnost nastavuje počet sloupců.
- Vlastnost lze aplikovat na blokové elementy.
- Vlastnost není dědičná.
- Hodnoty: <integer> (počet sloupců) | "auto".
- Implicitní hodnota: *auto*.

Poznámka: Pokud bude počet sloupců jeden, pak bude obsah formátován, jako by vlastnost nebyla použita.

Příklad:

```
body { column-count: 3 }
```

20.3 Vlastnost "columns"

- Souhrnná vlastnost pro nastavení sloupců.
- Vlastnost lze aplikovat na blokové elementy.
- Vlastnost není dědičná.
- Hodnoty: `<column-count> || <column-width>`.
- Implicitní hodnota: *dle jednotlivých vlastností*.

Příklad:

```
body { columns: 3 15em }
```

20.4 Vlastnost "column-gap"

- Vlastnost nastavuje mezeru mezi sloupci.
- Vlastnost lze aplikovat na blokové elementy.
- Vlastnost není dědičná.
- Hodnoty: `<length> | "normal"` (hodnotu určí prohlížeč, doporučená hodnota je 1em)
- Implicitní hodnota: *normal*.

20.5 Vlastnost "column-rule-color"

- Vlastnost nastavuje barvu linky mezi sloupci.
- Vlastnost lze aplikovat na blokové elementy.
- Vlastnost není dědičná.
- Hodnoty: `<color> | "transparent"`.
- Implicitní hodnota: *currentcolor*.

20.6 Vlastnost "column-rule-style"

- Vlastnost nastavuje styl linky mezi sloupci.
- Vlastnost lze aplikovat na blokové elementy.
- Vlastnost není dědičná.
- Hodnoty: `<border-style>` ("*none*" (bez linky) | "*hidden*" (skrytá) | "*dotted*" (tečkovaná) | "*dashed*" (čárkovaná) | "*solid*" (plná) | "*double*" (dvojitá plná) | "*groove*" (rytina) | "*ridge*" (reliéf) | "*inset*" (zapuštěný vzhled) | "*outset*" (vystoupilý vzhled).
- Implicitní hodnota: *none*.

20.7 Vlastnost "column-rule-width"

- Vlastnost nastavuje šířku linky mezi sloupci.
- Vlastnost lze aplikovat na blokové elementy.
- Vlastnost není dědičná.
- Hodnoty: `<border-width>` ("*thin*" (úzká) | "*medium*" (střední) | "*thick*" (široká) | `<length>`).
- Implicitní hodnota: *medium*.

20.8 Vlastnost "column-rule"

- Souhrnná vlastnost pro nastavení linky mezi sloupci.
- Vlastnost lze aplikovat na blokové elementy.
- Vlastnost není dědičná.
- Hodnoty: `<border-width>` | `<border-style>` | `<color>`.
- Implicitní hodnota: *dle jednotlivých vlastností*.

20.9 Vlastnost "column-break-before"

- Vlastnost nastavuje zalamování před sloupcem.
- Vlastnost lze aplikovat na blokové elementy.
- Vlastnost není dědičná.
- Hodnoty: *"auto"* (nastaví prohlížeč) | *"always"* (vynucené zalomení) | *"avoid"* (zakázané zalomení).
- Implicitní hodnota: *auto*.

20.10 Vlastnost "column-break-after"

- Vlastnost nastavuje zalamování za sloupcem.
- Vlastnost lze aplikovat na blokové elementy.
- Vlastnost není dědičná.
- Hodnoty: *"auto"* (nastaví prohlížeč) | *"always"* (vynucené zalomení) | *"avoid"* (zakázané zalomení).
- Implicitní hodnota: *auto*.

20.11 Vlastnost "column-break-inside"

- Vlastnost nastavuje zalamování mezi sloupci.
- Vlastnost lze aplikovat na blokové elementy.
- Vlastnost není dědičná.
- Hodnoty: *"auto"* (nastaví prohlížeč) | *"avoid"* (zakázané zalomení).
- Implicitní hodnota: *auto*.

21 Ruby

Ruby je obecně používaný název pro běh textu, který se objevuje v bezprostřední blízkosti jiného toku textu. Vztahuje se k základně a slouží jako poznámka nebo průvodce výslovností spojený s tokem textu. Tento modul právě popisuje použití ruby. Nebudu zde ale popisovat, či vysvětlovat chování a používání ruby. Tak jako v mnoha předešlých modulech (především text a font), je to otázka profesionální sazby. A vysvětlování tohoto není mým úkolem. Já pouze uvedu vlastnosti, které lze použít a jaký to bude mít vliv na zobrazení. Ti kdo chtějí vědět něco více, musí nastudovat publikace týkající se profesionální sazby a zde se dozví, jak je použít při tvorbě internetových stránek.

21.1 Vlastnost "ruby-position"

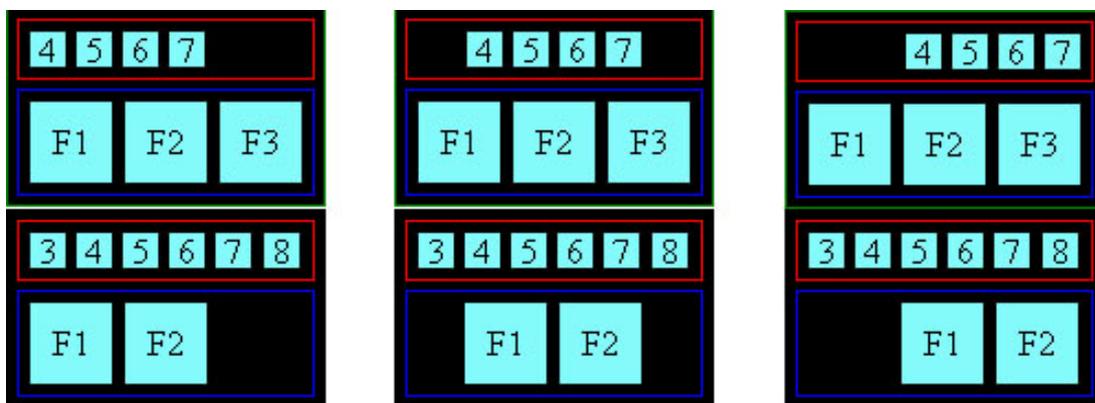
- Vlastnost je používána rodičovskými prvky se zobrazením *ruby-text*, kontrolující pozici ruby textu.
- Vlastnost lze aplikovat na rodičovské elementy s vlastností *display: ruby-text*.
- Vlastnost je dědičná.
- Hodnoty: *"before"* (ruby text se objevuje před základnou psaného text.) | *"after"* (ruby text se objeví až za základnou daného textu.) | *"right"* (ruby text se objeví napravo od základny napsaného textu. Na rozdíl od hodnot *before* a *after* se tato hodnota nevztahuje ke směru toku textu).
- Implicitní hodnota: *before*.

21.2 Vlastnost "ruby-align"

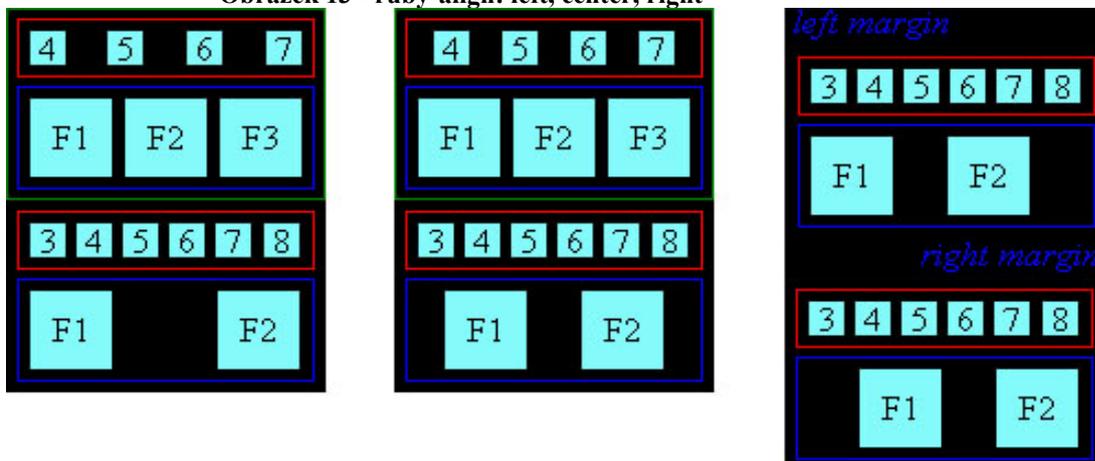
- Vlastnost kontroluje tok textu.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"auto"* (zarovnání určí prohlížeč) | *"start"* | *"left"* (zarovnání obsahu se začátkem okraje) | *"center"* (obsah je vycentrován v šíři základny) | *"end"* |

"right" (zarovnání obsahu ke konci okraje) | "distribute-letter" (jestliže je obsah menší než základna, pak je obsah rovnoměrně rozložen přes šíři základny. První a poslední znak bude na konci základny) | "distribute-space" (jestliže je obsah menší než základna, pak je obsah rozložen přes šíři základny s odsazením od konce základny.) | "line-edge" (jestliže ruby text nepřiléhá k okraji, je obsah zarovnán stejně jako při hodnotě *auto*. Jestliže přiléhá k okraji, pak je obsah stále uspořádán jako při *auto*, ale jestliže strana se dotýká konce linky, je obsah seřazen s odpovídajícím okrajem základny.).

- Implicitní hodnota: *auto*.



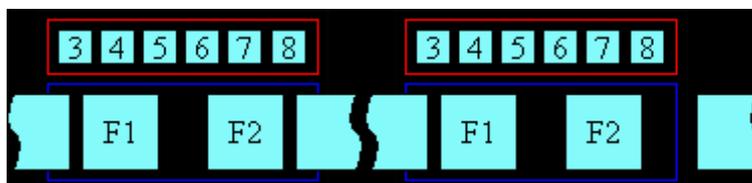
Obrázek 13 - ruby-align: left, center, right



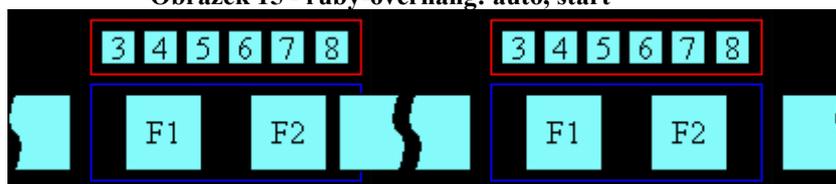
Obrázek 14 - ruby-align: distribute-letter, distribute-space, line-edge

21.3 Vlastnost "ruby-overhang"

- Vlastnost nastavuje, na které straně bude ruby text přečnívat.
- Vlastnost lze aplikovat na rodičovské elementy s vlastností *display: ruby-text*.
- Vlastnost je dědičná.
- Hodnoty: *"auto"* (ruby text může přesahovat text k základně na obou stranách) | *"start"* (může přesahovat ruby text, který textu předchází) | *"end"* (může přesahovat text, který následuje) | *"none"* (ruby text nesmí přesahovat mimo základnu).
- Implicitní hodnota: *none*.



Obrázek 15 - ruby-overhang: auto, start



Obrázek 16 - ruby-overhang: end, none

21.4 Vlastnost "ruby-span"

- Vlastnost kontroluje chování rozpětí anotačních elementů.
- Vlastnost lze aplikovat na rodičovské elementy s vlastností *display: ruby-text*.
- Vlastnost není dědičná.
- Hodnoty: *attr(x)* (parametr *x* je typu string. Funkce vrací typ *<number>*, určující počet prvků základny ruby, které jsou zahrnuty do prvků poznámek.) | *"none"* (neurčuje žádné zahrnutí výčtu).
- Implicitní hodnota: *none*.

Příklad: Následující příklad ukazuje ukázkou XML s použitím vlastnosti display, hodnot asociovaných se strukturou ruby a vlastností span-ruby.

```
myruby, myrb { display: ruby-base; }
myrbc, myrtc { display: ruby-base-container; }
myrt { display: ruby-text; ruby-span: attr(rbspan); }
...
<myruby>
<myrbc>
<myrb>07</myrb>
<myrb>07</myrb>
<myrb>2007</myrb>
</myrbc>
<myrtc>
<myrt>Měsíc</myrt>
<myrt>Den</myrt>
<myrt>Rok</myrt>
</myrtc>
<myrtc>
<myrt rbspan="3">Svatba</myrt>
</myrtc>
</myruby>
```

22 Line

Tento modul se zabývá vzhledem řádky uvnitř blokových a inline elementů. Také zahrnuje design základní čáry zarovnání uvnitř každé linky a umístění zapuštěné iniciály. Většina vlastností je úplně nová.

22.1 Vlastnost "text-height"

- Vlastnost pomáhá vkládaným boxům určit rozměr (výška v horizontálním toku).
- Vlastnost lze aplikovat na inline prvky a rodiče elementů se zobrazeným *ruby-text*.
- Vlastnost je dědičná.
- Hodnoty: *"auto"* (rozměr bloku je založen buď na čtverčikovém rozměru, určeným prvkem a vlastností *font-height* hodnota nebo na výšce buňky (rostoucí i klesající) související s velikostí písma prvku vybraného prohlížečem) | *"font-size"* (rozměr bloku závisí na čtverčíku a hodnotě vlastnosti *font-size*) | *"text-size"* (rozměr bloku závisí na výšce buňky a hodnotě vlastnosti *font-size*) | *"max-size"* (rozměr bloku závisí na maximu rozsahu směrem k přednímu okraji a za okrajem boxu, získaného vzhledem k všem potomkům elementů umístěných na stejné lince).
- Implicitní hodnota: *auto*.

22.2 Vlastnost "line-height"

- Vlastnost nastavující výšku řádky textu (tzv. meziřádkový proklad).
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"normal"* (prohlížeč nastaví proklad na základě vlastnosti *font-size*, doporučená hodnota je 1-1,2) | *<number>* (kladné číslo vyjadřující násobek vlastnosti *font-size*) | *<length>* (kladná velikost meziřádkového prokladu) |

`<percentage>` (poměr vůči vlastnosti *font-size*) | *"none"* (stejná hodnota jako má předcházející blokový prvek).

- Implicitní hodnota: *normal*.
- Funkčnost: Firefox 2.0
Opera 9.2
Internet Explorer 7.0

Příklad:

```
div { line-height: 1.2; font-size: 12px }  
div { line-height: 1.2em; font-size: 12px }  
div { line-height: 120%; font-size: 12px }
```

Všechny tři deklarace vyjadřují stejný mezířádkový proklad.

22.3 Line stacking

Line-stacking(skládání hran) je mechanismus, který nejprve určí hrany každého boxu a potom jsou tyto hrany postupně poskládány dohromady ve směru progresu. Zároveň řeší jakékoliv problémy s místem mezi přiléhajícími hranami.

22.3.1 Vlastnost "line-stacking-strategy"

- Vlastnost určuje metodu pro skládání hran uvnitř boxu.
- Vlastnost lze aplikovat na blokové elementy.
- Vlastnost je dědičná.
- Hodnoty: *"inline-line-height"* (velikost hromady je nejmenší hodnota obsahující rozšířený postup bloku, ze všech vkládaných elementů na té lince kdy jsou tyto elementy vhodně uspořádány) | *"block-line-height"* (výška zásobníku odpovídá prvku bloku a hodnotě vlastnosti *line-height*) | *"max-height"* (výška hromady je tak velká jako nejmenší hodnota, která obsahuje rozměr bloku ze všech vkládaných elementů, na té lince, kde jsou tyto elementy vhodně uspořádány) | *"grid-height"* (odpovídá nejmenšímu násobku velikosti bloku, prvku *line-height*, vypočtená hodnota, která může obsahovat postup bloku, ze všech vložených prvků na lince, ve které jsou prvky uspořádány).
- Implicitní hodnota: *inline-line-height*.

22.3.2 Vlastnost "line-stacking-ruby"

- Vlastnost určuje metodu skládání hran pro blokové elementy obsahující poznámku k elementu (prvek se zobrazením: *ruby-textcontainer*).
- Vlastnost lze aplikovat na blokové elementy.
- Vlastnost je dědičná.
- Hodnoty: *"exclude-ruby"* (ruby poznámky jsou ignorovány) | *"include-ruby"* (ruby poznámky prvků jsou při skládání linek na hromadu zvaženy).
- Implicitní hodnota: *exclude-ruby*.

22.3.3 Vlastnost "line-stacking-shift"

- Vlastnost určuje metodu skládání linek pro blokové elementy, obsahující elementy s posunutou základnou.
- Vlastnost lze aplikovat na blokové elementy.
- Vlastnost je dědičná.
- Hodnoty: *"consider-shifts"* (určuje výšku hromady, která zahrnuje přizpůsobenou horní hranu a dolní hranu jakýchkoli znaků, které mají posunutou základní čáru) | *"disregard-shifts"* (určuje výšku zásobníku, zahrnuje neposunutou horní hranu a dolní hranu kterýchkoliv znaků, které mají posunutou základní čáru).
- Implicitní hodnota: *consider-shifts*.

22.3.4 Vlastnost "line-stacking"

- Souhrnná vlastnost pro řazení hran.
- Vlastnost lze aplikovat na blokové elementy.
- Vlastnost je dědičná.
- Hodnoty: *<line-stacking-strategy>* || *<line-stacking-ruby>* || *<line-stacking-shift>*.
- Implicitní hodnota: *consider-shifts*.

22.4 Zarovnání účaří

Zarovnání účaří popisuje zarovnání textového obsahu, které je založeno na informaci obsažené v tabulce písma.

22.4.1 Identifikátory

Zarovnání účaří určuje následující obrázek.



Obrázek 17 - popis jednotlivých výrazů

22.4.2 Vlastnost "dominant-baseline"

- Vlastnost je užívána pro určení nebo znovu určení základní čáry tabulky.
- Vlastnost lze aplikovat na inline elementy.
- Vlastnost není dědičná.
- Hodnoty: *"auto"* | *"use-script"* (použití skriptu) | *"no-change"* (shodné s rodičovským prvkem) | *"reset-size"* (dominantní identifikátor základní čáry, základní čára tabulky a základní čára tabulky, velikost písma se mění podle hodnoty *font-size*) | *"alphabetic"* (dominantní základní linka se nastaví na alfabetskou základnu) | *"hanging"* (dominantní základní linka se nastaví na zavěšenou základnu) | *"ideographic"* (dominantní základní čára identifikátoru je nastaven na ideologickou základní čáru) | *"mathematical"* (dominantní základní linka se nastaví na základnu matematickou) | *"central"* (dominantní základní linka se nastaví na centrální základnu) | *"middle"* (dominantní základní linka se nastaví na střed základny) | *"text-after-edge"* (dominantní základní linka se nastaví na základnu *text-after-edge*) | *"text-before-edge"* (dominantní základní linka se nastaví na základnu *text-before-edge*).
- Implicitní hodnota: *auto*.

22.4.3 Vlastnost "alignment-baseline"

- Vlastnost nastavuje zarovnání inline elementu vůči svému rodiči.
- Vlastnost lze aplikovat na inline elementy.
- Vlastnost není dědičná.
- Hodnoty: *"baseline"* | *"use-script"* | *"before-edge"* | *"text-before-edge"* | *"after-edge"* | *"text-after-edge"* | *"central"* | *"middle"* | *"ideographic"* | *"alphabetic"* | *"hanging"* | *"mathematical"*.
- Implicitní hodnota: *baseline*.

Poznámka: Hodnoty jsou ekvivalentní k předešlé vlastnosti.

22.4.4 Vlastnost "alignment-adjust"

- Vlastnost nastavuje přesnější zarovnání prvků, jako je grafika, které nemají základnu tabulky nebo postrádají požadovanou základní čáru na své základní lince tabulky.
- Vlastnost lze aplikovat na inline elementy.
- Vlastnost není dědičná.
- Hodnoty: *"auto"* | *"baseline"* (bod zarovnání je v průsečíku začátku okraje a dominantní základny prvku) | *"before-edge"* (bod zarovnání je v průsečíku začátku okraje a částí před počátkem okraje) | *"text-before-edge"* (bod zarovnání je v průsečíku začátku okraje a prvkem) | *"middle"* (bod zarovnání je v průsečíku začátku okraje a středu základny daného elementu) | *"central"* (bod zarovnání je v průsečíku začátku okraje a střední základně daného elementu) | *"after-edge"* (bod zarovnání je v průsečíku začátku okraje a za okrajem prvku) | *"text-after-edge"* (bod zarovnání je v průsečíku začátku okraje a základně) | *"ideographic"* (bod zarovnání je v průsečíku začátku okraje a alfabetickou základnou) | *"alphabetic"* (bod zarovnání je v průsečíku začátku okraje a zavěšené základně) | *"hanging"* (bod zarovnání je v průsečíku začátku okraje a ideologické základní čáře) | *"mathematical"* (bod zarovnání je v průsečíku začátku okraje a základně) | *<percentage>* | *<length>*.
- Implicitní hodnota: *auto*.

22.4.5 Vlastnost "baseline-shift"

- Vlastnost umožňuje změnu pozice dominantní základny, relativně k dominantní základně prvku. Posunutý objekt může být dolním či horním indexem.
- Vlastnost lze aplikovat na inline elementy.
- Vlastnost není dědičná.
- Hodnoty: *"baseline"* (nebude použito žádné posunutí) | *"sub"* (dominantní základna je posunuta k defaultní pozici dolního indexu) | *"super"* (dominantní základna je posunuta k defaultní pozici horního indexu) | *<percentage>* (procentuální vyjádření posunutí) | *<length>* (vzdálenost posunutí).
- Implicitní hodnota: *baseline*.

22.4.6 Vlastnost "vertical-align"

- Vlastnost ovlivňuje vertikální pozicování vložených bloků generovaných vložených levelů uvnitř linie boxu. Následující hodnoty mají význam jen tehdy, když respektujeme rodičovské blokové prvky. Jestliže tento prvek generuje anonymní vložené bloky, pak tato vlastnost nemá žádný efekt, pokud nemá žádné předky.
- Vlastnost lze aplikovat na inline elementy a buňky tabulky.
- Vlastnost není dědičná.
- Hodnoty: *"auto"* | *"use-script"* (použití skriptu) | *"baseline"* (zarovnání alfabetské základny prvku, podle rodičovské základny. Když prvek tuto základnu nemá, zarovná se na dolní okraj boxu) | *"sub"* (zarovnání k dolnímu indexu prvku rodičovské základny) | *"super"* (zarovnání k hornímu indexu prvku rodičovské základny) | *"top"* (seřadí část před okrajem rozšířeného vkládaného boxu s předním okrajem linky boxu) | *"text-top"* (seřadí vrchol boxu s předním okrajem rodičovského prvku fontu) | *"central"* (zarovná střední základnu s předním okrajem rodičovského prvku) | *"middle"* (zarovná dominantní základní linku k okraji vloženého elementu) | *"bottom"* (zarovnání zadního okraje rozšířeného boxu se zadním okrajem linky boxu) | *"text-*

bottom" (zarovná vrchol boxu se zadním okrajem rodičovského prvku fontu) | *<percentage>* | *<length>*.

- Implicitní hodnota: *není definována*.
- Funkčnost: Firefox 2.0
Opera 9.2
Internet Explorer 7.0

22.4.7 Vlastnost "inline-box-align"

- Vlastnost určuje, která linka z víceřádkových vložených bloků bude zarovnána k předchozímu nebo následujícímu prvku.
- Vlastnost lze aplikovat na inline elementy.
- Vlastnost není dědičná.
- Hodnoty: *"initial"* (pro zarovnání používá počáteční linku vložených bloků) | *"last"* (pro zarovnání použije poslední linku vloženého bloku) | *<integer>* (vyjadřuje, kolikátá linka se má použít).
- Implicitní hodnota: *last*.

22.5 Initial line and Drop initial

U blokových elementů, které obsahují inline elementy se často první řádka zobrazuje jinak. Například se může použít jiný řez písma, barva, velikost. První řádek může být formátován pomocí pseudoelementu *::first-line*. Lze na něj použít následující vlastnosti: *font*, *color*, *background*, *text-decoration*, *baseline-alignment*, *word-spacing*, *letter-spacing*, *line-height*, *text-shadow*, *clear*.

Zapuštěná iníciała je typografický efekt, který zvýrazňuje první písmeno v odstavci. Úprava iníciały je možná pomocí pseudoelementu *::first-letter*. Lze na něj použít následující vlastnosti: *font*, *color*, *background*, *text-decoration*, *baseline-alignment*, *text-transform*, *letter-spacing*, *line-height*, *text-shadow*, *clear*, *margin*, *padding*, *border*, *drop-initial*.

22.5.1 Vlastnost "drop-initial-value"

- Vlastnost nastavuje zarovnání iniciály.
- Vlastnost lze aplikovat na pseudoelementy `::first-letter`.
- Vlastnost není dědičná.
- Hodnoty: `"initial"` (která posune iniciálu na první řádek) | `<integer>` (vyjadřuje, ke které lince se má iniciála posunout).
- Implicitní hodnota: `initial`.

Poznámka: Hodnota initial je stejná jako číselná hodnota 1.

22.5.2 Vlastnost "drop-initial-size"

- Vlastnost kontroluje zanořování iniciály.
- Vlastnost lze aplikovat na pseudoelementy `::first-letter`.
- Vlastnost není dědičná.
- Hodnoty: `"auto"` (nastavení podle prohlížeče) | `<line>` (posunutí iniciály bude relativní vzhledem k výšce vybrané linky) | `<length>` (délky zapuštění iniciály) | `<percentage>` (procentuální vyjádření zapuštění).
- Implicitní hodnota: `auto`.

22.5.3 Vlastnost "drop-initial-after-align"

- Vlastnost určuje, která seřizovací linka z n-tého line boxu (n je definováno vlastností `drop-initial-value`) je použita na primární spojovací bod s výchozím boxem pro písmena.
- Vlastnost lze aplikovat na pseudoelementy `::first-letter`.
- Vlastnost není dědičná.
- Hodnoty: `<alignment-baseline>`.
- Implicitní hodnota: `auto`.

22.5.4 Vlastnost "drop-initial-after-adjust"

- Vlastnost nastavuje základní zarovnávací bod pro iniciálu.
- Vlastnost lze aplikovat na pseudoelementy `::first-letter`.
- Vlastnost není dědičná.
- Hodnoty: `"central"` (bod zarovnání je průsečíkem konce okraje a centrální základnou) | `"middle"` (bod zarovnání je průsečíkem konce okraje a střední základnou) | `"after-edge"` (bod zarovnání je průsečíkem konce okraje a zadního okraje rozšířeného boxu) | `"text-after-edge"` (bod zarovnání je průsečíkem konce okraje a základny) | `"ideographic"` (bod zarovnání je průsečíkem konce okraje a ideografickou základnou) | `"alphabetic"` (bod zarovnání je průsečíkem konce okraje a alfabetskou základnou) | `"hanging"` (bod zarovnání je průsečíkem konce okraje a zavěšenou základnou) | `"mathematical"` (bod zarovnání je průsečíkem konce okraje a matematickou základnou)
- Implicitní hodnota: `text-after-edge`.

22.5.5 Vlastnost "drop-initial-before-align"

- Vlastnost nastavuje sekundární bod zarovnání pro iniciálu.
- Vlastnost lze aplikovat na pseudoelementy `::first-letter`.
- Vlastnost není dědičná.
- Hodnoty: `"caps-height"` (zarovnání podle velikosti kapitálek) | `<alignment-baseline>`.
- Implicitní hodnota: `auto`.

22.5.6 Vlastnost "drop-initial-before-adjust"

- Vlastnost nastavuje sekundární bod zarovnání pro iniciálu. Má však smysl, jen tehdy, když hodnota vlastnosti `drop-initial-size` je nastavena na `auto`
- Vlastnost lze aplikovat na pseudoelementy `::first-letter`.
- Vlastnost není dědičná.

- Hodnoty: *"before-edge"* (bod zarovnání je v průsečíku začátku okraje a části před počátkem okraje) | *"text-before-edge"* (bod zarovnání je v průsečíku začátku okraje a prvku) | *"middle"* (bod zarovnání je v průsečíku začátku okraje a středu základny daného elementu) | *"central"* (bod zarovnání je v průsečíku začátku okraje a střední základně daného elementu) | *"after-edge"* (bod zarovnání je v průsečíku začátku okraje a za okrajem prvku) | *"text-afteredge"* (bod zarovnání je v průsečíku začátku okraje a základně) | *"alphabetic"* (bod zarovnání je v průsečíku začátku okraje a alfabetickou základnou) | *"hanging"* (bod zarovnání je v průsečíku začátku okraje a zavěšené základně) | *"ideographic"* (bod zarovnání je v průsečíku začátku okraje a ideologické základní čáře) | *"mathematical"* (bod zarovnání je v průsečíku začátku okraje a základně).
- Implicitní hodnota: *auto*.

23 Stránkovaná média

Stránkovaná média (například papír, slide, atd.) se liší od souvislých médií tak, že je obsah rozdělen do několika oddělených nepohyblivých stran. Pro práci se stránkovanými médii jsou tyto možnosti:

- Úprava zalamování stránek.
- Můžeme specifikovat vlastnosti stránky, jako je velikost, orientace, odsazení, vnitřní i vnější okraje
- Záhloví a zápatí je umístěno uvnitř okrajů *margin*.
- Obsah, jako je například čítač stránek, je umístěn uvnitř zápatí nebo záhlaví
- Je možné kontrolovat vdovy a sirotky (typografické termíny pro práci s osamocenými řádkami)

CSS3 definuje *page model*, který určuje, jak má být dokument formátován uvnitř obdélníkové oblasti. Tato oblast se nazývá *page box* a má konečnou šířku a výšku. Často je *page box* stejný jako jeho fyzická reprezentace např. na papíře. CSS3 určuje formátování pro *page box*, ale výslednou reprezentaci šablony je na prohlížeči.

Přestože CSS3 nespecifikuje, jak má prohlížeč převádět *page boxy* na papír, obsahuje určité mechanismy, které říkají prohlížeči, jak má být výsledná stránka velká a jak je orientovaná.

23.1 Page box – pravidlo @page

Page box je speciální CSS box, který má obdélníkový tvar a je připraven pro tisk. Stejně jako ostatní boxy se skládá z vnitřních a vnějších okrajů, odsazení a obsahu. Obsah a okraje mají speciální funkce:

- Oblast dat se nazývá *page area*. Obsah dokumentu je právě v této oblasti. Okraje na této oblasti jsou určeny první stránkou dokumentu.

- Okraje *page boxu* jsou rozděleny na 16 oblastí. Každá oblast má své vlastní okraje, odsazení a obsah. Tyto oblasti se většinou používají pro záhlaví a zápatí.

Vlastnosti *page boxu* jsou určeny uvnitř stránky pomocí souboru pravidel *@page*. Rozdíl od ostatních boxů je ten, že se na ně nevztahují vlastnosti *width* a *height*. Velikost *page boxu* je určena pomocí vlastnosti *size*.

23.1.1 Stránkový model a pojmy.

Pro stránkový model jsou definovány tyto pojmy:

Stránka (*Page sheet*)

Netisknutelná část stránky (*Non-printable area*)

Obsahová část (*Printable area*)

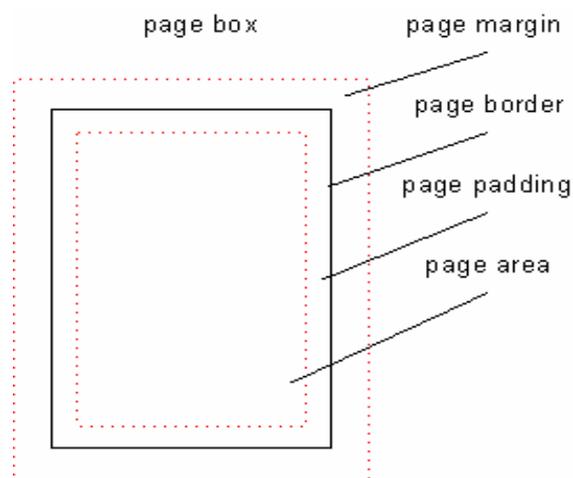
Model stránky (*Page box*)

Obsahová část (*Page area*)

Vnější okraje (*Margin box*)

Vnitřní okraje (*Page padding*)

Okraj boxu (*Page border*)



top-left-corner	
top-left	
top-center	
top-right	
top-right-corner	
left-top	
left-middle	
left-bottom	

Tabulka 8 - Rozdělení margin boxu

23.1.2 Typy stránek

Rozeznáváme různé druhy stránek. Mohou se lišit ve velikosti, rozvržení, orientaci stránek, pořadí tisku, apod. Pro popis stránek se používají následující termíny: Orientace stránky (*Page Orientation*), orientace na výšku (*Portrait Orientation*), orientace na šířku (*Landscape Orientation*), přední strana (*Front Side*), zadní strana (*Back Side*), oboustranný tisk (*Duplex Printing*), přilehlé strany (*Binding Edge treatment*), protilehlé strany (*Facing Pages*), levá stránka (*Left Page*), pravá stránka (*Right Page*), první stránka (*First Page*).

23.1.3 Vlastnost "size"

- Vlastnost určuje velikost a orientaci stránky.
- Vlastnost lze aplikovat na obsah stránky.
- Hodnoty: `<length> {1,2}` (nastavení velikosti stran stránky. První hodnota určuje šířku, druhá výšku. Pokud zadáme pouze jednu hodnotu, pak je šířka rovna výšce) | `"auto"` (velikost i orientaci určí prohlížeč) | [`<page-size>` (zde můžeme použít některou ze standardních velikostí: A5, A4, A3, B5, B4, letter, legal, ledger) || [`"portrait"` | `"landscape"`]]
- Implicitní hodnota: *auto*.

Příklad:

```
@page { size: A4 landscape; }  
stránka velikosti A4 orientovaný na šířku.
```

```
@page { size: 10cm 15cm; }  
stránka o velikosti 10x15 cm
```

23.1.4 Pořadí stránek

Při tisku oboustranných dokumentů můžeme mít různé formátování pro levou a pravou stranu. Toho můžeme dosáhnout pomocí definovaných pseudotříd `:left`, `:right`, `:first`, protože všechny stránky dokumentu do některé třídy patří.

Funkčnost: Opera 9.2

Příklad:

```
@page :left {  
  margin-left: 3cm;  
  margin-right: 4cm;  
}
```

```
@page :right {  
  margin-left: 4cm;  
  margin-right: 3cm;  
}
```

```
@page { margin: 2cm }  
@page :first {  
  margin-top: 10cm }
```

23.2 Margin boxy

Vnější okraje mohou být použity pro umístění záhlaví a zápatí, což jsou části stránky určené pro doplňující informace, jako je například číslo stránky nebo název dokumentu. Umístění záhlaví a zápatí je na rozhodnutí autora dokumentu.

23.2.1 Pravidla pro margin boxy.

At-rule pravidlo pro *margin* se skládá z identifikátoru, který označuje pozici boxu a bloku deklarací.

Příklad:

```
@page {  
  size: 8.5in 11in;  
  margin: 10%;  
  @top-left {  
    content: "CSS Level 3";  
  }  
  @top-right {  
    content: "Page " counter(page);  
  }  
}
```

V levé části záhlaví bude titulek dokumentu "CSS Level 3", v pravé části budou čísla stránek.

23.2.2 Implicitní hodnoty margin boxů

Margin boxy mají implicitně odsazení, vnitřní a vnější okraj nastaveny na nulu. Výchozí výška boxů je rovna příslušnému vnějšímu okraji *page boxu*. Implicitní hodnota pro vlastnost *content* je *none*. Implicitní výška a šířka boxů je *auto*. Výchozí zarovnání je definováno v následující tabulce.

Margin box	'text-align'	'vertical-align'
<u>top-left-corner</u>	right	middle
<u>top-left</u>	left	middle
<u>top-center</u>	center	middle
<u>top-right</u>	right	middle
<u>top-right-corner</u>	left	middle
<u>left-top</u>	center	top
<u>left-middle</u>	center	middle
<u>left-bottom</u>	center	bottom
<u>right-top</u>	center	top
<u>right-middle</u>	center	middle
<u>right-bottom</u>	center	bottom
<u>bottom-left-corner</u>	right	middle
<u>bottom-left</u>	left	middle
<u>bottom-center</u>	center	middle
<u>bottom-right</u>	right	middle
<u>bottom-right-corner</u>	left	middle

Tabulka 9 - zarovnání uvnitř margin boxů

23.3 Page Breaks

Tato část vysvětluje formátování stránkovaných médií. Máme k dispozici pět vlastností, které říkají prohlížeči, jak by měl zalamovat stránky a na které stránce by měl pokračovat. Každé zalomení stránky ponechává rozmístění v aktuálním *page boxu* a zbývající část dokumentu je rozmístěna v *page boxu* nové stránky.

23.3.1 Vlastnost "page-break-before"

- Vlastnost nastavuje zalomení před stránkou.
- Vlastnost lze použít na blokové elementy.
- Vlastnost není dědičná.
- Hodnoty: *"auto"* (záleží na prohlížeči) | *"always"* (vždy se zalamuje) | *"avoid"* (nezalamuje se) | *"left"* (zalomení tak, že je další stránka formátována jako levá) | *"right"* (zalomení tak, že je další stránka formátována jako pravá).
- Implicitní hodnota: *auto*.
- Funkčnost: Opera 9.2

23.3.2 Vlastnost "page-break-after"

- Vlastnost nastavuje zalomení za stránkou.
- Vlastnost lze použít na blokové elementy.
- Vlastnost není dědičná.
- Hodnoty: *"auto"* (záleží na prohlížeči) | *"always"* (vždy se zalamuje) | *"avoid"* (nezalamuje se) | *"left"* (zalomení tak, že je další stránka formátována jako levá) | *"right"* (zalomení tak, že je další stránka formátována jako pravá).
- Implicitní hodnota: *auto*.
- Funkčnost: Opera 9.2

23.3.3 Vlastnost "page-break-inside"

- Vlastnost nastavuje zalomení mezi stránkami.
- Vlastnost lze použít na blokové elementy.
- Vlastnost není dědičná.
- Hodnoty: *"auto"* (záleží na prohlížeči) | *"avoid"* (nezalamuje se).
- Implicitní hodnota: *auto*.
- Funkčnost: Opera 9.2

23.3.4 Vlastnost "page"

- Vlastnost může být použita k určení konkrétního typu stránky, na které by měl být element zobrazen.
- Vlastnost lze použít na blokové elementy.
- Vlastnost je dědičná.
- Hodnoty: *"auto"* | *<identifier>*
- Implicitní hodnota: *auto*.

Příklad:

```
@page rotated { size: landscape }  
table { page: rotated; page-break-before: right }
```

23.3.5 Vlastnost "orphans"

- Vlastnost určuje minimální počet řádků v odstavci, které musí zůstat na konci stránky.
- Vlastnost lze použít na blokové elementy.
- Vlastnost je dědičná.
- Hodnoty: *<integer>*
- Implicitní hodnota: 2.
- Funkčnost: FireFox 2.0
 Opera 9.2
 Internet Explorer 7.0

23.3.6 Vlastnost "widows"

- Vlastnost určuje minimální počet řádků v odstavci, který musí být na začátku stránky.
- Vlastnost lze použít na blokové elementy.
- Vlastnost je dědičná.
- Hodnoty: *<integer>*
- Implicitní hodnota: 2.

- Funkčnost: FireFox 2.0
Opera 9.2
Internet Explorer 7.0

23.4 Vlastnost "image-orientation"

- Tato vlastnost umožňuje pootočit obrázek.
- Vlastnost lze použít na obrázky.
- Hodnoty: *"auto"* | *<angle>* (kladný úhel vyjadřuje otočení ve směru hodinových ručiček, záporný v opačném směru).
- Implicitní hodnota: *auto*.

Příklad:

```
img.tilt      { image-orientation: -1deg }  
...  
* | *<length>* ]{1,2} | [ ["top" | "center" | "bottom"] || ["left" | "center" | "right"] ] ] | "auto" (nastaví zarovnání podle vlastnosti ).
- Hodnoty mají stejný význam jako u *background-position*.
- Implicitní hodnota: *0% 0%*.

## 24 Generovaný a nahrazovaný obsah

Tento modul popisuje, jak vkládat a přesouvat obsah po dokumentu. Vkládaný obsah může také dosadit čítače a řetězce, které mohou být použity pro záhlaví a zápatí, číslování sekcí, seznamů. Nakonec jsou popsány hlavně techniky pro deklaraci nahrazovaných obrázků, ořezávání a změnu velikosti.

### 24.1 Pseudoelementy

Pseudoelementy vytvářejí abstrakci nad dokumentem. Například jazyk dokumentu nenabízí přístup k prvnímu písmenu nebo první řádce. CSS pseudoelementy umožňují autorovi přístup k jinak nepřístupným údajům a způsob, jak přiřadit obsahu přiřadit styl, který v dokumentu neexistuje.

#### 24.1.1 Pseudoelementy `::before`, `::after`

Pseudoelement `::before` používají pro vkládání těsně před obsah elementu nebo před obsah jiného pseudoelementu. Vlastnost `content` se používá pro místo vložení.

Pseudoelement `::after` používají pro vkládání těsně za obsah elementu nebo za obsah jiného pseudoelementu. Vlastnost `content` se používá pro místo vložení.

*Příklad:*

```
abbr { content: attr(title); }
```

Toto pravidlo nahrazuje obsah elementu `<abbr>` obsahem elementu `<title>`.

```
P.pozn:before { content: "Poznámka: " }
```

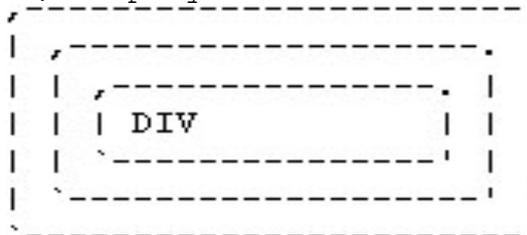
Toto pravidlo vkládá před každý odstavec třídy `pozn` řetězec "Poznámka".

#### 24.1.2 Pseudoelement `::outside`

Pseudoelement `::outside` je generován ihned nad jemu nadřazeným elementem. Pokud zadáme parametr `::outside(n)`, pak vy pseudoelement vyjadřuje n-tý pseudoelement generovaný kolem (n-1) pseudoelementu.

*Příklad:*

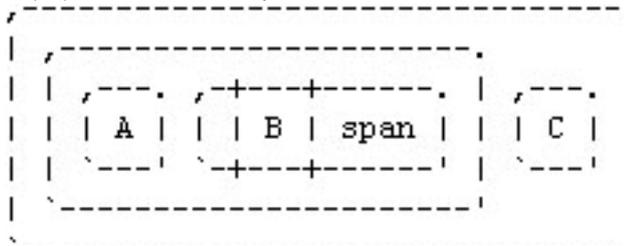
```
div { display: block; border: dashed; }
div::outside { display: block; border: dashed; }
div::outside(2) { display: block; border: dashed; }
```



Obrázek 19 - výsledné zobrazení příkladu

*Příklad:*

```
span { content: "span"; }
span::before { content: "B"; }
span::outside(1) { display: inline; }
span::outside(1)::before { content: "A"; }
span::outside(2) { display: inline; }
span::outside(2)::after { content: "C"; }
```



Obrázek 20 - výsledné zobrazení příkladu

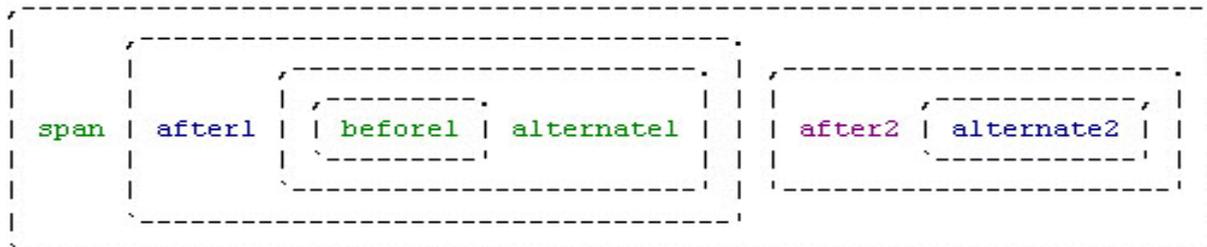
### 24.1.3 Pseudoelement `::alternate`

Tento pseudoelement je vytvořen nastavením vlastnosti *move-to* na nějaký identifikátor. Dědí ze svého asociovaného (pseudo)elementu.

*Příklad:*

```
span { content: "span"; color: green; }
span::alternate { content: "alternate1"; move-to: example1; }
span::alternate::before { content: "before1"; }
span::after { content: "after1" pending(example1); color: navy; }
span::after::alternate { content: "alternate2"; move-to: example2; }
```

```
span::after(2) { content: "after2" pending(example2); color: purple; }
```



Obrázek 21 - výsledné zobrazení příkladu

## 24.2 Vlastnost "move-to"

- Vlastnost způsobuje vyjmutí (pseudo)elementu z dokumentu a jeho pozdější vložení zpět do dokumentu. Obsah je vložen zpět pomocí hodnoty *pending()*.
- Vlastnost lze použít na všechny elementy, *::before*, *::after*, *::alternate*.
- Vlastnost není dědičná.
- Hodnoty: *"normal"* | *"here"* ((pseudo)element není přesunut) | *<identifier>* (element je zobrazen při dalším výskytu vlastnosti *pending(<identifier>)*).
- Implicitní hodnota: *normal*.

## 24.3 Vlastnost "quotes"

- Vlastnost specifikuje značky pro citaci uvnitř citací.
- Vlastnost lze použít na všechny elementy, *::before*, *::after*, *::alternate*, *::marker*, *::line-marker*, oblasti *margin*, oblasti *@footnote*.
- Vlastnost není dědičná.
- Hodnoty: *"foo"* | *"bar"*.
- Implicitní hodnota: *text*.

## 24.4 Čítače

### 24.4.1 Vlastnost "counter-increment"

- Vlastnost nastavuje, kolik a kterému počítadlu přičíst.

- Vlastnost lze použít na všechny elementy, `::before`, `::after`, `::alternate`, `::marker`, `::line-marker`, oblasti `margin`, oblasti `@footnote`, `@page context`.
- Vlastnost není dědičná.
- Hodnoty: [ `<identifier>` (název čítače) `<integer>` (hodnota inkrementace) ] | `"none"`.
- Implicitní hodnota: `none`.
- Funkčnost: Opera 9.2

## 24.4.2 Vlastnost "counter-reset"

- Vlastnost vynuluje čítač.
- Vlastnost lze použít na všechny elementy, `::before`, `::after`, `::alternate`, `::marker`, `::line-marker`, oblasti `margin`, oblasti `@footnote`, `@page context`.
- Vlastnost není dědičná.
- Hodnoty: [ `<identifier>` (název čítače) `<integer>` (nulovací hodnota) ] | `"none"`.
- Implicitní hodnota: `none`.
- Funkčnost: Opera 9.2

*Příklad:*

```
H1:before {
 content: "Kapitola " counter(chapter) ". ";
 counter-increment: chapter;
 counter-reset: section;
}
H2:before {
 content: counter(chapter) "." counter(section) " ";
 counter-increment: section;
}
```

Nastavení číslování sekcí ve tvaru "Kapitola 1", "1.1", "1.2", atd.

## 24.5 Vkládání a nahrazování obsahu pomocí vlastnosti "content"

- Vlastnost určuje, co bude zobrazeno uvnitř (pseudo)elementu.
- Vlastnost lze použít na všechny elementy, ::before, ::after, ::alternate, ::marker, ::line-marker, oblasti margin, oblasti @footnote, @page context.
- Vlastnost není dědičná.
- Hodnoty: [ <uri> ';' ]\* (seznam objektů) ["normal" | "none" | "inhibit" | <content-list> (pending(<identifier>) | <string> | "contents" | "footnote" | "endnote" | "section-note" | "list-item" | <counters> | <strings> | "open-quote" | "close-quote" | "no-open-quote" | "no-close-quote" | <glyph> | <uri> | <date> | "document-url" | <target>)]
- Implicitní hodnota: *normal*.

## 25 Generovaný obsah pro stránkovaná média

Tato specifikace popisuje různé funkce, které se běžně využívají pro papírové publikování. Některé z navrhovaných funkcí (dělení slov, nové styly pro seznamy) by mohly být využity i v jiných typech médií. Přesto se tato specifikace týká pouze stránkovaných médií.

### 25.1 Pojmenované řetězce

Pro lepší navigaci ve v tištěných materiálech se často využívá záhlaví a zápatí, které je vytištěno ve vnějším okraji stránky. Tato metoda nabízí dvě možnosti, jak se záhlavím a zápatím pracovat. Prvním jsou *pojmenované řetězce*, které kopírují text elementu, pro pozdější použití. Druhým je mechanismus pro přemísťování elementů do záhlaví a zápatí.

*Pojmenované řetězce* si můžeme představit jako proměnné, které udržují jeden textový řetězec. Jsou vytvořeny pomocí vlastnosti *string-set*, která do pojmenovaného řetězce hodnotu zkopíruje. Jediný důvod pro používání pojmenovaných stringů je jejich pozdější použití vlastností *content*

#### 25.1.1 Vlastnost "string-set"

- Vlastnost vytvoří dvojici pojmenovaný řetězec a do něj dosazený text.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: [[ <identifier> (název pojmenovaného řetězce) <content-list> (textový obsah) ] [, <identifier> <content-list>]\* ] | "none".
- Implicitní hodnota: *none*.

*Příklad:*

```
@page { @top-center { content: string(header) }}
@page { @right-middle { content: string(index) }}
@page { @top-left { content: string(entry) }}
h1 { string-set: header "Chapter " counter(chapter) content }
dt { string-set: index first-letter, entry content }
```

```
@page { @top-center { content: string(chapter, last-except) }}
h1 { string-set: chapter content }
```

Pokud se na stránce objeví element h1, pak bude buňka top-center prázdná. Na ostatních stránkách bude zobrazen předchozí h1 element.

## 25.2 Spojovací linky

Spojovací linka je vodící čára, pro lepší orientaci v textu. Typicky se používá k seznamu a jemu příslušnému kódu. Například název kapitoly a stránky nebo telefonní seznam a telefonní číslo.

V CSS3 je vodící linka tvořena posloupností znaků, kterou definuje funkce *leader()*. Tato funkce má jako parametr jednu z následujících hodnot: *dotted*, *solid*, *space*, *<string>*.

*Příklad:*

```
heading::after { content: leader(dotted) }
```

## 25.3 Křížové reference

V dokumentech je běžné že se v nějaké části odkazujeme na určitou sekci nebo stránku. Např. "viz sekce 3.2.1" nebo "podívejte se na str. 28". Křížové reference jsou generovány použitím speciálního stylu na odkaz, který ušetří čas a zamezí chybám. Místo běžného formátování může styl specifikovat, že cíl odkazu je straně...

*Příklad:*

```
a::after { content: "(viz. strana " target-counter(attr(href, url), page, decimal) ")" }
```

## 25.4 Hodnoty "target-counter" a "target-counters"

Číselné křížové reference jsou generovány pomocí *target-counter* a *target-counters*, které vypočtou cílovou hodnotu odkazu. Tyto funkce jsou podobné funkcím *counter()* a *counters()*, s tím rozdílem, že počítají hodnoty cílových elementů. *Target-counter* má dva povinné argumenty: odkaz a jméno čítače. *Target-counters* má tři povinné argumenty: odkaz, jméno čítače a řetězec pro oddělovač. Volitelný argument na konci výčtu vyjadřuje styl číslování, výchozí styl je *decimal*.

## 25.5 Hodnota "target-text"

Hodnota *target-text* generuje textovou referenci, která připojuje textový obsah cíle odkazu. Je kopírován pouze text. *Target-text* má jeden povinný argument, který vyjadřuje adresu na odkaz. Druhý volitelný argument specifikuje, jaký obsah má být generován. Jsou to čtyři možné hodnoty: *content*, *before*, *after*, *first-letter*. Výchozí hodnota je *content*.

*Příklad: Text Viz Kapitola 3 ("A better way") na straně 31 ... bychom vygenerovali takto:*

html

```
<p>Viz this chapter ...
...
<h2 id="chx">A better way</h2>
```

CSS:

```
h2 { counter-increment: chapter }
a { content: "Chapter " target-counter(attr(href, url),
chapter)
' (" target-text(attr(href), content) "') na straně
',
target-counter(attr(href, url), page);
```

## 25.6 Poznámky pod čarou

Poznámky pod čarou jsou dalším z častých nástrojů, používaných při tisku. Poznámka pod čarou je komentář nebo citace k textu, umístěna na konci stránky. Odkaz na poznámku pod čarou je volán pomocí *note-call*.

*Příklad:*

```
<style>
.footnote { float: note }
</style>
<p>Věta obsahuje slova často.
```

*Výsledek:*

Věta obsahuje slova <sup>1</sup>

<sup>1</sup> často.

Pro podporu poznámek pod čarou jsou přidány následující možnosti:

***float: footnote*** – nová hodnota vlastnosti *float*

***@footnote*** – nová oblast pro *page area*

***border-length*** – nová vlastnost pro *@footnote*

***::footnote-call, ::footnote-marker***

***footnote*** jako předdefinovanou hodnotu čítače

***content: target-move()*** – nová hodnota pro vlastnost *content*

## 25.6.1 Přeměna elementů na poznámky pod čarou

Element s vlastností *float: footnote* (nazývá se *footnote element*) je přesunutý do oblasti poznámek pod čarou a je nahrazen pseudoelementem, na který se odkazuje.

Pro každý nový *footnote element* je předdefinován čítač *footnote*, který je inkrementován vzhledem k nastavení konkrétní šablony.

*Příklad:*

```
span.footnote {
 float: footnote;
}
```

## 25.6.2 Oblast poznámek pod čarou

Podle specifikace daného stylu jsou všechny elementy s vlastností *float: footnote* jsou přesunuty do boxu s vlastností *content: pending(footnote)*. Toto je případ pro *@footnote area*.

Oblast poznámek pod čarou je umístěna na konci stránky. Elementy *float: bottom* jsou umístěny nad tuto oblast. Pokud na stránce žádná poznámka pod čarou není, pak oblast pro poznámky pod čarou nezabere žádný prostor. Rozmístění a formátování uvnitř oblasti poznámek pod čarou je určený nastavením této oblasti a formátováním prvků samotných.

Na oblast poznámek pod čarou lze použít tyto vlastnosti: *border, padding, margin, background, column-span, border-length*.

*Příklad:*

```
@footnote { content: pending(footnote) }
```

Část navrhované výchozí šablony.

```
@footnote {
 margin-top: 0.5em;
 border-top: thin solid black;
 border-length: 4em;
 padding-top: 0.5em;
 column-span: all;
}
```

Příklad využívající možnosti formátování.

### 25.6.3 Vlastnost "border-length"

- Vlastnost nastavuje velikost horizontálního oddělovače poznámek pod čarou.
- Vlastnost lze aplikovat na oblasti poznámek pod čarou.
- Vlastnost není dědičná.
- Hodnoty: *<length>* (délka horizontálního oddělovače) | *"auto"*.
- Implicitní hodnota: *auto*.

### 25.6.4 Volání poznámek pod čarou

Když je element přesunut do oblasti poznámek pod čarou, zanechá na svém původním místě volání odkazu. Obsah a formátování je nastaveno pomocí pseudoelementu *footnote-call*.

*Příklad:*

```
span.footnote::footnote-call {
 content: counter(footnote, super-decimal)
}
```

### 25.6.5 Značky

Ke každému elementu typu poznámka pod čarou je přidána pseudoelement *::note-marker*. Obvykle obsahuje stejná čísla nebo symboly jako jeho volání, aby bylo vidět, že k sobě patří.

*Příklad:*

```
.footnote::footnote-marker {
 content: counter(footnote, super-decimal);
 list-style-position: outside;
}
```

### 25.6.6 Čítač poznámek pod čarou

Poznámky pod čarou mají předdefinovaný čítač *footnote*. Pokud je přidán element s vlastností *position: footnote*, pak je čítač automaticky inkrementován. Element s vlastností *float: footnote* dědí od svého rodiče, nikoli od *@footnote*.

*Příklad:*

```
@page { reset-counter: footnote }
```

### 25.6.7 Hodnota "target-move"

K tomu, abychom mohli přivést obsah jiného elementu (např. odkaz na poznámku pod čarou), je zavedena hodnota *target-move*. Tato funkce vyžaduje jako argument hodnotu *URI*. V kombinaci s vlastností *content*, jsou cílové elementy přesunuty k pevnému elementu.

## 25.7 Poznámky na konec

Poznámky na konec jsou podobné poznámkám pod čarou, s tím rozdílem, že jsou umístěny až na konci sekce. Některé dokumenty užívají obou typů poznámek. Pro tyto poznámky by měla být přidána nová hodnota pro vlastnost *float*.

Elementy s vlastností *float: endnote* jsou přesunuty na místo definované vlastností *content: pending(endnote)*. Pokud takové místo není specifikováno, pak budou elementy přesunuty na konec dokumentu.

*Příklad:*

```
span.note {
 float: endnote;
}
body::after { content: pending(note) }

div.chapter::after {
 content: pending(endnote);
 display: block;
}
```

## 25.8 Running elements

Záhlaví a zápatí jde vytvořit pomocí pojmenovaných řetězců. Udrží ovšem pouze textovou informaci. Styly a struktura jsou ignorovány. Pro překonání těchto omezení jsou vytvořeny running záhlaví a zápatí.

Na rozdíl od poznámek pod čarou a poznámek na konec jsou tyto elementy přesunuty do oblasti záhlaví a zápatí a opakují se na každé stránce. Pro tyto elementy vznikla nová hodnota vlastnosti *position: running()*. Má jeden povinný argument, který znamená název vzniklého elementu. Running elementy nejsou skutečné elementy a jejich vznik neovlivňuje strukturu dokumentu. Running element udržuje pouze jeden element a jeho potomky. Kdykoliv je running elementu přiřazena nová hodnota, stará je ztracena. Obsah elementu může být získán pomocí hodnoty *element(<element>)*.

*Příklad: Záhloví je zobrazeno pouze při tisku. Záhloví je zobrazeno nahoře uprostřed na všech stranách, kromě těch, kde je h1 element.*

```
<style>
 div.header { display: none }
 @media print {
 div.header {
 display: block;
 position: running(header);
 }
 @page { @top-center { content: element(header, last-
except) }}
</style>
...
<div class="header">Introduction</div>
<h1 class="chapter">An introduction</div>
```

*Příklad: Kód mění running záhloví pouze na jedné straně.*

```
...
<style>
@page { @top-center {
 content: element(header, first) }}
.header { position: running(header) }
.once { font-weight: bold }
</style>
...
<div class="header">Not now</div>
<p>Da di ha di da di ...
 NOW!
 Not now
 ... da di ha di hum.</p>
...
```

## 25.9 Nové styly čítačů

### 25.9.1 list-style-type: super-decimal

Pro lepší podporu poznámek pod čarou byla přidána nová hodnota *super-decimal*. Jsou to malé, horní indexy. První tři mají svůj kód v tabulce Latin-1 a některá písma jich mají i více. Klíčové slovo *super-decimal* umožňuje použít tyto zdroje použít a tím nahradit použitý *font-size* a *vertical-align*.

## 25.9.2 Styly pro čítače

CSS pro vlastnost *list-style-style* a ostatní vhodná místa popisuje několik předdefinovaných stylů. Některé styly opakují stejný znak (např. kolečko nebo kroužek), zatímco jiné používají číselované seznamy (malé římské číslice). Pro zvýšení počtu stylu můžeme použít pravidlo `@counter-style`. Za pomoci tohoto pravidla můžeme pojmenovat nová řešení.

Pravidlo `@counter-style` se skládá z klíčového slova `@counter-style`, jeho jména a seznamu řetězců oddělených mezerou.

*Příklad:*

```
@counter-style daggers "*" "\2020" "\2021" "\A7" "#";
ol { list-style-type: daggers }
@counter-style ordinal "1st" "2nd" "3rd" "4th";
h1:before { content: counter(chapter, ordinal) " chapter"
}
```

## 25.9.3 list-style-type: symbol()

Tento funkční zápis pro definici nového list-style-typu je zaveden, abychom zbytečně nemuseli vytvářet pojmenovaný sty. Funkce *symbol()* vyžaduje, stejně jako pojmenované styly, seznam řetězců oddělených mezerou.

*Příklad:*

```
ol { list-style: " *" "\2020" "\2021" "\A7" "#" }
```

## 25.10 Nové hodnoty pro vlastnost "float"

V CSS2 je vlastnost *float* používána pro posunutí obsahu doleva nebo doprava vůči aktuální lince. Tato specifikace rozšiřuje vlastnost o několik nových hodnot, které umožňují posunout obsah do míst, které jsou běžně používány pro tisk. Několik nových klíčových slov může nebo musí být kombinováno dohromady. Na pořadí slov nezáleží.

*footnote* – element je změněn na typ footnote, toto klíčové slovo nesmí být kombinováno s ostatními.

***endnote*** – element je změněn na typ endnote, toto klíčové slovo nesmí být kombinováno s ostatními.

***inside*** – toto klíčové slovo je synonymem pro *left* na lichých stranách a *right* na sudých stranách.

***outside*** – toto klíčové slovo je synonymem pro *left* na sudých stranách a *right* na lichých stranách.

***page*** – pozměňuje význam *top*, *bottom*, *next* a může být použito pouze v kombinaci s jedním z nich.

***column*** – pozměňuje význam *top*, *bottom*, *next* a může být použito pouze v kombinaci s jedním z nich.

***gap*** – upřesňuje, že je element umístěn do mezery mezi sloupci, toto klíčové slovo má význam pouze v kombinaci s klíčovým slovem *column*.

***top*** – umísťuje element do horní části boxu s obsahem. V kombinaci s *page* je element umístěn do horní části stránky. V kombinaci s *column* je element umístěn do horní části sloupce. V kombinaci s *left*, *right*, *inside*, *outside* je element přenesen na určenou stranu, přičemž umožňuje obsahu být obtékán.

***bottom*** – umísťuje element do dolní části boxu s obsahem. V kombinaci s *page* je element umístěn do dolní části stránky. V kombinaci s *column* je element umístěn do dolní části sloupce. V kombinaci s *left*, *right*, *inside*, *outside* je element přenesen na určenou stranu, přičemž umožňuje obsahu být obtékán.

***next*** – v kombinaci s *page* umísťuje element na další stránku. V kombinaci s *column* je element umístěn do dalšího sloupce.

***unless-room*** – přidá podmínku, že pokud pro element není dostatek místa pro umístění na další stránku nebo sloupec, pak element není přemístěn. Má význam pouze v kombinaci s *next* nebo *dev-null*.

***dev-null*** – tato hodnota v kombinaci s *unless-room* určuje, že je element umístěn na momentální stránku nebo sloupec, pouze pokud je pro něj místo.

*Příklad:*

```
img { float: page bottom }
```

Obrázek je umístěn do dolní části stránky.

## 25.11 Dělení slov

V modulu text je popsána vlastnost *hyphenate*, která umožňuje pomocí stylů nastavovat dělení slov. Tato specifikace popisuje pro bližší určení slov další vlastnosti.

### 25.11.1 Vlastnost "hyphenate-dictionary"

- Vlastnost nastavuje slovník pro dělení slov.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"auto"* | *<uri>* (odkaz na slovník).
- Implicitní hodnota: *auto*.

### 25.11.2 Vlastnost "hyphenate-before"

- Vlastnost nastavuje minimum znaků před oddělovačem.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *<integer>*.
- Implicitní hodnota: 2.

### 25.11.3 Vlastnost "hyphenate-after"

- Vlastnost nastavuje minimum znaků za oddělovačem.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *<integer>*.
- Implicitní hodnota: 2.

### 25.11.4 Vlastnost "hyphenate-lines"

- Vlastnost nastavuje maximum oddělovačů.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *"none"* | *<integer>*.
- Implicitní hodnota: *none*.

### 25.11.5 Vlastnost "hyphenate-character"

- Vlastnost nastavuje řetězec pro oddělovač. Obvykle ho tvoří pouze jeden znak.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost je dědičná.
- Hodnoty: *<string>*.
- Implicitní hodnota: *U+2010*.

## 25.12 Záložky

Některé formáty dokumentů, mohou obsahovat záložky. Pro vytvoření záložek jsou definovány tři vlastnosti.

### 25.12.1 Vlastnost "bookmark-level"

- Vlastnost nastavuje úroveň záložky.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *"none"* | *<integer>*.
- Implicitní hodnota: *none*.

### 25.12.2 Vlastnost "bookmark-label"

- Vlastnost nastavuje jméno záložky.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *"content"* | *<attr>* | *<string>*.
- Implicitní hodnota: *content*.

*Příklad:*

```
a { bookmark-label: attr(title, string) }
h1 { bookmark-label: self }
#frog { bookmark-label: "The green frog";
```

### 25.12.3 Vlastnost "bookmark-target"

- Vlastnost nastavuje cíl záložky.
- Vlastnost lze aplikovat na všechny elementy.
- Vlastnost není dědičná.
- Hodnoty: *"self"* | *<uri>* | *<attr>*.
- Implicitní hodnota: *self*.

*Příklad:*

```
.bookmark {
 bookmark-label: attr(title, string);
 bookmark-target: attr(href, url);
}
...
<a class="bookmark" title="The green pear"
href="#pears"/>
```

## 25.13 Indikace změny v dokumentu

*Change bar* poukazuje na změnu v části dokumentu. Změněná část nemusí nutně odpovídat elementu (z důvodu struktury značkovacích jazyků) a není vždy možné takovýto element vytvořit.

Abychom se vyhnuli těmto omezením, začátek změny se bude týkat prvního dokumentu a znaménko konce změny bude přidruženo k následujícímu dokumentu.

*Příklad:*

```
p.change-start {
 change-bar: thin solid red;
 change-bar-class: changel;
 change-bar-offset: 0.3em;
 change-bar-side: left; /* or right, inside, outside */
}

p.change-end {
 change-bar-class: changel;
}
```

## 25.14 Značky pro pokračování

Značky udávají, že element pokračuje na další stránku. Pseudelement `::before-break` je zobrazen za poslední řádkou před koncem stránky. Pseudelement `::after-break` je zobrazen před první řádkou na nové stránce.

*Příklad:*

```
::before-break {
 content: image(to-arrow.png);
}

::after-break {
 content: image(from-arrow.png);
}
```

## 25.15 Seznam pojmenovaných stránek

V CSS2 vlastnost `page` přijímá pouze jednu hodnotu. Je jí buďto hodnota `auto` nebo `jméno stránky`. V případě, že je specifikovaná konkrétní stránka, je před ní vložen konec stránky a následně je do ní element vložen.

Tato specifikace umožňuje přiřadit vlastnosti `page` seznam hodnot. Obsah je průběžně vkládán do určených stran. Každý prvek na jednu stránku. Pokud je vyžadováno více stran, pak je poslední prvek opakovaný tolikrát, kolik je potřeba.

*Příklad:*

```
h2 { page: no-header auto }
```

Obsah elementu h2 by měl být umístěn na stránce *no-header*, potom je obsah umístěn dle hodnoty *auto*.

## 25.16 Generované seznamy

Většina knih obsahuje sekce, které jsou vytvořeny z hlavního obsah, např. obsah, rejstřík, slovníček pojmů, apod. Právě tyto sekce nazýváme generovanými seznamy. Některé seznamy jsou řazeny abecedně (rejstřík, slovník), jiné jsou řazeny dle výskytu v textu (obsah, seznam tabulek).

Pro tyto generované seznamy musí být vytvořen *prototype container*, do kterého budou přesunuty ostatní elementy, nebo bude mít svůj vlastní obsah.

### 25.16.1 Obsah

*Příklad: Tento příklad ukazuje, jak vygenerovat obsah s vodíci linkami a čísly stránek.*

```
...
<style>
 #toc { prototype: container }
 #toc-entry {
 prototype-insert-position: current;
 font-size: 14pt }
 #toc-entry::after { content: leader('. ') source-
counter(page) }
 h1.chapter { make-element: toc-entry content }
</style>
...
<div id="toc">
 <h1>Table of contents</h1>
 <div id="toc-entry"></div>
</div>
...
<h1 class="chapter">Introduction</h1>
...
```

Jsou zde tři nové vlastnosti a jedna nová hodnota pro vlastnost *content*.

```
#toc { prototype: container }
```

označuje #toc element jako *prototype container*.

```
prototype-insert-position: current;
```

označuje, že #toc zůstává na nynější pozici, tj. hned za předchozím seznamem.

```
#toc-entry::after { content: leader('. ') source-
counter(page) }
```

má novou hodnotu *source-counter(page)*, která přidá číslo stránky.

```
h1.chapter { make-element: toc-entry content }
```

pravidlo vytvoří nový element, který je izomorfní k #toc-entry elementu a je vložen na základě *prototype-insert-position*.

## 25.16.2 Slovník pojmů

*Příklad:*

```
...
<style>
#glossary { prototype: container }
#glossary-term { insert-position: sorted }
#glossary-definition { insert-position: current }
</style>
...
<div id="glossary">
<h2>Glossary of terms</h2>
<dl>
 <dt id="glossary-term">...</dt>
 <dd id="glossary-definition">...</dd>
</dl>
</div>
...
<h1 class="chapter">Introduction</h1>
...

```

Řazení by mělo být zajištěno pomocí hodnot *sorted* a *current*.

## 25.16.3 Rejstřík

Rejstřík je abecedně řazený vygenerovaný seznam, podobný slovníku pojmů. Navíc obsahuje velká písmena, pro lepší orientaci v textu. Např. všechny pojmy začínající písmenem "s" budou pod velkým "S".

Pro dosažení tohoto vzhledu je navržena následující strategie. Pro každý záznam budou vygenerovány dva záznamy. Jeden bude velké počáteční písmeno, druhý bude samotný záznam. Abychom zamezili duplicitám velkých písmen před každým záznamem, nadeklarujeme vlastnost *insert-policy*, která tyto duplicity vymaže.

*Příklad:*

```

...
<style>
#index {
 prototype: container }
#index-marker {
 insert-position: sorted
 insert-policy: unique;
 text-transform: uppercase }
#index-entry {
 insert-position: sorted }
#index-entry::after {
 content: leader(. .) source-counter(page) }
dfn.entry {
 make-element: index-marker first-letter, index-entry
content }
</style>
...
<p>An <dfn>emphasized element</dfn> stands out.</p>
...

```

## 25.16.4 Komplexní použití generovaných seznamů

*Příklad: Tento příklad vytváří několik typů generovaných seznamů.*

```

<style>
#toc-container {
 prototype: container }

#toc-entry-section {
 font-size: large;
 insert-position: current }

#toc-entry-subsection {
 font-size: medium;
 insert-position: current }

#toc-entry-section::after, #toc-entry-subsection::after {
 content: leader('. ') source-counter(page) }

#acronym-list {
 prototype: container }

#acronym-term {
 insert-position: sorted }

```

```
#acronym-definition {
 insert-position: current }

#index {
 prototype: container }

#index-marker {
 insert-position: sorted
 insert-policy: unique;
}

#index-entry {
 insert-position: sorted }

#index-entry::after {
 content: leader(. .) source-counter(page) }

h2 {
 make-element: toc-entry content }

h3 {
 make-element: toc-entry content }

acronym {
 make-element:
 index-marker first-letter,
 index-entry content,
 acronym content,
 acronym-definition attr(title);
}

dfn { make-element:
 index-marker first-letter(),
 index-entry content
}

</style>

<div id="toc">
<h2>Table of contents</h2>
 <div id="toc-container">
 <div id="toc-entry-section"></div>
 <div id="toc-entry-subsection"></div>
 </div>
</div>

<div id="acronym-list">
```

```
<h2>List of acronyms</h2>
<dl>
 <dt id="acronym"></dt>
 <dd id="acronym-definition"></dd>
</dl>
</div>

<div id="index">
<h2>Index</h2>
<div id="index-marker"></div>
<div id="index-entry"></div>
</div>

<body>

<h2>Introduction</h2>

<p>This part defines the a acronym: <acronym title="HyperText
Markup Language">HTML</acronym>.

<h3>More to learn</h3>

<p>An <dfn>emphasized element</dfn> element stands out.

</body>
```

## 26 Závěr

Ve srovnání se staršími specifikacemi přináší CSS level 3 mnoho nového. Stávající vlastnosti jsou upřesněny a rozšířeny o nové hodnoty. Na žádost webdesignerů také přibylo mnoho vlastností zcela nových.

První je určena pro grafickou podobu stránky. Jsou zde různé úpravy písma, textu, obrázků, nové možnosti rozmístění prvků, apod. Tato pravidla využijí především tvůrci webů, které jsou založeny na atraktivní grafice. Např. nadpis se stínem již nebude nutné nahrazovat obrázkem.

Druhá skupina se zabývá sazbou textu a formátováním stránky pro tisk. Můžeme tedy nastavovat velikost stránky, umístění záhlaví a zápatí, mezislovní mezery, meziřádkový proklad, levé a pravé stránky, apod. Možnosti formátování jsou srovnatelné s textovými editory jako je Microsoft Word nebo OpenOffice.org Writer. Díky tomu můžeme vhodně naformátovanou stránku použít namísto formátu doc nebo pdf.

V rozdělení do modulů můžeme najít výhody i nevýhody. Výhodou je tematické rozdělení vlastností a jejich schvalování nezávisle na ostatních modulech. Nevýhodou je sporné zařazení některých vlastností, a z toho plynoucí duplicita kódu nebo jejich složitější vyhledávání.

Cílem této práce bylo vytvořit v České Republice unikátní příručku pro tvorbu webových aplikací pomocí CSS level 3. Tento cíl jsem splnil. V době dokončení neexistuje žádná publikace, která by se CSS level 3 v takovém rozsahu zabývala. V některých příručkách jsou sice zmíněny některé možnosti, nebo rozšíření stávajících vlastností o nové hodnoty, ale žádná se nevěnuje pouze tomuto tématu.

Každému modulu je věnována jedna kapitola. Je v ní stručný popis, čím se modul zabývá a všechny dostupné vlastnosti. Každá vlastnost obsahuje svůj popis, použití, zda je dědičná či nikoliv, hodnoty a implicitní nastavení. Ke většině vlastností jsou přidány i jednoduché příklady, které ukazují konkrétní využití. Pokud je vlastnost podporována, pak jsou vypsány prohlížeče, ve kterých funguje.

Právě funkčnost je hlavním problémem CSS level 3. Přestože se na vývoji pracuje téměř deset let, tak žádný modul neprošel procesem schvalování. Dokonce jsou moduly, které se nenacházejí ani ve stavu pracovního návrhu. Jsou to například modul pro pozicování nebo modul pro práci s tabulkami. To zřejmě ovlivnilo i implementaci v dnešních prohlížečích. Z testovaných prohlížečů Firefox 2.0, Microsoft Internet Explorer 7.0 a Opera 9.2 nabízí dle mého názoru nejlepší podporu prohlížeč Opera 9.2, který například jako jediný umí pracovat s modulem pro stránkovaná média.

CSS level 3 jsou opravdu mocným nástrojem a použití v praxi zatím brzdí především jejich tvůrci. Práce je velmi pomalá a některé návrhy tak zůstávají v nezměněné podobě i několik let. V oblasti výpočetní techniky se dá vývoj předpovídat velmi obtížně, přesto se domnívám, že na plnou podporu v prohlížečích si budeme muset ještě pár let počkat.

## Seznam použitých zdrojů

- [1] CSS Advanced Layout  
<http://www.w3.org/TR/2005/WD-css3-layout-20051215/>
- [2] CSS Aural Style Sheets  
<http://www.w3.org/Style/Group/css3-src/css3-acss/>
- [3] CSS Background and Borders  
<http://www.w3.org/TR/2005/WD-css3-background-20050216/>
- [4] CSS Basic User Interface Module  
<http://www.w3.org/TR/2004/CR-css3-ui-20040511/>
- [5] CSS Box Model  
<http://www.w3.org/TR/2002/WD-css3-box-20021024/>
- [6] CSS Cascading and Inheritance  
<http://www.w3.org/TR/2005/WD-css3-cascade-20051215/>
- [7] CSS Color  
<http://www.w3.org/TR/2003/CR-css3-color-20030514/>
- [8] CSS Fonts  
<http://www.w3.org/TR/2002/WD-css3-fonts-20020802/>
- [9] CSS Generated Content for Paged Media  
<http://www.w3.org/TR/2007/WD-css3-gcpm-20070205/>
- [10] CSS Generated and Replaced Content  
<http://www.w3.org/TR/2003/WD-css3-content-20030514/>
- [11] CSS Hyperlink Presentation  
<http://www.w3.org/TR/2004/WD-css3-hyperlinks-20040224/>
- [12] CSS Line Layout  
<http://www.w3.org/TR/2002/WD-css3-linebox-20020515/>
- [13] CSS Lists  
<http://www.w3.org/TR/2002/WD-css3-lists-20021107/>
- [14] CSS Multi-column Layout  
<http://www.w3.org/TR/2005/WD-css3-multicol-20051215/>

- [15] CSS Namespaces  
<http://www.w3.org/TR/2006/WD-css3-namespace-20060828/>
- [16] CSS3 Paged Media Module  
<http://www.w3.org/TR/2006/WD-css3-page-20061010/>
- [17] CSS Presentation Levels  
<http://www.w3.org/TR/2003/WD-css3-preslev-20030813/>
- [18] CSS Reader Media Type  
<http://www.w3.org/TR/2004/WD-css3-reader-20040224/>
- [19] CSS Ruby  
<http://www.w3.org/TR/2003/CR-css3-ruby-20030514/>
- [20] CSS Speech  
<http://www.w3.org/TR/2004/WD-css3-speech-20041216/>
- [21] CSS Syntax  
<http://www.w3.org/TR/2003/WD-css3-syntax-20030813/>
- [22] CSS Text  
<http://www.w3.org/TR/2007/WD-css3-text-20070306/>
- [23] CSS Values and Units  
<http://www.w3.org/TR/2006/WD-css3-values-20060919/>
- [24] CSS Web Fonts  
<http://www.w3.org/TR/2002/WD-css3-webfonts-20020802/>
- [25] CSS3 – základy stylů v novém kabátě  
<http://interval.cz/clanky/css3-zaklady-stylu-v-novem-kabate/>
- [26] CSS3 – současná implementace v prohlížečích  
<http://interval.cz/clanky/css3-soucasna-implementace-v-prohlizecich/>
- [27] CSS3 – formátování dynamických částí dokumentu  
<http://interval.cz/clanky/css3-formatovani-dynamickych-casti-dokumentu/>
- [28] CSS3 – sloupcové rozvržení a prezentační úroveň  
<http://interval.cz/clanky/css3-sloupcove-rozvrzeni-a-prezentacni-uroven/>

- [29] CSS3 – nahrazovaný a generovaný obsah  
<http://interval.cz/clanky/css3-nahrazovany-a-generovany-obsah/>
- [30] CSS3 – formátování textových informací  
<http://interval.cz/clanky/css3-formatovani-textovych-informaci/>
- [31] CSS3 – nastavení barev, okrajů a pozadí  
<http://interval.cz/clanky/css3-nastaveni-barev-okraju-a-pozadi/>
- [32] Pavlyková, Petra: Tvorba internetových aplikací kaskádovými styly CSS verze 3, absolventská práce, Soukromá vyšší odborná škola a Obchodní akademie s.r.o., České Budějovice 2003.
- [33] Pexa, Petr: Jazyky XHTML, DHTML, CSS a WML, Nakladatelství Kopp, České Budějovice, 2006.