

**JIHOČESKÁ UNIVERZITA  
V ČESKÝCH BUDĚJOVICÍCH**

**Bakalářská práce**

Miroslav Klíma

2009

**JIHOČESKÁ UNIVERZITA  
V ČESKÝCH BUDĚJOVICÍCH  
PEDAGOGICKÁ FAKULTA**

**WAP 2.0 v XHTML-MP, WAP CSS  
kvalitní web na display mobilního telefonu**

**Miroslav Klíma**

Vedoucí práce: PaedDr. Petr Pexa

2009

## **Prohlášení**

Prohlašuji, že svoji bakalářskou práci jsem vypracoval/-a 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.

V Českých Budějovicích dne .....

.....

podpis

## **Anotace**

Tato publikace pojednává o jedné z nejkompexnějších a nejpřístupnějších služeb, určené pro mobilní zařízení, s názvem WAP. V teoretické části práce bude zpracována aktuální tematika tvorby webových prezentací pro mobilní zařízení ve formátu WAP 2.0, bude provedeno srovnání s technologiemi WAP 1.0 a podrobně popsány jazyky XHTML MP a WCSS.

Dále bude vytvořena široká sada praktických příkladů formou mobilní webové prezentace.

## **Abstract**

This publication deals with one of the most complex and accessible services suitable for mobile device called WAP. In the theoretical section, we approach the topical theme of building web presentations suitable for mobile devices in format of WAP 2.0. In this section, we are going to compare WAP 2.0 and technologies WAP 1.0, as well as languages WML, XHTML MP and Wap CSS will be further examined.

Furthermore a wide range of practical examples in the form of mobile web presentation is created.

## **Poděkování**

Děkuji panu PeadDr. Petru Pexovi za odborné vedení, ochotu a nemalou podporu při psaní mé závěrečné práce.

# Obsah

Úvod .....	10
<b>I. WAP, jak vznikl a kam se ubírá.....</b>	<b>11</b>
1.1 Historie protokolu WAP .....	11
1.2 Co je to WAP? .....	11
1.3 Architektura WAP 2.0 .....	12
1.3.1 World-Wide Web komunikační model .....	12
1.3.2 WAP komunikační model .....	13
1.3.3 Vrstvy WAP architektury:.....	15
1.3.4 Bezpečnostní model .....	19
1.3.5 Service Discovery .....	20
1.4 Architektura WAP 1.x .....	21
1.5 Zpětná kompatibilita .....	23
1.6 WAP / Proxy server .....	23
1.6.1 Konfigurace proxy serveru.....	24
1.7 WAP / Podpůrné servery .....	27
1.8 Síťové prvky WAP .....	28
1.9 Shrnutí kapitoly.....	29
<b>II. XHTML Mobile Profile.....</b>	<b>30</b>
2.1 Co je XHTML-MP? .....	30
2.1.1 Verze XHTML-MP .....	31
2.2 Vývojové prostředí .....	31
2.2.1 PSPad .....	32
2.3 Zobrazení WAP obsahu .....	33
2.4 Validace dokumentů .....	34
2.5 MIME typ .....	34
2.5.1 Dynamické generování MIME.....	35
2.6 Přípony XHTML-MP souborů.....	36
2.7 Konfigurace WAP serveru.....	36
2.7.1 Konfigurace Apache serveru.....	37
2.7.2 Konfigurace Tomcat serveru.....	37
2.7.3 Konfigurace IIS serveru .....	38
2.7.4 Uživatelská nastavení.....	38
2.8 Pravidla jazyka.....	39
2.9 XHTML Dokument .....	41

2.9.1	Struktura XHTML-MP .....	41
2.9.2	Komentáře XHTML MP .....	45
2.9.3	Metainformace .....	46
2.9.4	Vazba souborů .....	49
2.9.5	Cizí obsah .....	50
2.9.6	Blokové vs. řádkové elementy .....	51
2.9.7	Formátování textu .....	51
2.9.8	Vizuální změna písma .....	59
2.9.9	Seznamy .....	61
2.9.10	Obrázky .....	64
2.9.11	Tabulky .....	65
2.9.12	Odkazy .....	68
2.9.13	Základní adresování odkazu .....	71
2.9.14	Objekty stránky .....	72
2.9.15	Události .....	74
2.9.16	Formuláře .....	75
2.9.17	Společné atributy .....	83
2.9.18	XHTML MP vs WML .....	84
<b>III. Wireless profile Cascading Style Sheet .....</b>		<b>86</b>
3.1	WCSS verze .....	86
3.2	Výhody kaskádových stylů .....	86
3.3	Nevýhody kaskádových stylů .....	87
3.4	Syntaxe jazyka .....	87
3.4.1	MIME typ .....	87
3.4.2	Umístění v dokumentu .....	87
3.4.3	Komentáře .....	90
3.4.4	Selektory .....	90
3.4.5	Operátory .....	91
3.4.6	Kaskáda .....	92
3.4.7	Dědičnost .....	94
3.4.8	Datové typy .....	96
3.4.9	Média .....	98
3.4.10	Média v tabulkách stylů .....	99
3.4.11	Pseudo třídy .....	100
3.4.12	Box model .....	101
3.4.13	Vizuální formátování .....	107
3.4.14	Rozměry elementů .....	109
3.4.15	Rozměry rámců .....	110
3.4.16	Barvy a pozadí dokumentu .....	111

3.4.17	Vlastnosti písma .....	117
3.4.18	Seznamy .....	122
3.4.19	Vlastnosti textu.....	125
3.4.20	Viditelnost.....	130
3.5	WAP CSS rozšíření .....	131
3.5.1	Rolování obsahu.....	131
3.5.2	Přístupové klávesy .....	133
3.5.3	Rozšíření input .....	134
<b>IV.</b>	<b>Závěr .....</b>	<b>136</b>
<b>V.</b>	<b>Literatura .....</b>	<b>137</b>
<b>VI.</b>	<b>Přílohy.....</b>	<b>139</b>



## Úvod

**Wireless Application Protocol (WAP)** je celosvětovou otevřenou specifikací určenou pro vlastníky mobilních zařízení. Slouží pro snadný a okamžitý přístup k požadovaným informacím a je bezesporu nejdostupnější službou celého světa. Na jeho základě byly vyvinuty technologie, rozšiřující možnosti tvorby a kreativity webových návrhářů např. WML, XHTML-MP, WAP CSS.

A byla to právě ona přístupnost a nástup nových technologií, které mě vedli ke zvolení tohoto tématu. Již dlouho se zabývám problematikou Internetu a tvorby WWW stránek, technologie WAP a dokumenty pro mobilní zařízení byly proto velkou výzvou.

Publikace je členěna na tři obsáhlé části. Cílem první části je seznámit čtenáře se samotným protokolem WAP, jeho stavbou a možnostmi, dále provést ohlédnutí za předchozími verzemi WAP 1.x a zhodnotit výhody/ nevýhody nových přístupů.

Druhá část představuje značkovací jazyk XHTML-MP, tedy jazyk pro tvorbu WAP dokumentů. Kapitola si klade za cíl informovat o pravidlech a možnostech daného jazyka s demonstrací na praktických příkladech a závěrečném srovnání s jazykem WML.

Poslední část je věnována bezdrátovému profilu kaskádových stylů (WCSS). Podstata této závěrečné kapitoly tkví v ukázkách oddělení prezentace od struktury stránek, vysvětlení syntaxe a zhodnocení všech kladů/záporů při použití jazyka.

Globálním cílem je vytvoření první ucelené publikace v České Republice, zabývající se touto problematikou, doprovázené již výše zmíněnými praktickými příklady.

Příklady lze nalézt v sekci *Příklady* na přiloženém CD, nebo na internetové adrese <http://www.webmobile.comuv.com/>. Na téže zdrojích jsou umístěny i další soubory, na které je v průběhu práce odkazováno.

Druhá a třetí kapitola je z velké části tvořena způsobem vysvětlení, příklad a ukázky. Příklady zastupuje piktogram s příslušným umístěním v rámci své kategorie. Pro XHTML-MP je vyhrazena sekce „*Příklady/XHTML-MP*“, pro WCSS analogicky „*Příklady/WCSS*“. Ukázky jsou koncipovány tak, aby co nejlépe odrážely probíranou látku. Volil jsem mezi běžnými WAP prohlížeči.

## I. WAP, jak vznikl a kam se ubírá

Předtím než budou probrány samotné technologie pro vytváření a prezentaci WAP stránek, považuji za nezbytné, podívat se hlouběji do samotné technologie a fungování služby WAP jako takové.

### 1.1 Historie protokolu WAP

Za vznikem protokolu WAP stojí organizace **WAP fórum**, založená roku 1997. Ta v dubnu následujícího roku vydává svou první specifikaci WAP 1.0, v níž byly popsány prostředky pro přístup k službám internetu z mobilních telefonních zařízeních. WAP 1.1 byl vydán roku 1999. Následovala verze 1.2, která byla v červnu 2000 aktualizována na WAP 1.2.1, tedy na poslední verzi „jedničkové“ větve.

Aktuálním standardem je dnes WAP 2.0, ten vznikl a dotvářel se v letech 2001 a 2002. Zásadní rozdíly mezi verzemi 1.X a 2.0 budou popsány v samostatné kapitole.

Organizace WAP fórum sestávala ze společenství předních světových firem v oblasti mobilní komunikace. Jednalo se o společnosti Ericsson, Motorola, Nokia a Unwired Planet. Roku 2002 se WAP fórum, společně s dalšími průmyslovými fóry, sjednotilo do tzv. **Open Mobile Alliance (OMA)**.

OMA je společenství, které vyvíjí otevřené standardy pro světový mobilní průmysl. Jejími členy jsou výrobci software (Microsoft, Sun Microsystems, IBM, Oracle Corporation, Symbian,...), výrobci zařízení (Ericsson, Siemens, Nokia, Sony Ericsson, Philips, Motorola, Samsung, Texas Instruments,...), mobilní operátoři (Telefonica, Vodafone, T-Mobile,...) a další.

První zařízení, využívající služby WAP, nastoupily na zahraničním trhu koncem roku 1999. Na trhu tuzemském se pak objevily na přelomu roku 2000.

### 1.2 Co je to WAP?

Wireless Application Protocol je v praxi zavedená skupina dílčích specifikací rozdělených do jednotlivých funkčních oblastí WAP technologie (např. dílčí specifikace pro oblasti Security, User Agent aj.).

Jeho hlavním cílem bylo vytvořit spojení mezi mobilním bezdrátovým zařízením a světem internetu tak, aby byla umožněna nezávislost přístupu přes technologicky odlišné bezdrátové sítě a byla zachována možnost technologického růstu této myšlenky. Výsledkem byl vznik celosvětového bezdrátového protokolu.

### 1.3 Architektura WAP 2.0

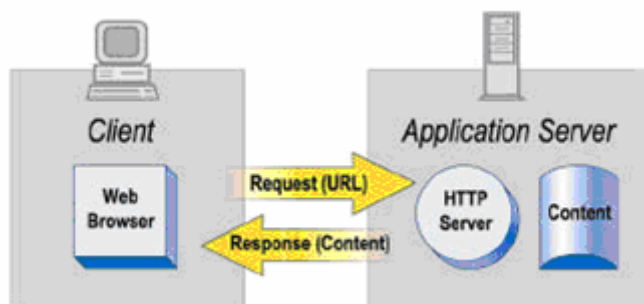
Jak již bylo řečeno, aktuálním standardem je WAP verze 2.0. Ten navazuje na úspěšně implementované předchozí vydání, které rozšiřuje a zdokonaluje.

WAP 2.0 je v mnoha směrech analogií fungování World Wide Web komunikačního modelu, známého ze světa internetu.

WWW model představuje velmi silný a flexibilní programovací model.

#### 1.3.1 World-Wide Web komunikační model

je znázorněn na obrázku č.1



**Obrázek 1 – World Wide Web programovací model**

**Zdroj:** WAP Forum. *WAP Architecture*. [s.l.] : [s.n.], 2001. Dostupný z WWW: <<http://www.openmobilealliance.org/tech/affiliates/wap/wap-210-waparch-20010712-a.pdf>>. The World-Wide Web Model, s. 12. Vlastní úprava.

Komunikace probíhá pomocí komunikačního protokolu (HTTP - Hypertext Transfer Protocol) způsobem **dotaz** (*Request*) / **odpověď** (*Response*). Klient (Web browser) pošle textový URL dotaz na cílový server. Server podá klientovi krátké informace o výsledku svého snažení (typ dokumentu, zde dokument nalezl aj.), za nimi již následuje samotný obsah dokumentu.

Existuje několik norem, nebo spíše doporučení, týkajících se světa internetu. Jednotlivé segmenty by se jimi měly řídit a v drtivé většině i řídí. Tato pravidla reprezentují RFC (Requests for Comments) dokumenty.

Uvedu zde pouze několik vybraných doporučení:

- Všechny serverová zařízení a veškerý internetový obsah musí být pojmenovány Internetovým standardem **Uniform Resources Locator (URL)**. **[RFC2396]**
- U veškerého obsahu je uveden specifický typ, tím je zajištěno korektní zpracování obsahu klienty. **[RFC2045, RFC2048]**.
- Webové prohlížeče musí podporovat všechny standardní formáty obsahu. To zahrnuje především HTML jazyk aj.
- V komunikaci se serverem musí klient podporovat všechny standardní komunikační protokoly. Nejběžnější z nich je HTTP **[RFC2616]**.

Klient a server mohou podporovat řadu technologií, které přispívají k většímu komfortu a k možnostem přenášené informace. Ty jsou lákadlem pro webové tvůrce i uživatele internetu. Pracují buď na straně serveru, nebo na straně klienta. (Server – PHP, ASP / Klient – HTML, XML, JavaScript, CSS).

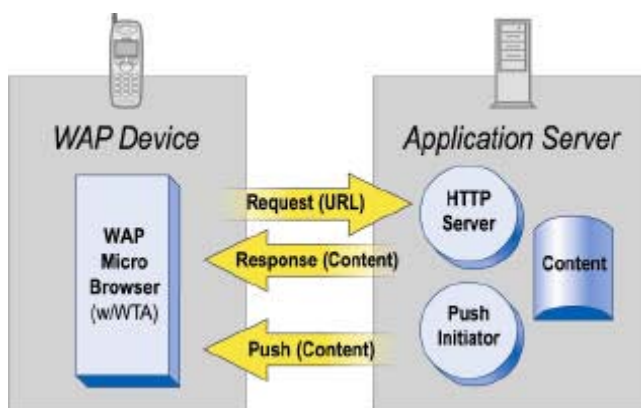
### 1.3.2 WAP komunikační model

WAP komunikační model je založen na WWW programovém modelu a přináší několik vylepšení. Výhody WWW modelu jsou nesporné, jedná se funkční řešení s velkým vývojovým potenciálem. Většina existujících standardů WWW byla převzata, nebo využita k tomu, aby utvářela „startovací čáru“ WAP technologie.

Převážná část rozšíření a modifikací byla vykonána z důvodu charakteristických vlastností bezdrátového prostředí.

#### Nejvýznamnější rozšíření:

- **Push** – mechanismus schopný poslat obsah bez předchozí žádosti.
- **Telephony Support (WTA)** – možnost využívat telefonní služby z aplikačního prostředí.



**Obrázek 2 – WAP programový model**

**Zdroj:** WAP Forum. *WAP Architecture*. [s.l.] : [s.n.], 2001. Dostupný z WWW:

<<http://www.openmobilealliance.org/tech/affiliates/wap/wap-210-waparch-20010712-a.pdf>>. The WAP Model, s. 13. Vlastní úprava.

Mechanismus dotaz-odpověď bývá často označován jako pull mechanismus, na druhé straně stojí mechanismus push. Obsah je přenášen za pomoci standardního komunikačního protokolu HTTP založeného na WWW. Chování a uživatelské rozhraní WAP prohlížeče je analogií klasického webového prohlížeče. Taktéž formáty dokumentu jsou založeny na formátech známých ze světa internetu.

V technologii WAP je definována řada součástí umožňujících komunikaci mezi mobilním zařízením a síťovými servery:

- WAP obsah, umístěný na serverech, jednoznačně identifikuje mechanismus URL. Další převzatý Web standard URI identifikuje prostředky samotného zařízení (př. volání kontrolních funkcí).

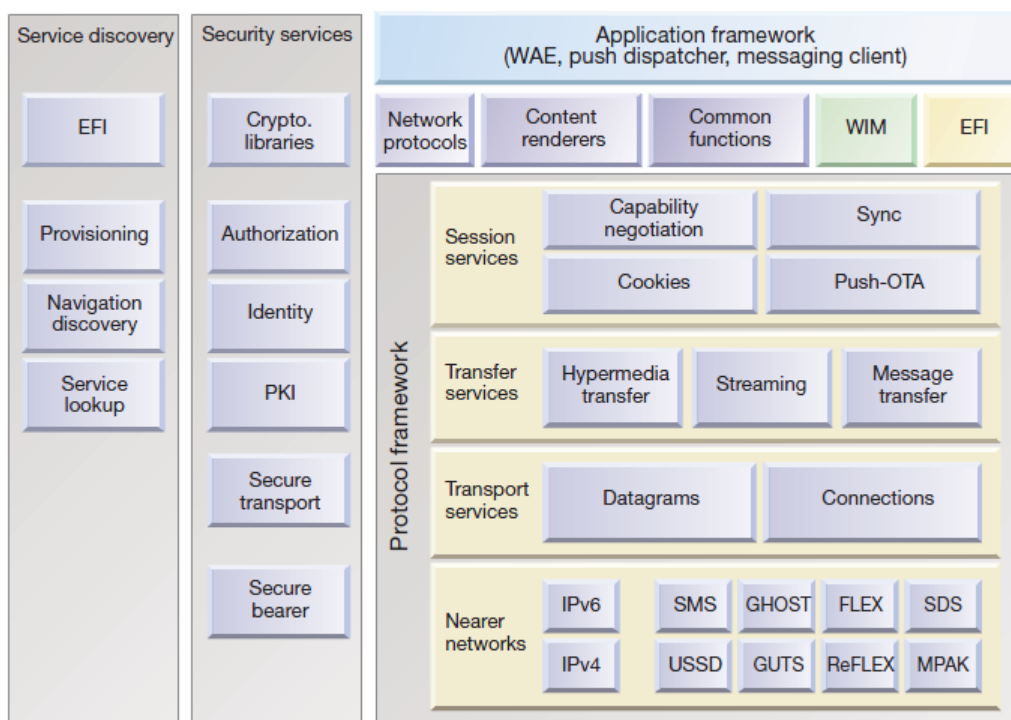
*URI (Uniform Resource Identifier / jednotný identifikátor zdroje) je textový řetězec sloužící k přesnému určení zdroje informací, tím může být dokument, služba atd. URL je podmnožinou URI. [RFC 3986]*

- Typy WAP obsahů odpovídají typům známým z WWW. Tím je umožněno korektní zpracování na straně WAP prohlížeče.
- Formáty obsahu WAP vychází z formátů známých z Web technologie, např. informace kalendáře, skriptovací jazyky, značkovací jazyky aj.
- Za pomoci komunikačních protokolů je umožněno odesílání žádostí na internetové servery z mobilních zařízení.

Protokoly a WAP obsah byly přizpůsobeny pro provoz na ručních bezdrátových zařízeních.

### 1.3.3 Vrstvy WAP architektury:

Architektura protokolu WAP je vrstvená. Jinými slovy, skládá se z jednotlivých funkčních vrstev, které jsou na sebe vzájemně vázané. Každá z vrstev poskytuje řadu funkcí a/nebo služeb, kterých využívají ostatní služby a aplikace. Tato komunikace probíhá přes jednoznačně stanovená rozhraní. Každá z vrstev je přístupná vrstvám nad ní, stejně jako ostatním službám a aplikacím.



**Obrázek 3 – Vrstvy WAP architektury**

**Zdroj:** ARNBY, Peter, HJELM, Johan, STARK, Peter. WAP 2.x architecture—Features, services and functions. [s.l.] : [s.n.], 2001. Dostupný z WWW:

<[http://www.ericsson.com/ericsson/corpinfo/publications/review/2001\\_04/files/2001043.pdf](http://www.ericsson.com/ericsson/corpinfo/publications/review/2001_04/files/2001043.pdf)>. s. 180.

### Bearer Network

Nejspodnějším segmentem architektury je vrstva nazvaná **Bearer Network** (Nearer network), ta reprezentuje různé komunikační mechanismy pro fyzický přenos dat. Je zde podpora IP verze 4 i 6, přenos lze potom realizovat přes řadu komunikačních služeb různých druhů poskytovatelů.

V síti druhé generace GSM může být využito služeb CSD, USSD, GPRS, EDGE aj. Jedná se v podstatě o vytvoření dílčí sítě, využívající stávající prostředky GSM (BTS stanice a další.). Hlavním parametrem služeb je přenosová rychlost.

Pro představu: GPRS dosahuje maximální rychlosti stahování 80kbit/s, EDGE při stejné konfiguraci 236kbit/s. Rozdíl v rychlostech je dán především v modulaci přenášených informací.

Vysokorychlostním nástupcem GSM má být UMTS (W-CDMA). UMTS je již standard spadající do 3. generace mobilní technologie, ta umožňuje souběžný přenos hlasu i dat (např. videohovory). V souvislosti s UMTS je potřeba zmínit technologii HSDPA, ta je součástí specifikace od 5. vydání UMTS (R5). Tato technologie se již blíží k pomyslné slibované hranici stahování 2Mbps.

Mimo bezdrátové sítě lze využít jako prostředníka pro připojení do sítě internetu přímo některého z ISP (Internet service provider). Toho lze dosáhnout pomocí dalších služeb podporovaných mobilním zařízením, např. WLAN. Zde je dnes přenosová rychlost omezena především rychlostí poskytovaného internetového připojení než technologie samotné.

### **Transportní vrstva (Transport Services layer)**

Transportní vrstva nabízí řadu služeb přístupných pro vyšší vrstvy a tyto služby mapuje dostupným službám Bearer Network. Vytváří tak hranici mezi různými druhy přenosů (nejnižší vrstva) a vyššími úrovněmi protokolu. Přenáší nestrukturovaná data přes nejspodnější vrstvu architektury.

Transportní služby:

- **Datagram služba** poskytuje přenos datagramů, tj. nezávislých samostatných celků dat, které mohou putovat přes různé cesty od zdroje k cíli.

Využívá se zde protokolů:

**UDP (User Datagram Protocol)** je transportní protokol, který nezaručuje doručení datagramů do cíle. Výhodou je jednoduchost.

a díky zpětné kompatibilitě WAP i

#### **WDP (Wireless Datagram Protocol)**

- **Spojovací služba** zajišťuje přenos dat v přesně stanovených krocích: navázání spojení, obousměrný spolehlivý přenos dat, uvolnění spojení. Je zde využito komunikačního přenosového protokolu **TCP** resp. **WP-TCP (Wireless Profiled TCP)**, optimalizovaného pro bezdrátový přenos. Ten slouží pro spojení přes **IP Bearer Network**.

*TCP je transportní protokol, který zaručuje spolehlivé doručení paketu do cíle. Pro tento účel je zde využito mnoho kontrolních mechanismů jako kontrolní součet, sekvenční číslo aj. V případě ztráty paketu jsou data opakovaně odeslána, tedy předána IP protokolu k samotnému přenosu.*

Pod názvem paket si lze představit jakoukoliv zprávu (blog informací) přenášenou sítí, zatímco datagramy představují pakety nespolehlivého přenosu dat (UDP).

### **Přenosová vrstva (Transfer Services layer)**

Přenosové služby umožňují strukturovaný přenos mezi dvěma body v síti.

#### **Přenosové služby:**

- **Hypermedia** přenosová služba slouží pro přenos samo-popisných hypermedia prostředků.

**WP-HTTP** (Wireless profiled Hypertext Transfer Protocol) zajišťuje hypermedia přenosovou službu skrze protokoly TCP i UDP. Prohlížeč, nebo server může navíc zvolit (vyžádat) šifrovaný přenos.

#### **Wireless HTTP je kompatibilní s HTTP 1.1.**

*HTTP 1.1 je internetový protokol pracující na principu dotaz- odpověď. Verze 1.1 přináší aktualizaci hlavičky dotazů. Komunikace probíhá formou běžného textu, který je na žádost nahrazen TLS (Transport Layer Security, viz kapitola Bezpečnostní model WAP).*

- **Streaming** služba představuje prostředek pro přenos stejnosměrných dat jako audio a video.
- **Message transfer** služba představuje prostředek pro přenos asynchronních multimediálních zpráv jako email nebo okamžité/rychlé zprávy. **MMS Encapsulation protokol** slouží pro přenos zpráv mezi WAP zařízením a MMS serverem.

### **Relační vrstva (Session Services layer)**

Relační služby zajišťují vytváření sdílených stavů mezi prvky sítě, které vznikají při komunikaci (přenos dat, dotaz aj.). Např. Push session sděluje, zda je zařízení schopno a připraveno přijmout data z Push Proxy.



### Relační služby:

- **Capability Negotiation** (schopnost vyjednávat) – WAP architektura obsahuje specifikace pro popisování a výběr informací a možností klienta, uživatele a prvků sítě (viz UAProf). To poskytuje možnost upravení obsahu serverem, který jej zasílá na klientské zařízení.
- **Synchronizační služba** poskytuje synchronizaci potřebných dat klienta s dalším zařízením (synchronizace kalendáře, kontaktů, email aj.).
- **The Push-OTA**(Over The Air) relační služba zajišťuje doručování obsahu mezi WAP serverem a mobilním zařízením.
- **Cookies** (sušenky) služba zajišťuje uchovávání malého bloku dat poslaného serverem. *Tato data se posílají v hlavičce HTTP dotazu při každém dalším zadání příslušné adresy serveru. Obsahují informace, např. obsah nákupního košíku, přihlašovací údaje aj.*

### Aplikační systémová vrstva (Application Framework)

Aplikační vrstva poskytuje univerzální aplikační prostředí, založené na WWW a Mobilní telefonní technologii. Jejím cílem je vytvořit prostředí umožňující vývoj aplikací a služeb.

#### Application Framework obsahuje:

- **WAE/WTA User-Agent** (tj. prostředí micro-prohlížeče) obsahuje podporu:
  - **Značkovací jazyky WML, XHTML-MP**
  - **Podpora skriptování (WMLScript)**
  - **Kaskádové styly (WCSS)**
  - **Telefonní služby a programové rozhraní**

Vše je přizpůsobeno ručním mobilním zařízením.

- **Push** služba umožňuje přenos dat ze serveru ke klientovi bez předchozího vyžádání klientem. Iniciátor přenosu je vždy některé externí zařízení. Komunikovat lze přes Proxy (Gateway) nebo přímo.
- **Multimediální služba** (Multimedia Message Service) zajišťuje přenos a zpracování multimediálních zpráv jako email a okamžité/rychlé zprávy.

- **Formáty obsahu** jsou různé, zařízením podporované, formáty dat jako obrázky, zvuk, video, animace, položky telefonního seznamu, informace kalendáře aj.

### 1.3.4 Bezpečnostní model

Důležitým aspektem WAP architektury je otázka bezpečnosti. Bezpečnostní služby lze nalézt v mnoha vrstvách WAP protokolu.

**Hlavními bezpečnostními mechanismy jsou:**

- **Soukromí (Privacy)** – Zařízení zajišťují soukromou komunikaci, která by měla zamezovat pochopení prostředními stranami.
- **Ověření (Authentication)** – Zařízení zavádějí ověření jednotlivých stran při komunikaci.
- **Integrita (Integrity)** – Zařízení zajišťují stálou a nerušenou komunikaci.
- **Bez-popření (Non-Repudiation)** – Zařízení při uskutečnění komunikace (vytvoření transakce, poslání zprávy aj.) nemohou tuto komunikaci (transakci, zprávu) později popřít.

**Příklady bezpečnostních služeb:**

- **Šifrovací knihovny (Cryptographic Libraries)** - Application framework knihovna poskytuje služby pro digitální podepsání dat k zajištění integrity a Non-Repudiation.
- **Ověření (Authentication)** – Bezpečnostní služby zajišťují různorodé mechanismy pro ověřování totožnosti serveru a klienta. V řídicí vrstvě může být použit HTTP Client Authentication. V transportní vrstvě mohou být použity protokoly WTLS a TLS.
- **Identita (Identity)** - Wireless Identity Module poskytuje funkce, které uchovávají a zpracovávají informace potřebné pro identifikaci a ověření uživatele. *WIM implementace může být např. Subscriber Identity Module (SIM) nebo různé smart karty a externí karty.*
- **PKI** – je prostředí (sada služeb) umožňující šifrování dat s veřejným klíčem

- **Bezpečnostní přenos** – Transportní vrstva obsahuje protokoly pro bezpečný přenos paketů (TCP) a datagramů (UDP).
  - **Wireless Transport Layer Security** je volitelná vrstva nad transportní vrstvou. Slouží pro bezpečný přenos datagramů.
  - **WAP Transport Layer Security (TLS 1.0)** slouží pro bezpečný přenos paketů.

Jedná se o mechanismy zajišťující šifrování dat na požádání. Vytváří bezpečný „tunel“ mezi serverem a klientským zařízením.

- **Secure Bearer** – Některé prvky nejnižší vrstvy umožňují poskytnutí jistých bezpečnostních mechanismů. Např. v IP protokolu (především IPv6) je integrováno bezpečnostní rozšíření IPSec umožňující šifrování a ověření dat. *V IPv4 je integrován volitelně.*

### 1.3.5 Service Discovery

Service Discovery tvoří součást WAP architektury, jeho služby lze nalézt v mnoha vrstvách.

#### Příklady služeb Service Discovery:

- **External Functionality Interface (EFI)** – Externí funkční rozhraní umožňuje aplikacím odhalit, které externí funkce a služby jsou na zařízení dostupné.
- **Provisioning** – Provisioning služba umožňuje distribuci zařízení s parametry důležitými pro přístup k síťovým službám. Je to proces, kterým je WAP klient nakonfigurován s minimálním přičiněním uživatele.
- **Service Lookup** – Service Lookup služba umožňuje vyhledávání parametrů skrze přeformulování jejich názvu, např. služba DNS.

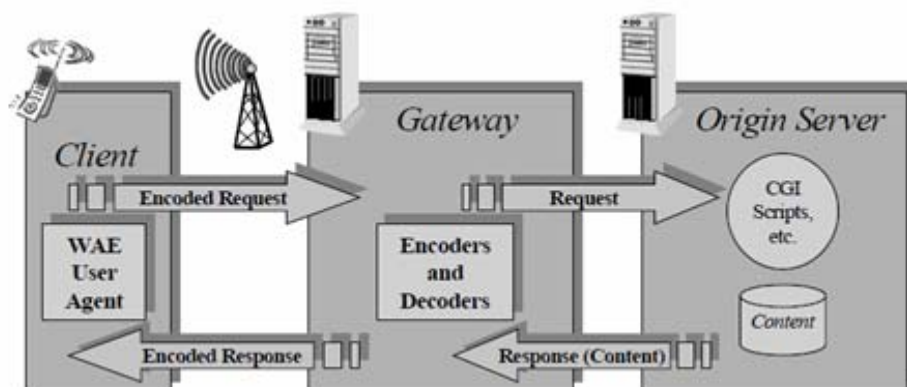
*DNS slouží pro překlad doménových jmen.*

*Např. z tvaru [www.openmobilealliance.org](http://www.openmobilealliance.org) na 195.238.226.77.*

- **Navigation Discovery** – Za pomoci služby Navigation Discovery může zařízení během komunikace nalézat nové síťové služby (např. secure pull server). Navigation Discovery protokol definuje **WAP Transport-Level End-to-End Security** specifikace.

## 1.4 Architektura WAP 1.x

Na Obrázku.4 je znázorněna komunikace klienta s aplikačním serverem, která v modelu 1.x musí probíhat přes WAP bránu (WAP Gateway).



**Obrázek 4 – WAP programový model**

**Zdroje:** WAP Forum. *WAP Architecture*. [s.l.] : [s.n.], 1998. Dostupný z WWW:

<[http://www.openmobilealliance.org/tech/affiliates/wap/Technical\\_June2000-20021106%5B1%5D.zip](http://www.openmobilealliance.org/tech/affiliates/wap/Technical_June2000-20021106%5B1%5D.zip)>. The WAP Model, s. 12.

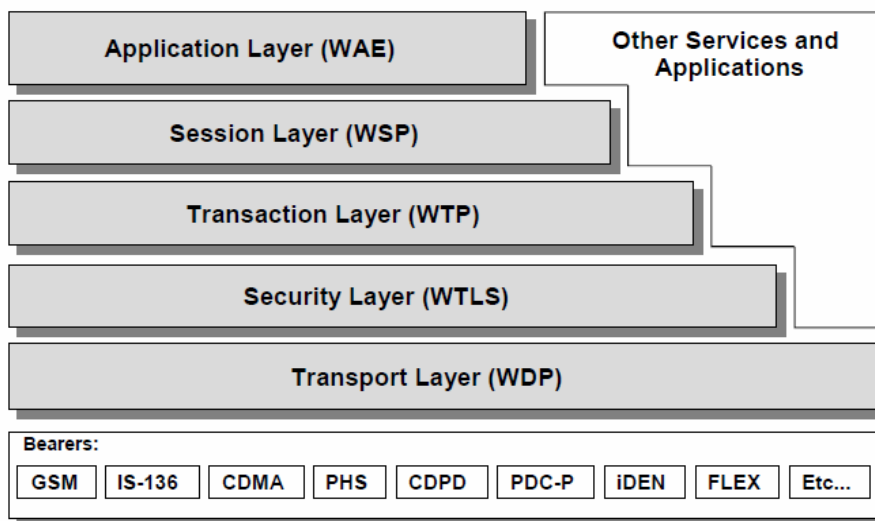
### WAP Gateway má následující funkce:

- Slouží jako brána protokolů, převádí příkazy protokolové sady WAP do protokolové sady WWW (obr.7)
- Kódování-dekódování WAP obsahu - do úspornější formy

WAP Gateway je obdobou nynějším WAP Proxy serverů. Ty díky zpětné kompatibilitě protokolu WAP 2.0 obsahují všechny funkce původní WAP brány a rozšiřují její možnosti.

WAP protokoly jsou vrstveny stejným způsobem, jak jej známe z WWW.

### Sada protokolů WAP:



**Obrázek 5 – Vrstvy WAP architektury**

**Zdroj:** WAP Forum. *WAP Architecture*. [s.l.] : [s.n.], 1998. Dostupný z WWW:

<[http://www.openmobilealliance.org/tech/affiliates/wap/Technical\\_June2000-20021106%5B1%5D.zip](http://www.openmobilealliance.org/tech/affiliates/wap/Technical_June2000-20021106%5B1%5D.zip)>.

Components of the WAP Architecture, s. 15.

- Datová/transportní vrstva – zde se uskutečňuje fyzická komunikace mezi mobilním zařízením a sítí poskytovatele.

#### **WDP (Wireless Datagram Protocol)**

- Bezpečnostní vrstva – tato vrstva zajišťuje bezpečný přenos dat

#### **WTLS (Wireless Transport Layer Security) protokol**

- Transakční vrstva - zajišťuje transakční služby pro uživatelské relace

#### **WTP (Wireless Transaction Layer)**

- Relační vrstva - zajišťuje výměnu dat mezi aplikací a WAP Gateway

#### **WSP (Wireless Session Protocol)**

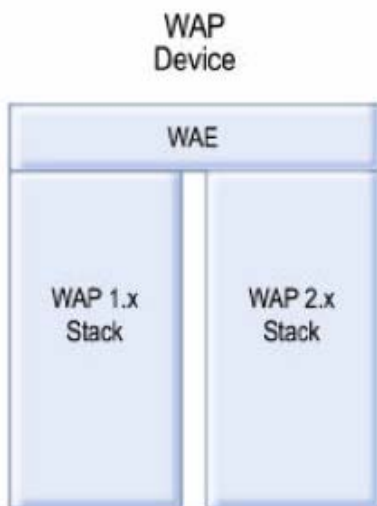
- Aplikační vrstva - aplikační prostředí, umožňující vytváření aplikací a služeb. Obsahuje podporu značkovacího jazyka **WML**. Implementuje skriptovací jazyk **WMLScript**. Podporuje rozhraní **Wireless Telephony Application (WTA, WTAI)**.

#### **WAE (Wireless Application Environment) protokol**

Wireless Telephony Application (WTA, WTAI) je programové rozhraní, které umožňuje aplikacím využívat služby mobilního zařízení a sítě operátora.

## 1.5 Zpětná kompatibilita

WAP 2.0 je zpětně kompatibilní s architekturou WAP 1.x. Aplikační vrstva (WAE) pracuje nad starší i novou sadou protokolů. Tyto sady jsou navzájem nezávislé, mobilní zařízení je dle sítě automaticky přepíná.

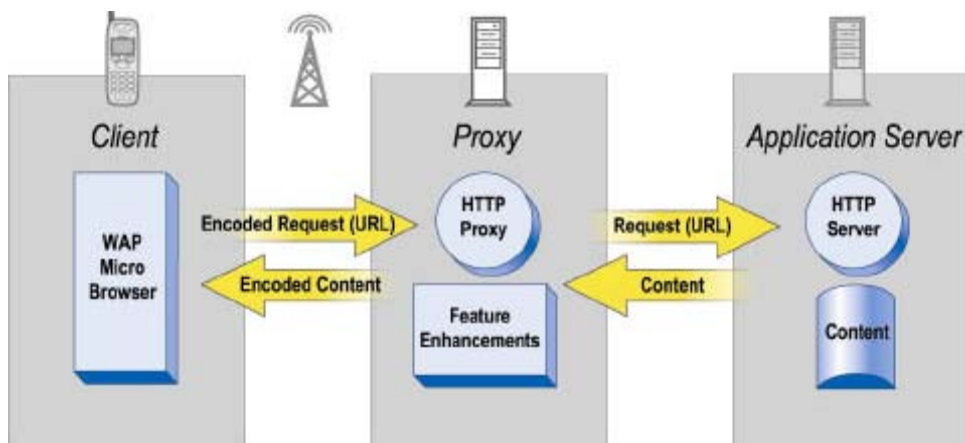


**Obrázek 6 – Zpětná kompatibilita WAP**

**Zdroj:** WAP Forum. *WAP Architecture*. [s.l.] : [s.n.], 2001. Dostupný z WWW:

<<http://www.openmobilealliance.org/tech/affiliates/wap/wap-210-waparch-20010712-a.pdf>>. Sample Configurations of WAP Technology, s. 23. Vlastní úprava.

## 1.6 WAP / Proxy server



**Obrázek 7 – Komunikace s Proxy serverem**

**Zdroj:** WAP Forum. *WAP Architecture*. [s.l.] : [s.n.], 2001. Dostupný z WWW:

<<http://www.openmobilealliance.org/tech/affiliates/wap/wap-210-waparch-20010712-a.pdf>>. Feature/Performance-Enhancing Proxies, s. 14.

Proxy server je síťové zařízení, které slouží jako prostředník v komunikaci mezi aplikačním serverem a koncovým uživatelským zařízením (např. PC, mobilní telefon). Proxy server vystupuje vůči aplikačnímu serveru v roli klienta. Jakmile

obdrží požadované informace, okamžitě je odesílá zpět na koncové zařízení. Tento mechanismus je dobře znám z WWW.

**Použití tohoto řešení v technologii WAP poskytuje různé funkce:**

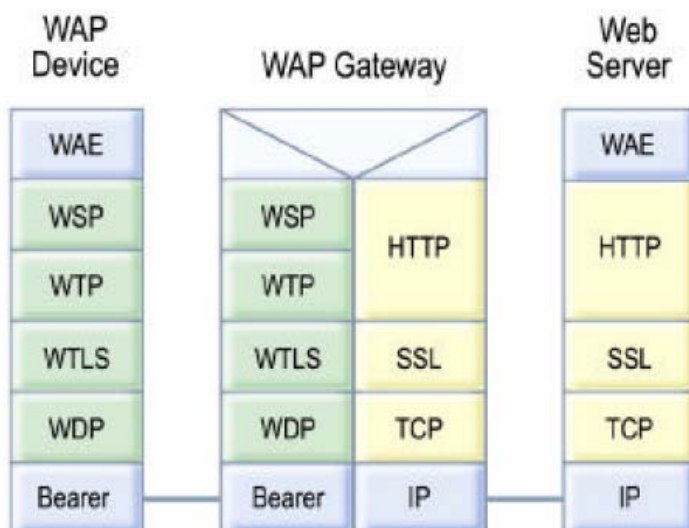
- Protocol Gateway (viz kapitola WAP model 1.x)
- Kódování-dekódování WAP obsahu, zejména kvůli lepší datové propustnosti. Kodér přeloží WAP obsah do kompaktního formátu, který zlepší datovou propustnost tím, že sníží jeho velikost.
- User Agent Profile Management popisuje schopnosti a osobní předvolby aplikací klienta
- Kešování proxy zlepšuje přístup k často navštěvovanému obsahu. Tento obsah je uložen v dočasné paměti proxy serveru, nemusí se tedy znovu získávat z původního zdroje.

### **1.6.1 Konfigurace proxy serveru**

S ohledem na potřeby a možnosti koncových WAP/aplikačních serverů existuje mnoho nastavení prostředníků, kteří jsou mnohdy nezbytní pro úspěšnou komunikaci mezi zařízeními. Uvedu zde pouze výčet několika možností konfigurace.

#### **WAP Gateway**

Typickým příkladem konfigurace je WAP Gateway. Tento způsob komunikace byl zaveden v protokolu WAP verze 1.x (viz kapitola WAP model 1.x). WAP brána převádí hypermedia komunikaci mezi datagram protokoly (WSP, WTP, WTLS, WDP) a spojovalými protokoly internetu (HTTP, SSL, TCP).



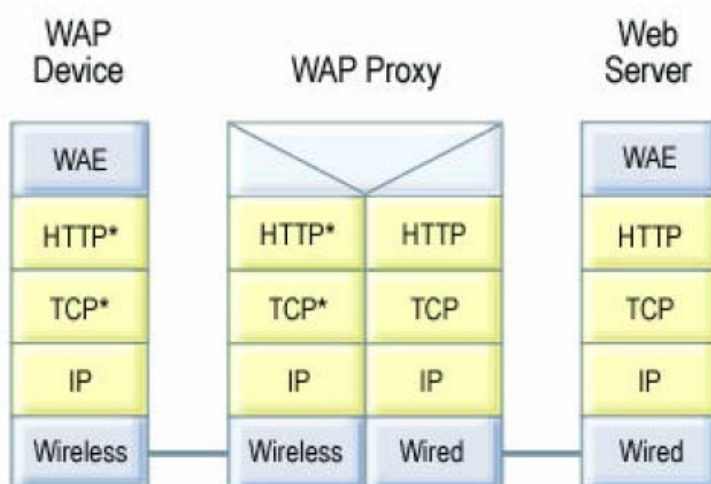
**Obrázek 8 – Konfigurace WAP Gateway**

**Zdroj:** WAP Forum. *WAP Architecture*. [s.l.] : [s.n.], 2001. Dostupný z WWW:

<<http://www.openmobilealliance.org/tech/affiliates/wap/wap-210-waparch-20010712-a.pdf>>. Sample Configurations of WAP Technology, s. 21. Vlastní úprava.

### WAP HTTP Proxy (TCP,HTTP)

Obr.8 znázorňuje nastavení Proxy pro HTTP komunikaci s použitím HTTP a TCP. Tato konfigurace nalézá široké využití u webového přístupu a multimediálních dat. Spojuje klasické drátové a bezdrátové sítě i protokoly a optimalizuje TCP i HTTP komunikaci.



**Obrázek 9 - Konfigurace WAP HTTP Proxy (HTTP,TCP)**

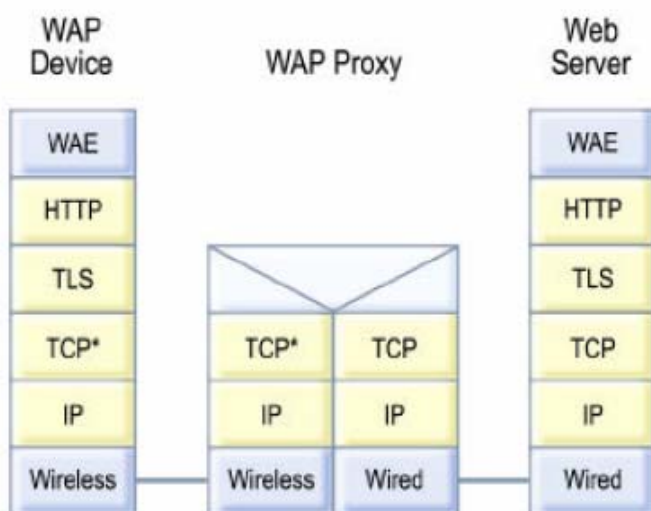
**Zdroj:** WAP Forum. *WAP Architecture*. [s.l.] : [s.n.], 2001. Dostupný z WWW:

<<http://www.openmobilealliance.org/tech/affiliates/wap/wap-210-waparch-20010712-a.pdf>>. Sample Configurations of WAP Technology, s. 22. Vlastní úprava.



### WAP Proxy TLS tunel (TCP)

Obr.9 představuje příklad komunikace nazvaný TLS Tunneling. Konfigurace vytváří bezpečnostní tunel (end-to-end security) mezi serverem a klientským zařízením. Použití TLS vyžadují například elektronické obchody.



**Obrázek 10 - Konfigurace WAP TLS tunelu (TCP)**

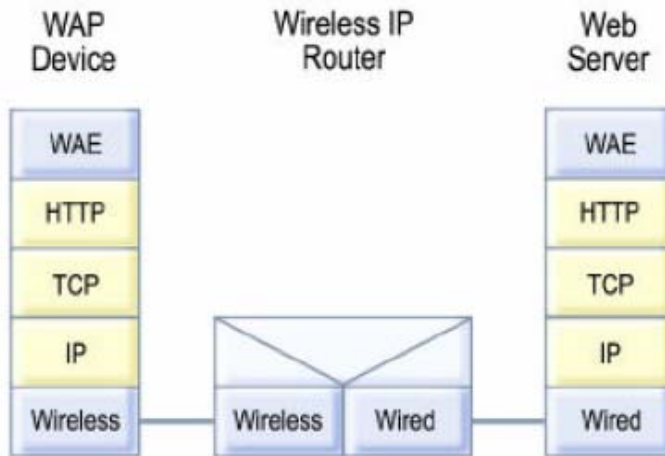
**Zdroj:** WAP Forum. *WAP Architecture*. [s.l.] : [s.n.], 2001. Dostupný z WWW:

<<http://www.openmobilealliance.org/tech/affiliates/wap/wap-210-waparch-20010712-a.pdf>>. Sample Configurations of WAP Technology, s. 22. Vlastní úprava.

### Přímý přístup (Direct Access)

Poslední ukázkou je Direct Access neboli přímý přístup. Zde se jako prostředek využívá bezdrátový IP Router, který je standardní součástí IP sítě. Router převádí IP pakety z jedné sítě (Wireless) do druhé (Wired) a naopak. Používá se na propojení dvou různých typů sítí např. LAN a WAN.

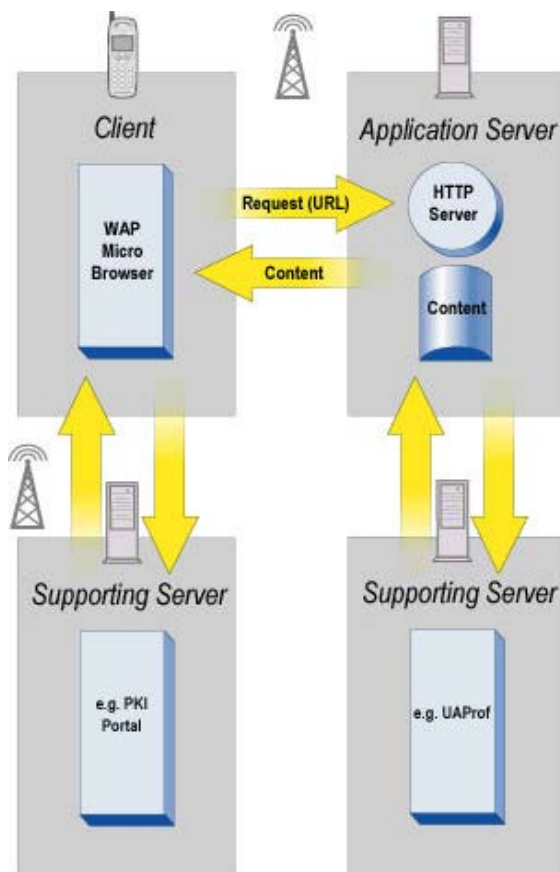
Tato konfigurace může být aplikována na zavedené bearer-level security (IPSec).



**Obrázek 11 - Wireless IP Router**

**Zdroj:** WAP Forum. *WAP Architecture*. [s.l.] : [s.n.], 2001. Dostupný z WWW: <<http://www.openmobilealliance.org/tech/affiliates/wap/wap-210-waparch-20010712-a.pdf>>. Sample Configurations of WAP Technology, s. 23. Vlastní úprava.

## 1.7 WAP / Podpůrné servery



**Obrázek 12 - Podpůrné servery**

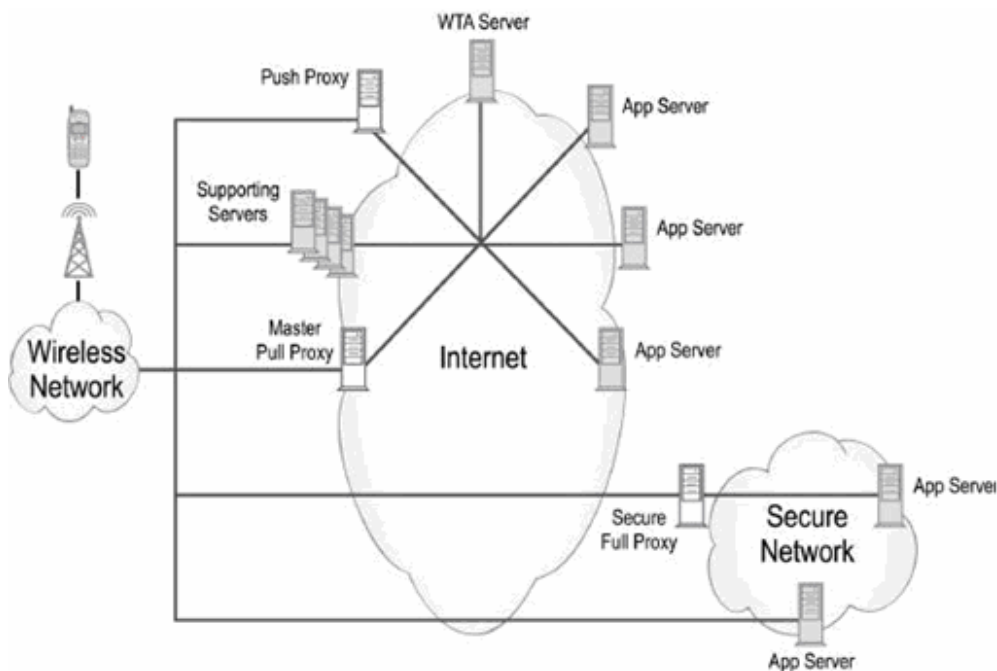
**Zdroj:** WAP Forum. *WAP Architecture*. [s.l.] : [s.n.], 2001. Dostupný z WWW: <<http://www.openmobilealliance.org/tech/affiliates/wap/wap-210-waparch-20010712-a.pdf>>. **Supporting Servers**, s. 15.

WAP Architektura zahrnuje podporu serverů, které poskytují své služby zařízením, proxy a aplikacím, v momentě kdy je vyžadují. Tyto servery mívají často specifické funkce a jsou využívány ve velkém množství aplikací; např. PKI portál, UAPROF servery, Informační servery.

**UAPROF servery** poskytují informace o technických a softwarových možnostech mobilních zařízení (např. velikost displeje, jazyk telefonu aj.). Tyto informace jsou uloženy v XML souborech. URL adresu takového souboru generuje a odesílá zařízení v hlavičce dotazu na cílový aplikační server. Ten se může na základě těchto informací příslušně zachovat a odpovědět.

**PKI portál** poskytuje nástroje pro certifikaci pomocí klíče – zabezpečení spojení.

## 1.8 Síťové prvky WAP



**Obrázek 13 - Příklad sítě WAP**

**Zdroj:** WAP Forum. *WAP Architecture*. [s.l.] : [s.n.], 2001. Dostupný z WWW:

<<http://www.openmobilealliance.org/tech/affiliates/wap/wap-210-waparch-20010712-a.pdf>>. WAP Network Elements, s. 16. Vlastní úprava.

Na obrázku je znázorněn příklad komunikace WAP klienta (prohlížeč v mobilním telefonu) s aplikačními servery. Tato komunikace může probíhat přímou cestou, nebo přes řadu různých zástupných serverů (proxy). Wap klient obsahuje mechanismus pro selekci těchto zástupců, díky kterému je zajištěn výběr těch

nejvhodnějších pro danou službu. Klient může komunikovat jak s proxy servery, tak se servery podpůrnými, a využívat všechny výhody, které tyto technologie přináší.

## 1.9 Shrnutí kapitoly

Oproti předcházejícím specifikacím představuje protokol WAP 2.0 významný krok v před, a to především v oblasti komunikace, prezentace a bezpečnosti dat. Nové možnosti ocení nejen webový designér, ale především koncoví mobilní uživatelé.

### Významná plus:

#### **Prezentace**

V protokolu WAP 2.0 je obsah prezentován formou XHTML-MP dokumentů. S použitím tohoto jazyka lze využít pro vizuální zobrazení na displej kaskádových stylů (WCSS). Tyto nové možnosti s sebou přinášejí lepší formátování, prezentaci a úpravu dokumentů téměř srovnatelnou s vývojem webu na Internetu.

#### **Komunikace**

Komunikace mezi aplikací a serverem probíhá nad standardní sadou protokolů, založenou na modelu WWW (HTTP, TLS, TCP/UDP, IP). To má za následek zrychlení, flexibilitu a celkovou optimalizaci komunikace. Není zde již nutnost využívat prostředníka, jak tomu bylo u starších verzí protokolu. Tato možnost je však zachována a její myšlenka rozvedena.

Na zlepšení komunikace se podílejí i další služby (např. UAProf), s jejich pomocí lze významně redukovat a přizpůsobovat přenášený obsah konkrétnímu mobilnímu zařízení.

#### **Bezpečnost**

Bezpečnostní služby lze nalézt téměř ve všech vrstvách architektury WAP. Využívá se zde známých a ověřených mechanismů WWW (TLS, IPsec aj.). Díky tomu je komunikace resp. přenos dat bezpečnější než tomu bylo za starých časů protokolů 1.x.

## II. XHTML Mobile Profile

XHTML Mobile Profile (eXtensible HyperText Markup Language Mobile Profile) je značkovací jazyk určený pro strukturování dokumentů na mobilních bezdrátových zařízeních. Je součástí protokolu WAP od verze 2.0.

Byl vyvinut společností WAP Fórum (dnešní Open Mobile Alliance) a představen v říjnu 2001. Do této doby „vládl“ mobilním zařízením jazyk WML (kříženec HTML a XML).

Nová koncepce přinesla mnohá vylepšení. Vytváření stránek pro mobilní zařízení je téměř totožné s tvorbou klasického webu. Došlo k odstranění pomyslné bariéry mezi značkovacími jazyky pro WAP a WWW. Stránky napsané v XHTML-MP interpretují i klasické webové prohlížeče (podporující XHTML). Weboví návrháři tak mohou s výhodou využívat stávajících vědomostí, postupů a programových nástrojů, na které byli zvyklí ze světa Internetu.

Jsou zde zachovány požadavky na čistotu kódu, oddělení struktury od prezentace dokumentu a princip sémantického webu. S těmito požadavky je úzce spjata možnost využití kaskádových stylů (WCSS).

Díky zpětné kompatibilitě s předchozími WAP protokoly lze prohlížet i vytvářet dokumenty založené na starších, resp. nových postupech vývoje, tedy jak WML / WMLSCRIPT, tak novém XHTML MP / WCSS.

Taktéž stávající XHTML stránky lze bez větších obtíží upravit tak, aby je byly schopné interpretovat i mini-prohlížeči WAP dokumentů.

XHTML-MP představuje novou, flexibilnější a kreativnější etapu vývoje stránek určenou na displej mobilních bezdrátových zařízení.

Veškeré příklady z této kapitoly lze nalézt v sekci „*Příklady/XHTML-MP*“ na příloženém CD, nebo na internetové adrese <http://www.webmobile.comuv.com/>.

### 2.1 Co je XHTML-MP?

XHTML-MP je z hlediska syntaxe podmnožinou jazyka XHTML (HTML+XML), neboli striktního HTML vyvinutého konsorciem W3C. Jedná se o

podmnožinu nazvanou XHTML Basic, která je rozšířena o některé dodatečné elementy a atributy z plného XHTML.

Jazyk XHTML-MP je modularizovaný. Jednotlivé skupiny dílčích elementů jsou rozděleny do samostatných logických modulů (např. moduly Text, Struktura, Hypertext aj.).

### 2.1.1 Verze XHTML-MP

XHTML-MP vychází z jazyka XHTML Basic 1.0. V dnešní době existují tři verze jazyka XHTML-MP, z nichž každá přidává nové elementy a jejich atributy. Ty jsou potom v novějších verzích dále zachovány. Toto platí přinejmenším do XHTML MP 1.2.

**Verzemi XHTML-MP jsou:**

- **XHTML Mobile Profile 1.0 (21. října 2001)** – Verze 1.0 rozšiřuje XHTML Basic o elementy formulářů `fieldset` a `optgroup`, dále o atribut `start` v elementu `ol` a atribut `value` v `li`. Do prezentačního modulu byly přidány elementy `b`, `big`, `hr`, `i`, `small`. Pro potřeby stylování byly přidán element a atribut `style`.
- **XHTML Mobile Profile 1.1 (16. srpna 2004)** – Verze 1.1 implementuje navíc elementy a atributy umožňující scriptování. Přidány byly události `onload`, `onunload`, `onclick`, `ondblclick`, aj. Z elementů byly přidány `noscript` a `script`.
- **XHTML Mobile Profile 1.2 (18. ledna 2005)** – Verze 1.2 přináší malé změny v modulu formuláře, přidává elementy `button` a `legend`. Dále přidává atribut `inputmode` do elementu `input` a `textarea`.

Podrobnou tabulku s popisem modulů a jednotlivých verzí jazyka naleznete v **Příloze A** na konci publikace.

## 2.2 Vývojové prostředí

XHTML-MP dokument představuje textový dokument s příslušnou asociační příponou XHTML technologie (`*.xhtml`, `*.html`, `*.htm`). Je tedy možné vytvářet stránky pomocí běžných textových a programových editorů (např. programy Poznámkový blok, PSPad, Vim).

Oproti běžným editorům existují různé specializované programy zabývající se problematikou tvorby WAP stránek (např. Nokia Mobile Internet Toolkit). Součástí těchto nástrojů bývají často mnohé další programové moduly, určené k simulování chování a komunikace WAP.

Třetí možností, která zajistí pohodlné psaní a strukturování WAP stránek, je využit stávajících webových editorů, upravených ke spolupráci s příslušnou syntaxí jazyka XHTML-MP (případně dalších jazyků pro mobilní zařízení). Programátor tak může s výhodou používat nástroje, na které byl doposud zvyklý. K tomuto účelu výborně poslouží český freeware produkt z plzeňské dílny Jana Fialy. Řeč je o programu PSPad.

### 2.2.1 PSPad

je volně šířený univerzální programovací editor, určen pro platformu Microsoft Windows. Byl lokalizován do mnoha jazyků. Úplné funkce a možnosti editoru lze nalézt na adrese <http://www.pspad.com/>.

Na českém freeware trhu vyniká především **vlastnostmi našeptávače, automatického doplňování textu, zvýraznění syntaxe** a v neposlední řadě **vytváření a úpravu syntaktických šablon** programovacích jazyků aj.

**Syntaktická šablona** je šablona vytvořená pro účely našeptávače kódu. Obsahuje veškerou syntaxi daného jazyka. U XHTML to jsou povolené elementy a atributy. Ty lze za pomoci dialogového okna našeptávače (Ctrl+Space) snadno vkládat do stránky.

Pro tento účel jsem vytvořil sadu syntaktických šablon jazyků určených pro zobrazení na mobilních zařízeních. Zahrnují všechny aktuální verze jazyků XHTML-MP a XHTML Basic.

Naleznete je v sekci „*Software/Sablony*“ na příloženém CD , nebo internetové adrese <http://www.webmobile.comuv.com/>.

#### Instalace šablon

Instalace šablon je velmi jednoduchá. Příslušnou šablonu je potřeba nakopírovat do složky *Context*, která se nachází v kořenovém adresáři programu. Tato složka obsahuje veškeré syntaktické šablony PSPad. Nakopírováním se přepíše stávající

XHTML syntaxe na syntax vybraného jazyka. Posléze stačí vytvořit nový XHTML soubor (*Soubor/Nový*).

Pokud požadujete založení souboru s výchozí kostrou vybraného jazyka, lze to provést nakopírováním XHTML koster do složky *Template*.

Naleznete je v sekci „*Software/Kostrý*“ na příloženém CD , nebo internetové adrese <http://www.webmobile.comuv.com/>.

V tomto případě zvolíte v nabídce „*Soubor/Nový*“ záložku *Podle Šablony*. Kostry budou umístěny pod položkou XHTML.

### 2.3 Zobrazení WAP obsahu

Zobrazování stránek na mobilním zařízení zajišťuje WAP prohlížeč (WAP User Agent). Uživatelských agentů existuje velké množství a to na všech mobilních platformách .

Uvedu zde pouze malý výčet např.:

- **Internet Explorer Mobile**, typický pro **Windows Mobile**
- **Nokia Mini Map Browser**, typická pro platformu **Symbian**
- **Mobile Safari**, typické pro **iPhone** a **iPod**
- Různé analogie web prohlížečů:
  - **Opera Mini, Opera Mobile** / Prohlížeč Opera
  - **Fennec** / Prohlížeč Firefox

Jednotlivé prohlížeče se předhánějí především v rychlosti zobrazení požadovaných informací, dále potom v množství podporovaných technologií, funkcí a standardů (např. podpora Adobe Flash, funkce zoom, maximální počet otevřených oken aj.). Mimo obrazovku mobilního telefonu existuje několik možností, jak lze prohlížet WAP obsah bez přičinění WAP prohlížeče.

Použití WAP emulátoru: WAP emulátor je program určený k simulování WAP prohlížeče mobilního telefonu. V dnešní době jsou k dispozici v režimu online i offline.

#### Online WAP emulátory:

- **Opera Mini** / <http://www.opera.com/mini/demo>



- **TagTag** / <http://www.tagtag.com/site/emulator>

Další možností je klasický webový klient, podporující daný MIME typ. V tomto směru nabízí jeden z předních prohlížečů, s názvem Opera, PDA zobrazovací režim („Zobrazit/Malý display(PDA)“).

## 2.4 Validace dokumentů

Stejně jako při tvorbě WWW stránek, tak i ve světě mobilního internetu existují mechanismy, pomocí kterých lze snadno kontrolovat validitu/čistotu napsaného programového kódu. Nejznámějším nástrojem schopným spolupracovat i s mobilních XHTML je **W3C Validátor**. Naleznete ho na adrese <http://validator.w3.org/>.

Validita je důležitá pro účely korektního zobrazení WAP obsahu na různých typech prohlížečů, není však jediná! Ve hře jsou různá kódování, MIME typy aj. Takovéto chyby odhalí často až samotný mobilní prohlížeč (nebo emulátor).

Dalším zajímavým počinem společenství W3C je **W3C mobileOK Checker**. Ten umožňuje kontrolu a následná doporučení, jakým způsobem přizpůsobit stávající XHTML stránky mobilnímu provozu.

Nachází se na adrese <http://validator.w3.org/mobile/>

## 2.5 MIME typ

Několikrát jsem se v textu zmínil o tzv. MIME typu (Multipurpose Internet Mail Extensions). MIME je internetovým standardem určeným pro identifikaci obsahu elektronické pošty. Společně s daty je odesílána informace o tom, co tyto data vlastně znamenají (zda se jedná o obrázek, HTML, zvuk aj.). Standard se nepoužívá pouze pro účely e-mailu, tvoří v neposlední řadě rozšíření protokolu HTTP.

MIME informace jsou umístěny v HTTP hlavičce (tak i hlavičce elektronické pošty), předcházejí samotným datům a informují příjemce (uživatelský prohlížeč), s čím má to dočinění. Příjemce se posléze rozhodne a patřičně zachová.

MIME využívá dvouúrovňové klasifikace identifikace obsahu ve tvaru "**typ/podtyp**" (type/subtype). Rozeznáváme sedm hlavních datových typů (typ), zde se budeme zabývat pouze dvěma z nich.

### MIME typy:

- Typ „**text**“ – jakýkoliv srozumitelný text např.:
  - ASCII text ("text/**plain**")
  - HTML („text/**html**“)
  - CSS („text/**css**“)
  - Formát "richtext" nebo "enriched"
- typ "**application**" – zastupuje druh dat, které musejí být zpracovány nějakým konkrétním programem(např. skripty aj.)

Další - **image**, **audio**, **video**, **multipart**, **message** v kombinaci s příslušnými podtypy.

### Jazyk XHTML MP podporuje MIME typy:

1. **application/vnd.wap.xhtml+xml**
2. application/xhtml+xml
3. text/html

Pro mobilní prohlížeče je povinný typ application/vnd.wap.xhtml+xml specifikovaný Open Mobile Alliance, ostatní dva jsou pouze povolené. Typ application/xhtml+xml je běžně používaný pro XHTML dokumenty, stejně jako text/html pro dokumenty HTML. V praxi to znamená, že 2. a 3. typ nemusí mobilní prohlížeč vůbec podporovat, na druhou stranu je však podporují klasické webové prohlížeče.

Použitý typ text/html navíc nemusejí webové prohlížeče interpretovat jako striktní HTML!

Jeden z předních webových prohlížečů FireFox nabízí možnost doinstalovat plugin jménem XHTML Mobile Profile, s jehož pomocí je schopen zpracovat i obsah typu application/vnd.wap.xhtml+xml.

### 2.5.1 Dynamické generování MIME

Jak již název napovídá, MIME typ lze generovat automaticky v závislosti na možnostech cílového prohlížeče. Ten posílá v HTTP hlavičce Accept informace o podporovaných MIME typech. Na tomto základě probíhá výběr a generování výsledného MIME, které je odeslané společně s daty prohlížeči..

Na straně serveru je potřeba využít některého ze skriptovacích jazyků např. ASP, JSP, PHP, Perl aj.

### Příklad

Dynamické generování MIME v ASP:

```
<% String acceptHeader = request.getHeader("accept");
if (acceptHeader.indexOf("application/vnd.wap.xhtml+xml") != -1)
    response.setContentType("application/vnd.wap.xhtml+xml");
else if (acceptHeader.indexOf("application/xhtml+xml") != -1)
    response.setContentType("application/xhtml+xml");
else
    response.setContentType("text/html"); %>
```

**Zdroj:** XHTML MP (XHTML Mobile Profile) . [s.l.] : [s.n.], c2008. Dostupný z WWW:  
<[http://www.developershome.com/wap/xhtmlmp/xhtml\\_mp\\_tutorial.asp?page=mimeTypeFileExtension#6.1.1.Choosing%20MIME%20Types%20Dynamically|outline](http://www.developershome.com/wap/xhtmlmp/xhtml_mp_tutorial.asp?page=mimeTypeFileExtension#6.1.1.Choosing%20MIME%20Types%20Dynamically|outline)>. Choosing MIME Types Dynamically, s. 7.

## 2.6 Přípony XHTML-MP souborů

Klasické přípony XHTML.MP jazyka jsou „\*.xhtml“, „\*.html“ i „\*.htm“. Je možné použít jakoukoliv příponu, jen je potřeba přiřadit v konfiguraci serveru správný MIME typ.

## 2.7 Konfigurace WAP serveru

WAP server je klasický webový server hostující dokumenty určené na obrazovku mobilního telefonu. Často je však potřeba provést jistá nastavení, která zajistí funkčnost a zobrazení mobilního obsahu v mini prohlížeči.

### Nejpoužívanější WWW/WAP servery:

- **Apache** – multiplatformní open-source server, existuje pro platformy Microsoft Windows a Linux / UNIX a mnohé další. Ke stažení je na adrese <http://httpd.apache.org/>.
- **Microsoft IIS** – Microsoft produkt podporující platformu MS Windows.
- **Apache Tomcat** - multiplatformní open-source server je implementací technologii Java Servlet and JavaServer Pages, existuje na mnoha platformách např. Microsoft Windows a Linux / UNIX. Ke stažení na adrese <http://jakarta.apache.org/tomcat/>.

Hlavní záležitost, kterou je potřeba vyřešit, je konfigurace výše zmíněných MIME typů stránek a dalších souborů mobilní technologie (např. WCSS). Konfigurace spočívá v mapování MIME typů na přípony souborů.

WAP Server bude následně schopen reagovat na dotazy odesláním dat s příslušným MIME v hlavičce odpovědi.

### 2.7.1 Konfigurace Apache serveru

Propojení mezi příponami souborů a MIME typy zajišťuje konfigurační soubor „*domovský\_adresář\_Apache/conf/mime.types*“. Editujte soubor „*mime.types*“.

Mapovací řetězec má tento tvar:

```
mimeTyp přípona1 přípona2 přípona3 ...
```

#### Příklad

Mapování MIME na přípony souborů v Apache serveru:

```
text/html html htm
application/vnd.wap.xhtml+xml xhtml
text/css css
```

### 2.7.2 Konfigurace Tomcat serveru

V verzích 4.x a 5.x je konfigurační soubor umístěn v adresáři „*domovský\_adresář\_tomcat/conf/web.xml*“.

Mapování probíhá pomocí XML elementů `<mime-mapping>`, `<extension>`, `<mime-type>`.

#### Příklad

Mapování MIME na přípony souborů v Tomcat serveru:

```
<web-app>
...
  <mime-mapping>
    <extension>xhtml</extension>
    <mime-type>application/vnd.wap.xhtml+xml</mime-type>
  </mime-mapping>
  <mime-mapping>
    <extension>html</extension>
    <mime-type>text/html</mime-type>
  </mime-mapping>
  <mime-mapping>
    <extension>htm</extension>
    <mime-type>text/html</mime-type>
  </mime-mapping>
...
</web-app>
```

### 2.7.3 Konfigurace IIS serveru

IIS server poskytuje pro mapování grafické uživatelské rozhraní.

Následující kroky znázorňují nastavení IIS 5.0 na Microsoft Windows 2000:

- Otevřete IIS a přejděte na:  
*Control Panel -> Administrative Tools -> Internet Service Manager*
- Po levé straně naleznete stromovou nabídku, ve které kliknete pravým tlačítkem myši na název vašeho počítače a zvolte položku *Properties*
- Otevře se okno *Properties*, kde kliknete na tlačítko *Edit* v nabídce *Computer MIME Map*. Vyvolá se okno *File Types*.
- Zde lze mazat, upravovat a přidávat nové mapovací sekvence.

Pro ostatní IIS (verze 3.0 a výše) by měl být postup podobný.

### 2.7.4 Uživatelská nastavení

Výše uvedená nastavení vyžadují, aby dotyčný měl pro konfiguraci příslušná administrátorská práva. Pokud tomu tak není, např. umístíte li obsah na server, který spravuje hostingová společnost (běžná záležitost), máte přesto možnost provádět jistá uživatelská nastavení a tím ovlivňovat funkci serveru. Pokud to poskytovatel dovolí.

#### Nastavení Apache serveru

Pod serverem Apache je možnost využít konfiguračního souboru *.htaccess*. Jedná se o textový soubor, vytvořený a umístěný do požadovaného adresáře na server. Obsahuje pravidla, která následně platí pro veškerý obsah složky (tedy i na všechny podsložky složky), do které byl soubor umístěn. Souborů *.htaccess* může být na serveru více. Jejich platnost sahá až do dalšího výskytu téhož souboru, kterým tyto vlastnosti můžete změnit.

#### ***.htaccess* nastavuje např.:**

- vlastní chybová hlášení
- použití alternativního výchozího souboru (*index.xhtml/html*)
- zákaz a povolování k vstupu IP adres
- zákaz nebo povolení procházení adresářů, přidání hesla ke složce aj.

Obsahuje vše co lze nastavit v konfiguračním souboru *http.conf*. Používání *.htaccess* je nutné v tomto souboru nastavit (instrukce AllowOverride).

Pokud budete soubor kopírovat na server pomocí FTP klienta, nemusí být po nakopírování viditelný. Tečna na začátku souboru značí skrytý soubor (Linuxu,Unixu). Potom záleží pouze na dovednostech FTP klienta. Např. Total Commander umožňuje zobrazování skrytých souborů pod položkou *Sít' -> FTP - zobrazit skryté soubory*.

Mapování MIME typů pomocí *htaccess* naleznete v souboru *MIME-mapovani.zip* v sekci „Příklady/htaccess“ na přiloženém CD , nebo internetové adrese <http://www.webmobile.comuv.com/>. Na stejném umístění se nacházejí i další užitečné konfigurace, které s výhodou uplatníte na vašem Apache WAP serveru.

### Serverové technologie

Určitá nastavení lze také provádět za použití technologií, pracujících na straně serveru (např. ASP, JSP, Perl, PHP aj.). Kód se vkládá přímo do souboru dokumentu.

Nastavení MIME typů odesílaných XHTML pomocí ASP:

```
<% Response.ContentType = " mimeType " %>
```

Tento zápis zajistí odeslání souboru \*.asp s MIME typem *mimeType* v HTTP hlavičce.

Nastavení MIME typů odesílaných XHTML pomocí PHP:

```
<?php header("ContentType: mimeType"); ?>
```

## 2.8 Pravidla jazyka

XHTML-MP je jazyk vycházející z XHTML; tedy z HTML přeformulovaného pomocí zásad XML dokumentů. Oproti HTML je potřeba striktně dodržovat jistá pravidla, která mají zásadní vliv na správné vykreslení WAP stránek. Mobilní prohlížeče mohou být k některým pravidlům do jisté míry tolerantní. Do budoucna však toto platit nemusí a jistě by nemělo.

### Pravidlo A – Všechny elementy musí být uzavřeny

Všechny párové i nepárové značky musí být ukončeny.



**Příklad**

Chybně	Správně
<pre>&lt;p&gt;Odstavec 1 &lt;p&gt;Odstavec 2 &lt;p&gt;Odstavec 3</pre>	<pre>&lt;p&gt;Odstavec 1&lt;/p&gt; &lt;p&gt;Odstavec 2&lt;/p&gt; &lt;p&gt;Odstavec 3&lt;/p&gt;</pre>
<pre>&lt;hr&gt; &lt;br&gt;</pre>	<pre>&lt;hr /&gt; &lt;br /&gt;</pre>

**Pravidlo B – Všechny elementy a atributy musí být psány malými písmeny**

Jazyk XHTML-MP je citlivý na velikost písmen (case-sensitive). Elementy a atributy se píšou malými písmeny.

**Příklad**

Chybně	Správně
<pre>&lt;P CLASS="Ods1" &gt;P1&lt;/P&gt; &lt;P CLASS="Ods2" &gt;P2&lt;/P&gt;</pre>	<pre>&lt;p class="Ods1" &gt;P1&lt;/p&gt; &lt;p class="Ods2" &gt;P2&lt;/p&gt;</pre>

**Pravidlo C – Všechny hodnoty atributů musí být v uvozovkách**

Toto pravidlo platilo již v HTML, v XHTML MP je striktně vyžadováno.

**Příklad**

Chybně	Správně
<pre>&lt;p class=Ods1 &gt;P1&lt;/p&gt; &lt;p class=Ods2 &gt;P2&lt;/p&gt;  &lt;ul id=Seznam&gt;   &lt;li title=P1&gt;Položka1&lt;/li&gt;   &lt;li title=P2&gt;Položka2&lt;/li&gt; &lt;/ul&gt;</pre>	<pre>&lt;p class="Ods1" &gt;P1&lt;/p&gt; &lt;p class="Ods2" &gt;P2&lt;/p&gt;  &lt;ul id="Seznam"&gt;   &lt;li title="P1"&gt;Položka1&lt;/li&gt;   &lt;li title="P2"&gt;Položka2&lt;/li&gt; &lt;/ul&gt;</pre>

**Pravidlo D – Všechny atributy musí mít hodnotu****Příklad**

Chybně	Správně
<pre>&lt;p class &gt;P1&lt;/p&gt; &lt;p class &gt;P2&lt;/p&gt;  &lt;ul id &gt;   &lt;li title &gt;Položka1&lt;/li&gt;   &lt;li title &gt;Položka2&lt;/li&gt; &lt;/ul&gt;</pre>	<pre>&lt;p class="Ods1" &gt;P1&lt;/p&gt; &lt;p class="Ods2" &gt;P2&lt;/p&gt;  &lt;ul id="Seznam"&gt;   &lt;li title="P1"&gt;Položka1&lt;/li&gt;   &lt;li title="P2"&gt;Položka2&lt;/li&gt; &lt;/ul&gt;</pre>

**Pravidlo E – Elementy se nesmějí křížit**

Platí již od HTML, v XHTML MP (XHTML) je striktně vyžadováno.

**Příklad**

Chybně	Správně
<pre>&lt;em&gt;&lt;p&gt; Další odstavec &lt;/em&gt;&lt;/p&gt;</pre>	<pre>&lt;p&gt;&lt;em&gt; Další odstavec &lt;/em&gt;&lt;/p&gt;</pre>
<pre>&lt;ul&gt;   &lt;b&gt;&lt;li&gt;Polozka1&lt;/b&gt;&lt;/li&gt;   &lt;li&gt;&lt;i&gt;Polozka2&lt;/li&gt;&lt;/i&gt; &lt;/ul&gt;</pre>	<pre>&lt;ul&gt;   &lt;li&gt;&lt;b&gt;Polozka1&lt;/b&gt;&lt;/li&gt;   &lt;li&gt;&lt;i&gt;Polozka2&lt;/i&gt;&lt;/li&gt; &lt;/ul&gt;</pre>

**2.9 XHTML Dokument**

Protože jazyk XHTML-MP existuje v několika verzích, jednotlivé příklady se budou odvíjet vždy od nejnižší z nich. Tedy nejnižší verze, která podporuje probíranou problematiku (elementy, atributy).



kostry\_dok/

**2.9.1 Struktura XHTML-MP**

Minimální kostra WAP stránky:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.0//EN"
"http://www.wapforum.org/DTD/xhtml-mobile10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

  <head>
    <title>XHTML Mobile Profile 1.0</title>
  </head>

  <body>
    <h1>XHTML Mobile Profile 1.0</h1>
    <p>
      Kostra dokumentu jazyka XHTML-MP 1.0. <br />
      Kódování utf-8, MIME typ
      application/vnd.wap.xhtml+xml
    </p>
  </body>
</html>
```





## XML deklarace

Na prvním řádku kostry dokumentu se nachází tzv. XML prolog, ten obsahuje informace o verzi XML dokumentu (jelikož XHTML-MP dokumenty jsou XML dokumenty) a použitém kódování (*příklad 1*).

### Příklad

Deklarace XML prologu.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml version="1.0" encoding="iso-8859-2"?>
```

Pokud je nastaveno kódování UTF-8 nebo UTF-16, nemusí se XML deklarace uvádět (*příklad 2*). Případně lze vynechat pouze atribut `encoding` (*příklad 3*). Pokud je kódování jiné než obě uvedená (např. latinka), je potřeba deklaraci plně zachovat. Prohlížeč by nastavil výchozí sadu UTF-8 a stránka by byla nečitelná (*příklad 4*).

## DOCTYPE deklarace

Definice typu dokumentu (DTD) je pro jazyk XHTML-MP povinná. Říká prohlížeči, kde nalézt soubor s informacemi o použitém druhu programovacího jazyka. Tento soubor obsahuje veškerou syntaxi daného jazyka. Právě těchto informací využívají různé validační nástroje při kontrole dokumentu.

### Příklad

Deklarace DOCTYPE jazyka XHTML-MP 10.

```
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile
1.0//EN""http://www.wapforum.org/DTD/xhtmlmobile10.dtd">
```

## Kořenový element <html>

Kořenovým elementem dokumentu je párová značka `<html></html>`. Ta vyznačuje začátek a konec samotného dokumentu. Všechny další elementy se nacházejí uvnitř této značky. Element `html` musí obsahovat definici jmenného prostoru (*namespace*).

**Jmenný prostor** je oblast názvů, které spolu jistým způsobem souvisí. Určuje její URI identifikátor. Všechny XHTML elementy spadají do prostoru `http://www.w3.org/1999/xhtml`, tento odkaz určuje význam XML elementů

a atributů (toto je atribut, toto je značka). Podle významu by cílová aplikace měla s elementy/atributy naložit. Pokud jmenný prostor nezná, měla by neznámé elementy/atributy ignorovat.

Pro definici jmenných prostor slouží atribut `xmlns`, hodnota tohoto atributu se vztahuje na všechny elementy v něm obsažené (až do další `xmlns` definice). Za atributem `xmlns` následuje vždy tzv. prefix. Prefix je od atributu oddělen dvojtečkou (`xmlns:prefix`), touto hodnotou může být i prázdný řetězec (`xmlns`). Prefix slouží pro identifikaci elementů, které spadají do daného jmenného prostoru. Atributy tento prefix „dědí“ od svých elementů. Lze jej tedy uvést u názvu elementu a případné atributy mohou zůstat bez něj. Nejlépe to vystihují následující ukázky.

### Příklad

Definice jmenného prostoru s prázdným prefixem. Prázdný prefix je definován u elementů `title`, `h1`, `ul`, `li`, ty proto spadají do tohoto jmenného prostoru.



namespace/  
příklad1

```
<html xmlns="http://www.w3.org/1999/xhtml">
...
<title>Titulek dokumentu</title>
...
<h1>Jmenné prostory</h1>
<p>V html definice xmlns.</p>
<ul>
  <li>Polozka1</li>
  <li>Polozka2</li>
  <li>Polozka3</li>
  <li>Polozka4</li>
</ul>
...
</html>
```

### Ukázky

## Jmenné prostory

V html definice xmlns.

- Polozka1
- Polozka2
- Polozka3
- Polozka4

Internet Explorer Mobile  
(Windows Mobile 6)

## Jmenné prostory

V html definice xmlns.

- Polozka1
- Polozka2
- Polozka3
- Polozka4

Opera 9.5 beta  
(Windows Mobile - Touchscreen)



## Příklad

Dokument bez deklarace jmenného prostoru. Budou vypuštěny všechny neznámé elementy (neznámé pro prohlížeč). Zůstane pouze čistý obsah.

```
<html>
...
<title>Titulek dokumentu</title>
...
<h1>Jmenné prostory</h1>
<p>Bez definice jmenného prostoru.</p>
<ul>
  <li>Polozka1</li>
  <li>Polozka2</li>
  <li>Polozka3</li>
  <li>Polozka4</li>
</ul>
...
</html>
```

## Ukázky

Tento příklad zobrazily oba výše zmíněné WAP prohlížeče shodně s předchozími ukázkami. Ačkoli XHTML-MP je XML dokument, validační nástroj W3C neobjevil v dokumentu žádné chyby. Test byl proveden na všech třech aktuálních verzích jazyka (verze 1.0/1.1/1.2). Dle mého názoru se jedná o další ukázkou tolerance a podporu nestandardního chování ze strany WAP prohlížečů. WAP prohlížeče nejsou v této problematice jednotné; viz níže.

Titulek dokumentu Jmenné prostory  
Bez definice jmenného prostoru.  
Polozka1 Polozka2 Polozka3 Polozka4

Skyfire 0.9.333  
(Windows Mobile)

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
- <html>
- <head>
```

Emulátor Fenec/Firefox  
1.0a1

## Příklad

Další dva možné způsoby ukazují použití prefixu (*příklad3*) a dědičnosti `xmlns` (*příklad4*).



## Hlavička dokumentu <head>

Začátek a konec hlavičky dokumentu tvoří párový element `<head></head>`. Obsah elementu se nezobrazuje. Uvádí se zde informace týkající se dokumentu samotného, nebo informace vztahující se na celý dokument. Např. různé

metainformace `<meta />`, vazby souborů `<link />`, titulek stránky `<title>`, styly dokumentu `<style>`, skripty `<script>` aj.

### Titulek stránky `<title>`

Obsah párového elementu `<title></title>` značí titulek dokumentu. Tento titulek má několik funkcí:

- některé mobilní prohlížeče ho zobrazují jako nadpis okna
- tvoří nadpis ve webových vyhledávačích aj.

### Příklad

Deklarace titulního názvu dokumentu.

```
<title>Titulek dokumentu</title>
```



title/  
příklad1

### Ukázka



Opera 9.5 beta  
(Windows Mobile - Touchscreen)



Emulátor Fennec/Firefox1.0a1

### Tělo stránky `<body>`

Veškerý zobrazitelný obsah dokumentu je umístěn v párové značce `<body></body>`. Jelikož XHTML MP patří mezi jazyky značkovací, měla by být veškerému obsahu přiřazena patřičná značka, která co nejpřesněji vystihuje jeho obsah. **Text by tedy neměl být umístěn samovolně.**

### Příklad

Správné umístění	Chybné umístění
<pre>&lt;body&gt;   &lt;h1&gt;Ahoj světe&lt;/h1&gt; &lt;/body&gt;</pre>	<pre>&lt;body&gt;   Ahoj světe &lt;/body&gt;</pre>

## 2.9.2 Komentáře XHTML MP

V XHTML MP se umístěny komentáře mezi sekvenci znaků `<!-- -->`.

```
<!--Toto je jednořádkový i víceřádkový komentář -->
```

### 2.9.3 Metainformace

Meta informace představují informace popisující samotný dokument (stránku), nejsou součástí obsahu. Umístíme se do elementu `<head></head>`.

#### Element `<meta />`

Jedná se o nepárovou značku. Popis se provádí za pomoci atributů. Informace obsažené v meta jsou zdrojem pro servery i klientské aplikace. Ty je buď mohou, nebo nemusí využít. Pokud prohlížeč některým atributům nerozumí, přejde jej.

Možnosti meta elementu je nejlépe předvést na příkladech.

#### Příklad

Nejvyužívanější meta informací je deklarace MIME a kódování dokumentu (*příklad1*). Tato informace radí serveru, s jakým MIME typem má dokument odeslat (pokud tento MIME typ umí) a naopak prohlížeči, v jakém kódování má obsah zobrazit. Některé WAP prohlížeče (Např. Internet Explorer Mobile/WM6) tuto informaci přímo vyžadují (*příklad2*), bez ní riskujete chybné zobrazení znaků v obsahu dokumentu. META element s kódováním by měl být umístěn ještě před značkou `title`. Mohl by vzniknout problém v zobrazení znaků titulku stránky. (*příklad3*).



mime/  
příklad1  
příklad2  
příklad3

```
<head>
...
<meta http-equiv="Content-Type"
content="application/xhtml+xml; charset=utf-8" />
...
</head>
```

#### Ukázky

##### **META informace**

Obsahuje meta s MIME a kódováním.

ÉÍÁÝŽŘČŠĚ

éíáýžřčšě

##### **META informace**

Bez meta elementu, kódování v xml.

Ã%ooÃ □Ã □Ã □Ã½Å~ÃCEÃ ÅŠ

Ã©Ã-ÃiÃ½Å¾4Ã™Ã □ÃiÃ>

Internet Explorer Mobile  
(Windows Mobile 6)

**Příklad**

V meta elementu lze uvádět různorodé informace o popisu obsahu. Lze uvádět např. **autory částí dokumentu**. Jména by měla být zapsána v angličtině, aby nevznikl problém zobrazení znaků např. Ruštiny (v jiném kódování než UTF).

```
<meta name="author" content="XHTML code: Miroslav Klima;
e-mail: miroslavklima@email.cz" />
<meta name="author" content="CSS code: Miroslav Klima;
e-mail: miroslavklima@email.cz" />
```

Dále zde mohou být uvedena různá **klíčová slova** (keywords), **popis** dokumentu (description), **copyright** stránky, **jazyk** stránky aj. Informaci description využívají webové prohlížeče při popisu vyhledané stránky.

```
<meta name="description"
content="Toto je ukázka použití značky META." />
<meta name="keywords" content="slovo1, slovo2, slovo15" />
<meta name="copyright" content="© 2009, MADE in HOME" />
<meta name="content-language" content="cs" />
```

**Příklad**

Užitečným příkladem použití je možnost ovládat uložení načtené stránky do vyrovnávací paměti (cache paměť), ze které se v případě žádosti stránka vyzvedne (pokud jí nevypršela platnost/expirační doba). Meta elementy ve stránce říkáme, můžeš/nemůžeš uložit stránku, případně po jak dlouhou dobu.

Vše se odehrává na straně klienta!

Zákaz umístění stránky do vyrovnávací paměti (**příklad 5**) :

```
<meta http-equiv="cache-control" content="no-cache" />
```

Pokud element uveden není, je cachování povoleno. V tomto případě je vhodné nastavit dobu, po které se má dokument opětovně načíst ze serveru.

Nastavení expirační doby (**příklad 6**):

```
<meta http-equiv="Expires" content="Datum" />
```

**Datum** expirace se uvádí ve tvaru:

Wed, 30 Apr 2009 12:00:00 GMT



mime/  
příklad4



mime/  
příklad5  
příklad6  
příklad7

Jedná se Greenwichský hlavní čas (GMT), nikoli lokální. Je zde také možnost nastavit hodnotu na 0 nebo -1. Ty však nejsou standardní.

Definování expiračního času od přijetí dokumentu – 600 vteřin (*příklad 7*):

```
<meta http-equiv="Cache-Control" content="max-age=600" />
```

S hodnotou `max-age=0` docílíte stejného efektu jako s `no-cache`.

### Příklad

Další příklad představuje použití hodnoty `refresh` atributu `http-equiv`, kterou je možné celkem snadno provádět obnovování nebo samotné přesměrování stránky na stránku jinou.

**Obnovení stránky (*příklad 8*):**

Pro správnou funkci je potřeba zajistit obnovu stránky ze serveru a ne z vyrovnávací paměti cache. Stránka se obnoví každých 5 vteřin.

```
<meta http-equiv="Cache-Control" content="no-cache" />
<meta http-equiv="refresh" content="5" />
```

**Přesměrování (*příklad 9*):**

Jednoduchou úpravou předchozího řešení získáme možnost přesměrování na definovanou internetovou adresu.

```
<meta http-equiv="refresh" content="10;URL=hello.xhtml" />
```

Prohlížeč přejde po 10 vteřinách na požadovanou URL adresu.

### Ukázky

<p><b>META informace</b></p> <p>Přesměrování na URL (10s).</p>
--

<p><b>Hello body</b></p>
--------------------------

Internet Explorer Mobile  
(Windows Mobile 6)



mime/  
příklad8  
příklad9

## 2.9.4 Vazba souborů

### Element `<link />`

Nepárový element `link` slouží pro vytváření neviditelného vztahu mezi dvěma soubory. Lze ho umístit pouze do hlavičky stránky. Poskytnuté informace mohou, ale nemusí WAP prohlížeče využít.

Nejběžněji se udává ve tvaru:

```
<link rel="vztah" type="MIME typ" href="URL cesta">
```

Používá se především pro spojení se souborem kaskádových stylů (WCSS).

### Příklad

Vazba se souborem kaskádových stylů (*Příklad 1*).

```
<link rel="stylesheet" type="text/css" href="style.css" />
```

Tímto řekneme prohlížeči že má co dočinění se souborem kaskádových stylů (`stylesheet`), jehož mime typ je `text/css` a je umístěn na relativní adrese `style.css`.



link/  
příklad1

### Příklad

Pro mobilní stránky existuje možnost, připojit k dokumentu obrázkovou ikonu (favicon). Tu následně prohlížeč při návštěvě zobrazí (nejčastěji vedle URL).

Definujeme ji následujícím způsobem (*příklad 2*).


```
<link rel="shortcut icon" href="favicon.ico"/>
```

Favicon musí mít rozměry 16x16 nebo 32x32 px. Tyto rozměry platí pro klasické WWW stránky a já je využiji i pro stránky mobilní. Ikonu je možné snadno vyrobit přejmenováním bitmapy na `*.ico`. Není to však úplně korektní, jelikož `ico` je odlišný formát. Funkci ale splní.



link/  
příklad2

### Ukázka

Vytvořil jsem favicon o velikosti 32x32px. 



Emulátor Fennec/Firefox 1.0a1



## 2.9.5 Cizí obsah

### Styly <style>

Párová značka <style></style> byla přidána z důvodu propojení dokumentu a kaskádových stylů. Definiuje se v hlavičce stránka a její platnost je následně celý dokument. V XHTML se jakýkoliv cizí kus kódu umísťuje do bloku CDATA. Atribut type udává použitý druh jazyka.

#### Příklad

Značka style v hlavičce dokumentu.

```
<style type="text/css">
  /* <![CDATA[*/
  /*   ]]> */
</style>
```



style/  
příklad1

Atribut style slouží pro přiřazení interního stylu vybraným elementům.

### Script <script>, noscript <noscript>

Párová značka <script></script> byla přidána z důvodu propojení dokumentu s některým skriptovacím jazykem. Umožňuje vkládání různých druhů skriptů (atribut type), a to buď přímo (CDATA), nebo odkazem (atribut src). Umísťuje se jak do hlavičky stránky (platnost na celý dokument), tak i v rámci těla dokumentu. Párová značka <noscript></noscript> byla přidána v důsledku možnosti propojení s některým skriptovacím jazykem. Interpret XHTML zobrazí její obsah v případě neúspěšného provedení skriptu, jinak bude ignorována. Umísťuje se pouze do těla dokumentu.

Elementy script a noscript jsou součástí jazyka XHTML-MP 1.1.

#### Příklad

Ukázka použití scriptování.

```
<script type="text/javascript">
  /*<![CDATA[ */
  alert("Kliknutím na OK jste zvolil červenou pilulku!");
  /*   ]]> */
</script>
...
<noscript><p>
Váš prohlížeč nepodporuje JavaScript!</p></noscript>
```



script/  
příklad1

## 2.9.6 Blokové vs. řádkové elementy

Jak již název napovídá, značky jazyka jsou dvojího typu, a to blokové a řádkové. Blokové elementy představují souvislý blok (např. odstavec), řádkové pak úsek textu. Snadno je od sebe rozeznáte podle odřádkování. Dva bloky nemohou, z hlediska strukturovaném dokumentu, existovat vedle sebe. Tedy blokový element vždy na konci odřádkuje.

Z předchozího odstavce vyplývá ještě jedna skutečnost. **Řádkové elementy (úseky textu) by měly být součástí nějakého bloku.**

### Příklad

Blokové značky `p` sdružují řádkový obsah `em`, `b`, `i`.

```
<p>Odstavec je typickým příkladem blokového elementu.</p>
<p>Blokové elementy sdružují řádkový obsah.</p>
<p>
  Jsou to <em>úseky textu</em> <b>stojící</b>
  <i>vedle sebe</i> v bloku.
</p>
```



blokvsline/  
priklad1

### Ukázky

**Blok vs řádka**

Odstavec je typickým příkladem blokového elementu.

Blokové elementy sdružují řádkový obsah.

Jsou to *úseky textu* **stojící vedle sebe** v bloku.

Internet Explorer Mobile  
(Windows Mobile 6)

**Blok vs řádka**

Odstavec je typickým příkladem blokového elementu.

Blokové elementy sdružují řádkový obsah.

Jsou to *úseky textu* **stojící vedle sebe** v bloku.

Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## 2.9.7 Formátování textu

### Odstavce <p>

Základním formátovacím elementem dokumentu je bezesporu odstavec (paragraph). Odstavec je značkou blokovou a párovou. Do `<p></p>` lze umístit jakýkoliv řádkový obsah nebo řádkové elementy.

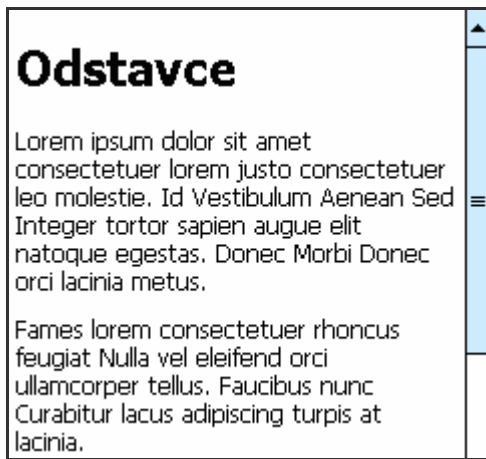
**Příklad**

Definice odstavců.

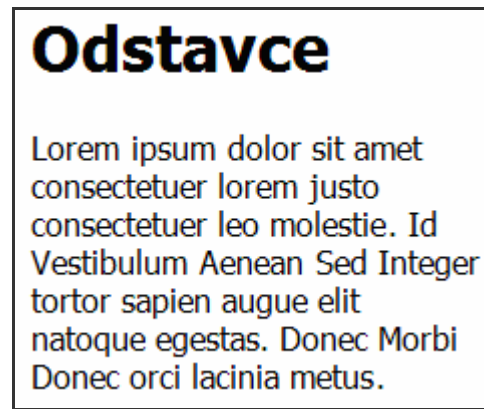
```
<p>Lorem ipsum dolor sit amet ...</p>
<p>Fames lorem consectetur rhoncus ...</p>
<p>Fusce tortor sem laoreet sociis ...</p>
<p>Nibh Sed ut pretium tortor ipsum ...</p>
```



p/  
příklad1

**Ukázky**

Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

**Nadpisy**

K vyznačení nadpisů slouží sada elementů h (heading). Jedná se o párové blokové elementy. Za písmenem h je vždy uveden stupeň (priorita) daného nadpisu. Rozeznáváme 6 druhů nadpisů: <h1>, <h2>, <h3>, <h4>, <h5>, <h6>

**Příklad**

Vizuální podoba nadpisů (velikost písma).

```
<h1>Nadpis první úrovně</h1>
<h2>Nadpis druhé úrovně</h2>
<h3>Nadpis třetí úrovně</h3>
<h4>Nadpis čtvrté úrovně</h4>
<h5>Nadpis páté úrovně</h5>
<h6>Nadpis šesté úrovně</h6>
```

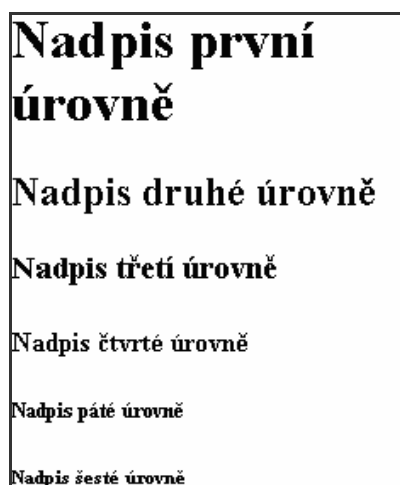


h/  
příklad1

## Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Skyfire 0.9.333  
(Windows Mobile)

Zalomování řádků `<br />`

Potřebujeme-li vizuálně ukončit psaný řádek, využijeme nepárového řádkového elementu `<br />` (break). Element představuje vynucené ukončení řádky.

## Příklad

Řádkování v rámci odstavce.

```
<p>Lorem ipsum dolor <br />
sit amet consectetur consectetur <br />
hendrerit volutpat id Nullam. <br />
Morbi vel mattis at Quisque quis <br />
quis eu id et Curabitur.
</p>
```

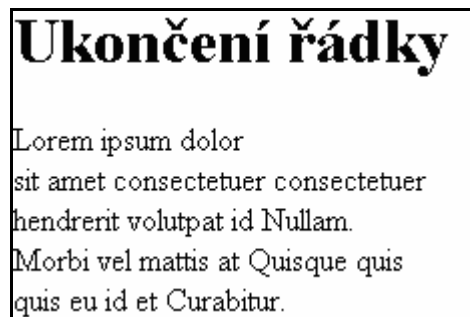


br/  
příklad1

## Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Skyfire 0.9.333  
(Windows Mobile)

## Možné druhy zápisů

Následující příklad ukazuje použití řádkových (in-line) typů zápisu kódu, vzorů, proměnných, strojopisu aj. Většina z nich je velmi málo používaná.

### Příklad

Definice různých druhů zápisů textu.

```
<p>
  <code>Výpis kódu (code).</code>
  <dfn>Definice nového výrazu (dfn)</dfn>
  <kbd>Strojopis (kbd)</kbd>
  <var>Proměnná (var)</var>
  <samp>Výstup programu (samp)</samp>
</p>
```



font/  
příklad3

### Ukázky

#### Různé druhy zápisů

```
Výpis kódu (code).
Definice nového výrazu (dfn)
Strojopis (kbd)
Proměnná (var)
Výstup programu (samp)
```

Internet Explorer Mobile  
(Windows Mobile 6)

#### Různé druhy zápisů

```
Výpis kódu (code).
Definice nového výrazu (dfn)
Strojopis (kbd)
Proměnná (var)
Výstup programu (samp)
```

Opera 9.5 beta  
(Windows Mobile - Touchscreen)

### Předformátovaný text <pre>

V jazyce XHTML MP nemáme běžně možnost manipulace s tzv. bílými znaky. Pokud několikrát odřádkujeme, nebo umístíme několik mezer za sebou, bílý prostor se spojí a vznikne nám jediná mezera. Je to z důvodu potřeby formátování kódu dokumentu. Pokud bychom chtěli sami ovlivnit, kolik bílých znaků v textu použijeme, museli bychom vše uzavřít do blokového elementu <pre></pre>.

### Příklad

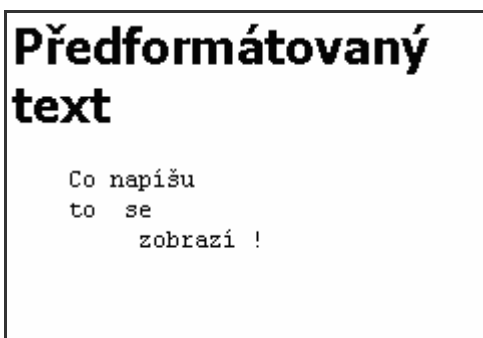
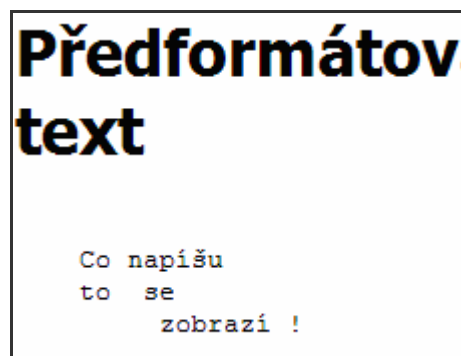
Zachování bílých znaků v textu.

```
<pre>
Co napíšu
to se
    zobrazí !
</pre>
```



font/  
příklad4

## Ukázky

Internet Explorer Mobile  
(Windows Mobile 6)Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Zkratky

Zápis zkratk se provádí za pomoci párových řádkových (in-line) elementů `<abbr><abbr>` a `<acronym></acronym>`. Abbr představuje definici zkratky, jejíž plný název se uvádí v atributu `title`. Často se zobrazuje s tečkovým podtržením. Acronym zastupuje zkratku, která se nehlašuje.

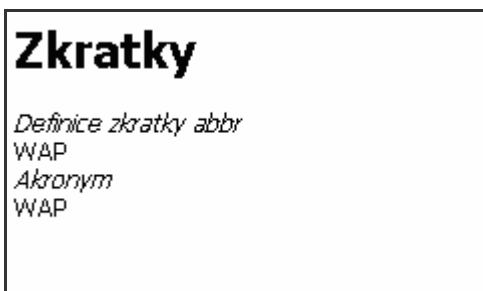
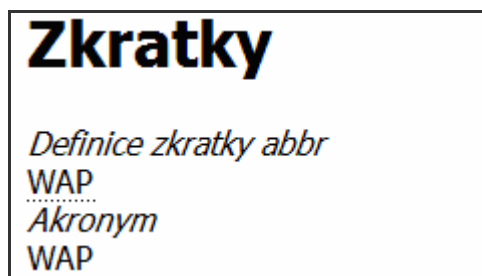
## Příklad

Sémantické zvýraznění.

```
<p>
  <em>Definice zkratky abbr</em>
  <abbr title="Wireless Application Protocol">WAP</abbr>
  <em>Akronym</em>
  <acronym>WAP</acronym>
</p>
```

zkratky/  
příklad1

## Ukázky

Internet Explorer Mobile  
(Windows Mobile 6)Opera 9.5 beta  
(Windows Mobile - Touchscreen)Adresa `<address>`

Párová značka `<address></address>` vymezuje prostor pro popis adresy. Zobrazuje se stejně jako odstavec, její písmo je však zkosené (kurzíva). Jde o blokový prvek.

**Příklad**

Adresa ve stránce.

```
<address>
Franta Vávra, Horní dolní 52, Za bukem 1234
</address>
<address>Franta Vávra, (zabukem@hornidolni.cz)</address>
```

address/  
priklad1**Ukázky**

**Adresa**

**Bydliště**

*Franta Vávra  
Horní dolní 52  
Za bukem 1234*

**E-mail**

*Franta Vávra, (zabukem@hornidolni.cz)*

Internet Explorer Mobile  
(Windows Mobile 6)

**Adresa**

**Bydliště**

*Franta Vávra  
Horní dolní 52  
Za bukem 1234*

**E-mail**

*Franta Vávra,  
(zabukem@hornidolni.cz)*

Opera 9.5 beta  
(Windows Mobile - Touchscreen)**Citace**

Pro potřeby citování textu vznikly značky:

- bloková citace `<blockquote></blockquote>`
- řádkové citace `<q></q>`, `<cite></cite>`

**Bloková citace** se používá pro citování delších bloků textu. Jak název napovídá, jedná se o blokový element. Citovaný text se vyznačuje větším krajním odsazením. Obsahuje atribut `cite`, který definuje zdroj citovaných dat.

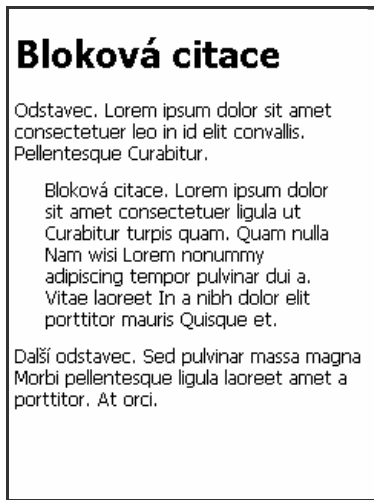
**Příklad**

Definice blokové citace.

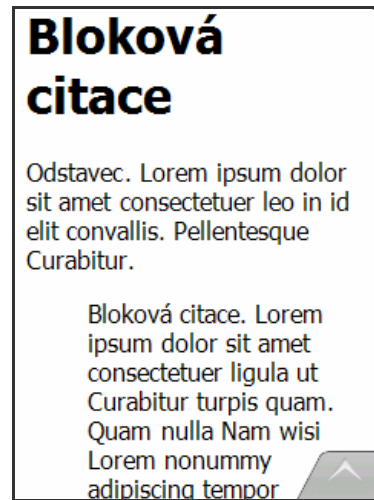
```
<blockquote cite="http://webmobile.comuv.com">
  <p>Bloková citace ...</p>
</blockquote>
```

citace/  
priklad1

## Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

**Řádková citace** se používá pro citování menších úseků textu. Lze to provést za pomoci in-line značek `<q></q>` (quote) a `<cite></cite>`. Element `q` je novější a oproti `<cite>` obsahuje atribut `cite`. Prohlížeče zobrazují oba elementy zkoseným písmem (kurzívou), k značce `q` pak často přidávají uvozovky.

## Příklad

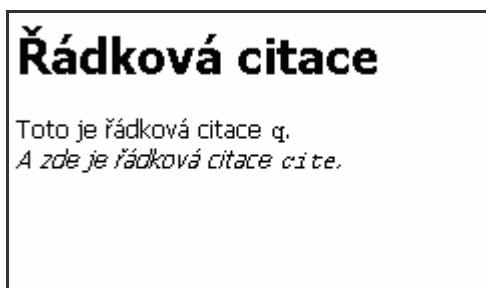
Deklarace řádkových citací.

```
<p>
  <q cite="URL">Toto je řádková citace q.</q>
  <cite>A zde je řádková citace cite.</cite>
</p>
```

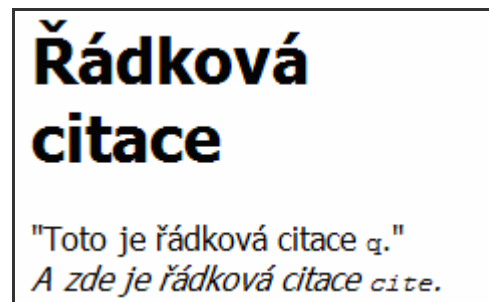


citace/  
příklad2

## Ukázka



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Zdůraznění písma

Pro sémantické zvýraznění požadovaných úseků textu využíváme párových řádkových elementů `<em>` a `<strong>`.



- `<em>` znamená zvýraznění/ důraz (emphasis).
- `<strong>` znamená silné zdůraznění

Vizuálně se tyto značky formátují - `<em>` jako kurzíva `<i>`, `<strong>` jako tučné písmo `<b>`. Je zde právě ona sémantická složka, která je významně rozlišuje. Pokud na element `<em>` narazí hlasový syntezátor, zvýší vzápětí hlas. Naopak `<i>` by měl přejít bez povšimnutí.

### Příklad

V příkladu jsou proti sobě postaveny sémantická zdůraznění a jejich ekvivalenty z prezentačního modulu.



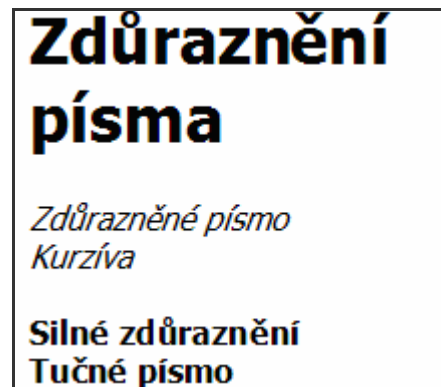
font/  
příklad2

```
<p>
  <em>Zdůrazněné písmo</em><br />
  <i>Kurzíva</i><br /><br />
  <strong>Silné zdůraznění</strong><br />
  <b>Tučné písmo</b>
</p>
```

### Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

### Div, span

Párové elementy `<div></div>` a `<span></span>` byly zahrnuty do jazyka XHTML MP pro potřeby označení/vymezení požadovaných úseků dokumentu. Nenesou žádnou sémantickou hodnotu! Smyslem jejich existence je společné použití s kaskádovými styly. `Div` je elementem blokovým, `span` naopak elementem řádkovým (in-line).

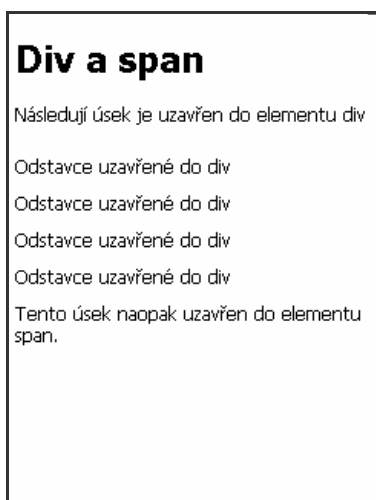
**Příklad**

Vizuálně se elementy `div` a `span` nijak neprojevují, pouze `div` odřádkuje.

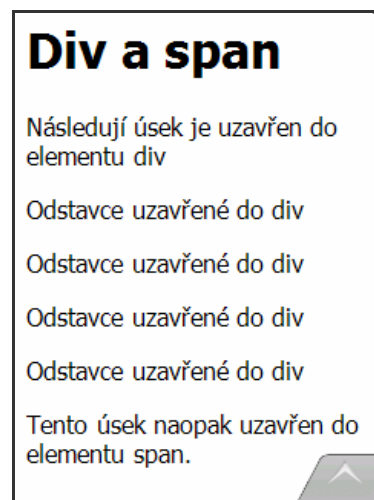
```
<div>
  <p>Odstavce uzavřené do div</p>
  <p>Odstavce uzavřené do div</p>
  <p>Odstavce uzavřené do div</p>
  <p>Odstavce uzavřené do div</p>
</div>
<p>
  <span>Tento úsek naopak uzavřen do elementu span.</span>
</p>
```



style/  
příklad2

**Ukázky**

Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

**2.9.8 Vizuální změna písma****Fonty**

Pro vizuální změnu písma využíváme značky prezentačního modulu. Jedná se o sadu párových řádkových elementů. **Tyto elementy nenesou žádnou sémantickou hodnotu!** Existují pouze pro potřebu vizuálního formátování, k tomuto účelu však slouží kaskádové styly (WCSS).

Jejich používání se v rámci sémantického webu nedoporučuje, v budoucnu může být bráno za chybu.

**Příklad**

Značka `<b>` představuje tučné písmo, `<i>` kurzívu, `<small>` zmenšení písma o jeden stupeň, `<big>` potom zvětšení písma o jeden stupeň.



font/  
příklad1

```

<p>
  <b>Tučné písmo</b><br />
  <i>Kurzíva</i><br />
  <small>Zmenšené písmo</small><br />
  <big>Zvětšené písmo</big><br />
  <b><i>Tučná kurzíva</i></b>
</p>

```

### Ukázky

**Vizuální změna písma**

Tučné písmo  
*Kurzíva*  
 Zmenšené písmo  
 Zvětšené písmo  
***Tučná kurzíva***

Internet Explorer Mobile  
(Windows Mobile 6)

**Vizuální změna písma**

Tučné písmo  
*Kurzíva*  
 Zmenšené písmo  
 Zvětšené písmo  
***Tučná kurzíva***

Opera 9.5 beta  
(Windows Mobile - Touchscreen)

### Vodorovná čára <hr />

Za zkratkou hr se skrývá anglický název *horizontal rule*, neboli vodorovná čára.

Jedná se o blokový nepárový element.

#### Příklad

Definice čar v textu.

```

<hr />
<p>Lorem ipsum dolor sit .... p>
<hr />
<p>Sed Vestibulum est ... </p>
<hr />
<p>Tortor id Ut consectetuer ... </p>

```



hr/  
příklad1

### Ukázky

**Vodorovná čára**

---

Lorem ipsum dolor sit amet consectetuer felis pellentesque malesuada Integer sed. Vestibulum Donec Ut.

---

Sed Vestibulum est libero Pellentesque risus vitae ut eu pellentesque.

---

Tortor id Ut consectetuer consectetuer dui pede Vestibulum turpis tincidunt Donec. Sociis id convallis.

Internet Explorer Mobile  
(Windows Mobile 6)

**Vodorovná čára**

---

Lorem ipsum dolor sit amet consectetuer felis pellentesque malesuada Integer sed. Vestibulum Donec Ut.

---

Sed Vestibulum est libero Pellentesque

Skyfire 0.9.333  
(Windows Mobile)

## 2.9.9 Seznamy

V jazyce XHTML MP rozeznáváme tři typy seznamů. Značka `<ul></ul>` představuje neuspořádaný seznam (unordered list), značka `<ol></ol>` naopak seznam uspořádaný (ordered list). Třetím z řady je seznam definic (definition list) `<dl></dl>`.

### Uspořádané seznamy `<ol></ol>`

se vyznačují číslováními položkami. Jednotlivé položky jsou uzavřeny do elementu `<li>` (list item). Číslo položek je možné ovlivnit atributem `start` elementu `<ol>`, další možností je potom atribut `value` elementu `<li>`.

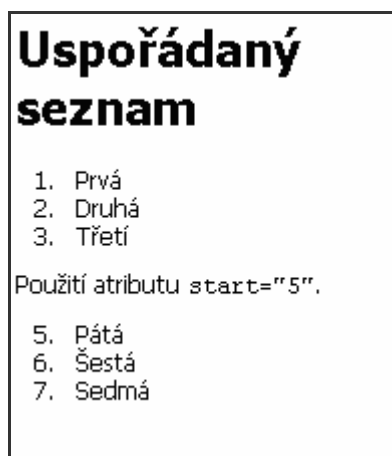
#### Příklad

Číslovaný (uspořádaný) seznam a číslovaný seznam s použitím `start` atributu. Start atribut představuje počáteční hodnotu číslování (od jaké hodnoty se má číslovat).

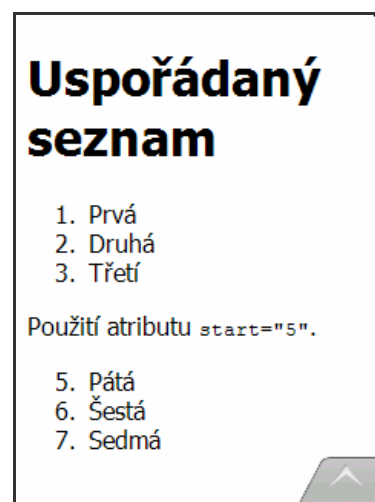


```
<ol>
  <li>Prvá</li>
  <li>Druhá</li>
  <li>Třetí</li>
</ol>
<p>Použití atributu start="5".</p>
<ol start="5">
  <li>Pátá</li>
  <li>Šestá</li>
  <li>Sedmá</li>
</ol>
```

#### Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)



## Příklad

Uspořádaný seznam s použitou hodnotou `value` a `start`. Hodnota `value` udává číslo zobrazované před položkou seznamu, čísla dalších položek se od něho následně odvíjejí. Atributy `value` a `start` se dají kombinovat, v tom případě má větší prioritu element `li`.

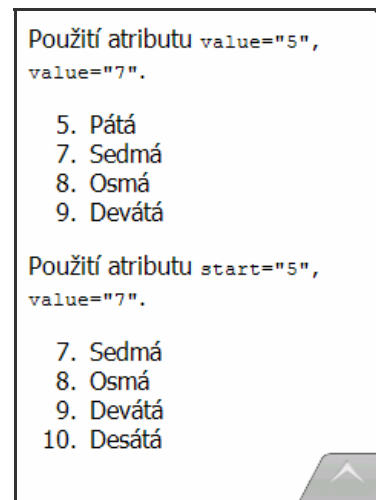
```
<p>Použití atributu value="5", value="7".</p>
<ol>
  <li value="5">Pátá</li>
  <li value="7">Sedmá</li>
  <li>Osmá</li>
  <li>Devátá</li>
</ol>

<p>Použití atributu start="5", value="7".</p>
<ol start="5">
  <li>Pátá</li>
  <li value="7">Sedmá</li>
  <li>Osmá</li>
  <li>Osmá</li>
</ol>
```

## Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Neuspořádaný seznam `<ul></ul>`

umísťuje před své položky kulaté značení. Ty se taktéž umísťují do elementu `<li></li>`.

**Příklad**

Deklarace neuspořádaného seznamu.

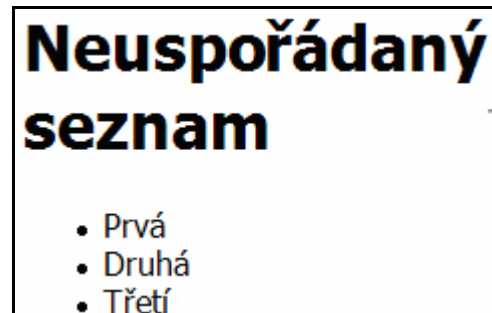
```
<ul>
  <li>Prvá</li>
  <li>Druhá</li>
  <li>Třetí</li>
</ul>
```



seznamy/  
příklad3

**Ukázky**

Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

**Seznam definic <dl></dl>**

je zvláštním druhem seznamu. Jeho položky se zapisují střídavě ve tvaru <dt></dt> a <dd></dd>. Dt představuje definici termínu, dd jeho vysvětlení.

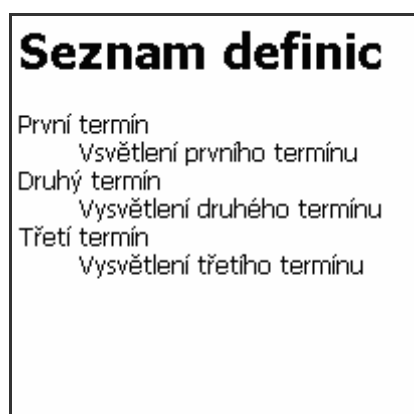
**Příklad**

Deklarace seznamu definic.

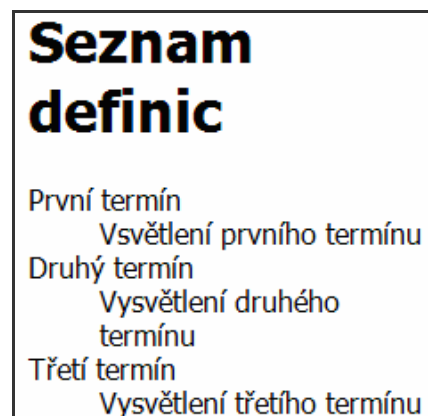
```
<dl>
  <dt>První termín</dt>
  <dd>Vysvětlení prvního termínu</dd>
  <dt>Druhý termín</dt>
  <dd>Vysvětlení druhého termínu</dd>
</dl>
```



seznamy/  
příklad4

**Ukázka**

Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## 2.9.10 Obrázky

### Element `<img />`

Obrázek vložíme do stránky nepárovým elementem `<img />`.

Zapisuje se ve tvaru:

```

```

Z důvodu úspory místa bychom měli používat běžné internetové formáty, tedy png, jpg a gif. Pokud některý formát prohlížeč neumí, zobrazí obsah atributu `alt`, který definuje alternativní text. Zdrojové umístění souboru uvedeme v atributu `src` (source).

Rozměry obrázků nastavujeme atributy `width` a `height`. Ovlivňujeme tím pouze vykreslení v rámci dokumentu, skutečné rozměry zůstanou nezměněny. Pokud nastavíme hodnotu jednomu z těchto atributů, druhý se dle poměru stran skutečných rozměrů přepočítá. Nastavíme-li atributy oba, je možná deformace vykresleného obrazu. Obrázek je řádkový (in-line) element!

### Příklad

Běžně používané WWW formáty obrázků.

```
<p>
</p>
<p>
</p>
<p>
</p>
```



img/  
příklad1

### Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

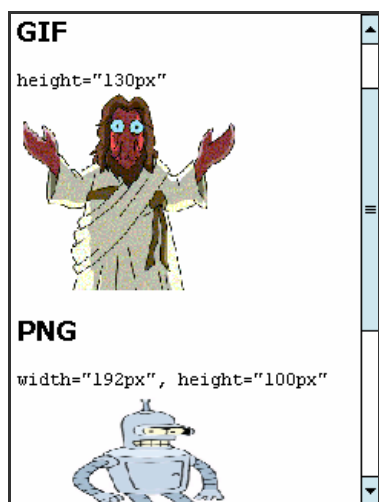
**Příklad**

Nastavení rozměrů a ukázka deformace obrázků.

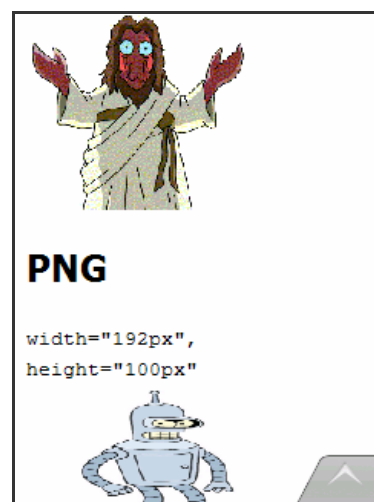
```
<p>
</p>
<p>
</p>
<p>
</p>
```



img/  
priklad2

**Ukázky**

Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

**Příklad**

Další možnost jak popsat obrázek v dokumentu přináší atribut `longdesc`, jehož hodnota odkazuje na soubor s požadovanými informacemi.

```
<p></p>
```



img/  
priklad3

**2.9.11 Tabulky**

Tabulky sdružují informace, které mají určitou logickou souvislost. Dříve se za jejich pomoci vytvářelo rozvržení dokumentů (layout). Do jedné buňky se umístilo menu, do druhé např. obsah, byl to typický příklad chybného použití!



## Struktura tabulek

- Tabulku začíná a končí element `<table></table>`. Tomu je možné nastavit několik užitečných atributů, atributem `width` lze nastavovat šířku tabulky, atribut `summary` slouží pro stručné shrnutí obsahu.
- Za značkou `table` následuje popis tabulky `<caption></caption>`.
- Jednotlivé řádky reprezentují elementy `<tr></tr>`.
- Ty seskupují množinu buněk označených buď `<td></td>`, nebo `<th></th>`. `Td` označuje buňku jako takovou, `th` reprezentuje hlavičku buňky, liší se pouze ve vykreslení obsahu - obsah `th` se zobrazuje vycentrovaný a tučným písmem.
- Pomocí hodnot atributů `valign` a `align`, definovaných pro jednotlivé řádky (`tr`), můžeme ovládat vertikální a horizontální zarovnání buněk v každém z nich. Lze je také uvést samostatně pro každou buňku zvlášť.
- Často používanými atributy jsou tzv. přesahy do okolních oblastí, jedná se o atributy `rowspan` (přesah v rámci řádku) a `colspan` (přesah v rámci sloupce).

Atribut `border` není v elementu `table` povolen. Pro lepší orientaci a pochopení problematiky používám orámování za pomoci kaskádových stylů.

### Příklad

Tabulka s popisem, elementy `th` a `td`, atributem `summary` a zadanou šířkou.

```
<table summary="Ukázková tabulka ..." width="200px">
<caption>Popisek tabulky</caption>
  <tr>
    <th>Hlav. buňka 1</th>
    <td>Buňka 0.1</td>
    <td>Buňka 0.2</td>
  </tr>
  <tr>
    <th>Hlav. buňka 2</th>
    <td>Buňka 1.1</td>
    <td>Buňka 1.2</td>
  </tr>
</table>
```



table/  
příklad1

## Ukázky

**Tabulka**

Tabulka s popiskem, buňkami (th, td), atributem `summary` a zadanou šířkou.

Popisek tabulky

<b>Hlav. buňka 1</b>	Buňka 0.1	Buňka 0.2
<b>Hlav. buňka 2</b>	Buňka 1.1	Buňka 1.2

Internet Explorer Mobile  
(Windows Mobile 6)

**Tabulka**

Tabulka s popiskem, buňkami (th, td), atributem `summary` a zadanou šířkou.

Popisek tabulky

<b>Hlav. buňka 1</b>	Buňka 0.1	Buňka 0.2
<b>Hlav. buňka 2</b>	Buňka 1.1	Buňka 1.2

Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Příklad

Přesah buněk v rámci sloupců (Buňka 0.1) a řádků (Buňka 0.2).

```
<tr>
  <th>Hlav. buňka 1</th>
  <td colspan="3">Buňka 0.1</td>
  <td rowspan="2">Buňka 0.2</td>
</tr>
```



table/  
příklad2

## Ukázky

**Tabulka**

Tabulka s atributy `colspan` a `rowspan`.

Popisek tabulky

<b>Hlav. buňka 1</b>	Buňka 0.1			Buňka 0.2
<b>Hlav. buňka 2</b>	Buňka 1.1	Buňka 1.2	Buňka 1.3	

Internet Explorer Mobile  
(Windows Mobile 6)

**Tabulka**

Tabulka s atributy `colspan` a `rowspan`.

Popisek tabulky

<b>Hlav. buňka 1</b>	Buňka 0.1			Buňka 0.2
<b>Hlav. buňka 2</b>	Buňka 1.1	Buňka 1.2	Buňka 1.3	

Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Příklad

Příklad znázorňuje zarovnání obsahu buněk atributy `align` (left, right, center) a `valign` (middle, top, bottom, baseline). Vertikální zarovnání lze nastavit pro celý řádek u elementu `tr`.



table/  
příklad3

```

<th>Svisle</th>
  <td rowspan="2" valign="top">TOP</td>
  <td rowspan="2" valign="bottom">BOTTOM</td>
  <td rowspan="2" valign="middle">MIDDLE</td>
  <td rowspan="2" valign="baseline">BASELINE</td>

<th rowspan="2">Vodorovně</th>
  <td colspan="2">NORMAL</td>
  <td colspan="2" align="left">LEFT</td>
  <td colspan="2" align="center">CENTER</td>
  <td colspan="2" align="right">RIGHT</td>

```

## Ukázky

<b>Tabulka</b>				
Zarovnání obsahu buněk				
Popisek tabulky				
<b>Svisle</b>	NORMAL	TOP	BOTTOM	MIDDLE
<b>Hl.buňka</b>				
<b>Vodorovně</b>	NORMAL		LEFT	
	CENTER		RIGHT	

Internet Explorer Mobile  
(Windows Mobile 6)

Buňky tabulky obsahují kromě probraných atributů ještě skupinu dalších, velmi málo užívaných. Uvedu zde pouze jejich názvy: *axis*, *scope*, *abbr*, *headers*.

## 2.9.12 Odkazy

### Element <a>

Jedním ze základních cílů značkovacích jazyků (kromě strukturovaného značení) byla schopnost vytvořit vazby mezi jednotlivými dokumenty. Tuto nutnost zajistil párový in-line element <a></a> (anchor). Hlavním atributem je *href*, jehož hodnota odkazuje na požadovanou pozici. Pozicí může být cizí dokument, konkrétní místo v dokumentu stávajícím, nebo konkrétní místo v dokumentu cizím.

Mezi další atributy patří *title* (titulek odkazu), *charset* (kódování odkazovaného dokumentu), *hreflang* (jazyk odkazovaného dokumentu), *rel* (specifikuje vztah odkazovaného dokumentu k aktuálnímu dokumentu), *rev* (specifikuje vztah aktuálního dokumentu k odkazovanému dokumentu), *type* (MIME typ odkazovaného dokumentu).

Posledními dvěma atributy jsou `tabindex` a `accesskey`. `Tabindex` udává tabulační hodnotu elementu na stránce (pořadí přechodu odkazů za pomoci klávesnice - většinou tabulátoru), jeho hodnotou je číslo od 0 – 32767. `Accesskey` představuje přístupovou klávesu odkazu, po jejímž stisknutí odkaz aktivujeme. Hodnotou je symbol požadovaného tlačítka.

Odkazy mohou být buď absolutní, nebo relativní. Relativní adresa se odvozuje od pozice adresáře, ve kterém je stránka umístěna. Absolutní udává plnou URL adresu.

### Příklad

Odkaz s atributy `hreflang`, `charset`, `type`, `title`.

```
<a href="hello.xhtml" hreflang="en" charset="utf-8"
type="application/xhtml+xml" title="Lorem ipsum
dokument">Odkaz</a>
```



a/  
příklad1

### Ukázky

## Odkazy

Odkaz s atributy `title`, `hreflang`,  
`type`, `charset`.

[Odkaz1](#)  
[Odkaz2](#)  
[Odkaz3](#)  
[Odkaz4](#)

## Hello body

Lorem ipsum dolor sit amet  
consectetuer congue volutpat Phasellus  
nec ac. Tristique felis ante ligula Proin  
enim pede ipsum Nunc Phasellus eget.  
Nibh id semper ante dolor sem Aenean  
Aenean nulla dui volutpat. Quis felis non  
libero senectus ante id Sed justo elit

Internet Explorer Mobile  
(Windows Mobile 6)

### Příklad

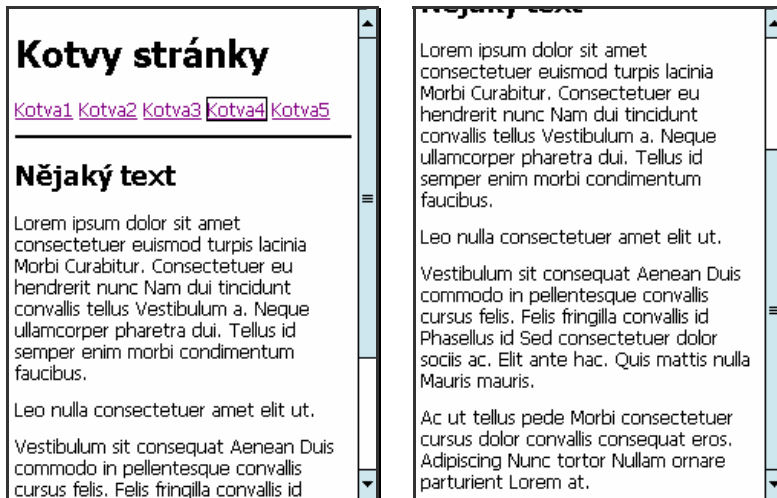
Jak již bylo řečeno, odkazy můžeme směřovat na určitá místa aktuálního dokumentu (tzv. kotvy). Odkazovaným místem je element s určeným atributem ve tvaru `id="jméno"`, jeho hodnotu uvedeme jako `href="#jméno"`.

```
<a href="#nadpis">Kotva1</a>
...
<h2 id="nadpis">Nějaký text</h2>
```



a/  
příklad2

## Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)

## Příklad

Na určená místa cizího dokumentu se odkazujeme pomocí zápisu  
`href="URL_adresa#jméno"`, kde jméno je hodnotou id atributu.



a/  
příklad3

Aktuální dokument:

```
<a href="hello.xhtml#nadpis">Kotva1</a>
```

V hello.xhtml dokumentu:

```
<h1 id="nadpis">Hello body</h1>
```

## Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)

## Příklad

Definice `tabindex` (hodnoty na sebe nemusí navazovat, pokud se vyskytuje stejná hodnota, zohledňuje se pořadí v dokumentu) a `accesskey`.



a/  
příklad4

```

<a href="hello.xhtml" tabindex="1" accesskey="0" >Odkaz (0) </a>
<a href="hello.xhtml" tabindex="3" accesskey="1" >Odkaz (1) </a>
<a href="hello.xhtml" tabindex="5" accesskey="2" >Odkaz (2) </a>
<a href="hello.xhtml" tabindex="7" accesskey="3" >Odkaz (3) </a>
<a href="hello.xhtml" tabindex="9" accesskey="4" >Odkaz (4) </a>
<a href="hello.xhtml" tabindex="11" accesskey="5" >Odkaz (5) </a>
<a href="hello.xhtml" tabindex="12" accesskey="6" >Odkaz (6) </a>
<a href="hello.xhtml" tabindex="10" accesskey="7" >Odkaz (7) </a>
<a href="hello.xhtml" tabindex="8" accesskey="8" >Odkaz (8) </a>
<a href="hello.xhtml" tabindex="6" accesskey="9" >Odkaz (9) </a>
<a href="hello.xhtml" tabindex="4" accesskey="*" >Odkaz (*) </a>
<a href="hello.xhtml" tabindex="2" accesskey="#" >Odkaz (#) </a>

```

## Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)

## 2.9.13 Základní adresování odkazu

### Element `<base />`

Nepárový element `<base />` udává základní umístění, od kterého se odvozují všechny relativní odkazy ve stránce. Základní URL adresu uvádíme v atributu `href`.

### Příklad

Definice výchozí umístění relativních odkazů.

```

<base href="Absolutní_URL_adresa/adresář/" />
...
<a href="hello.xhtml" title="Lorem ipsum">Odkaz1</a>

```



base/  
příklad1

## 2.9.14 Objekty stránky

### Element `<object>`

Párový in-line element `<object></object>` umožňuje vkládat do stránky libovolné objekty, které jsou umístěny v externích souborech (obrázky, zvuk, video...). Jeho obsahem je sada nepárových atributů `<param />`, které slouží k nastavení parametrů objektu. Vše další se bere jako alternativní text, ten se vypíše v případě neúspěšného načtení objektu. Prohlížeče samy o sobě zobrazují pouze základní druhy objektů (obrázky JPG,PNG,GIF, textové soubory), na ostatní vyžadují spuštění externích aplikací (mpeg).

Objekty se do sebe mohou navzájem vnořovat. Pokud se nepodaří načtení prvního, vyzkouší se druhý atd.

### Atributy elementu `object`:

- `data` je nejdůležitějším atributem elementu `object`, obsahuje URL adresu se souborem spustitelného objektu (je možné uvést přímo zdrojový kód – nepoužívá se)
- `classid` slouží pro definování externí aplikace, která bude objekt zpracovávat. Zadává se URL, nebo odkaz do registru.
- `type` udává MIME typ dat, deklarovaných v atributu `data`
- `codetype` udává MIME typ dat, deklarovaných v atributu `classid`
- `codebase` nastavuje výchozí URL adresu, od které budou odvozovány všechny relativní adresy v `object`
- `archive` hodnota představuje sadu URL odkazů oddělených mezerami, na kterých se nachází archivy uvedených objektů (různé soubory potřebné při běhu aplikace atd.)
- `tabindex` představuje tabulační hodnotu objektů
- `standby` může reprezentovat text, který se předloží uživateli při nahrání objektu
- `name` je jméno objektu (zpracování formulářem)

- `width` a `height` udávají rozměry objektu, můžou být v px nebo v % dostupné šířky

#### Atributy elementu `param`:

- `name` je jméno parametru, které předáváme objektu
- `value` představuje hodnotu nastaveného parametru
- `valuetype` specifikuje typ hodnoty parametru, jsou 3 možnosti:
  - `ref` – URL adresa zdroje s parametry
  - `data` – hodnota po vyhodnocení předána objektu jako řetězec
  - `object` – odkaz na již definovaný objekt v dokumentu
- `type` udává MIME typ URL zdroje uvedeného v `ref`

#### Příklad

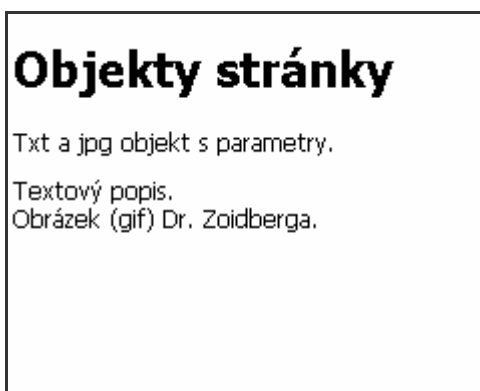
Vložení objektu `popis.txt` a `futurama.gif` s možnými parametry.

```
<object data='popis.txt' type='text/plain'>
  <param name="parametr1" VALUE="hodnota1" />
  <param name="parametr2" VALUE="hodnota2" />
  Textový popis.
</object>
<object data='futurama.gif' type='image/gif'>
  <param name="parametr1" VALUE="hodnota1" />
  <param name="parametr2" VALUE="hodnota2" />
  Obrázek (gif) Dr. Zoidberga.
</object>
```



object/  
příklad1

#### Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)



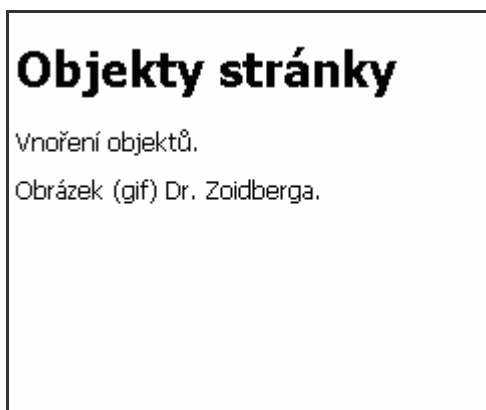


## Příklad

Ukázka vnoření objektů, soubory `sample.*` neexistují. Prohlížeč se nejprve pokusí načíst první z nich (`sample.mpeg`), neuspěje a přejde do těla elementu `object`. Parametr vždy patří k objektu, jehož je dítětem.

```
<object data='sample.mpeg' type='video/mpeg'>
  <param name='start' value='10' />
  <param name='loop' value='false' />
  <object data='sample.txt' type='text/plain'>
    <object data='futurama.gif' type='image/gif'>
      Obrázek (gif) Dr. Zoidberga.
    </object>
  </object>
</object>
```

## Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

### 2.9.15 Události

Jazyk XHTML-MP od verze 1.1 dovoluje popis reakcí na různé události vyvolané uživatelem. Události se přiřazují k jednotlivým elementům, přičemž různé elementy mohou také podporovat různé události. Reakční hodnotou je většinou nějaký skript, který se po zavolání provede.

Zápis: `jméno_události="script"`

**Zde je uveden pouze přehled dostupných událostí:**

`onblur`, `onfocus`, `onreset`, `onsubmit`, `onload`, `onunload`,  
`onchange`, `onselect`, `onclick`, `ondblclick`, `onmousedown`,

onmouseup, onmouseover, onmousemove, onmouseout,  
onkeypress, onkeydown, onkeyup

### 2.9.16 Formuláře

Formuláře vznikly za účel lepší interakce dokumentu a uživatele, toho je docíleno jednoduchými prvky uživatelského grafického rozhraní (GUI).

- Vše začíná a končí párovým elementem `<form></form>`.
- Uvnitř se nalézají ovládací prvky formulářů.
  - vstupní pole `<input></input>`
  - prvek výběru `<select></select>`
  - položka výběru `<option></option>`
  - tlačítko `<button></button>`
  - rozsáhlejší textové pole `<textarea></textarea>`
  - popisek `<label></label>` aj.

Ovládací prvky je možné sdružovat. Skupinu polí sdružuje element `<fieldset></fieldset>`, volby v značce `select` sdružuje element `<optgroup></optgroup>`. Posledním elementem je popisek skupiny polí (`fieldset`) `<legend></legend>`.

Jazyk XHTML-MP 1.2 již využívá plný formulářový modul, já však zůstanu u modulu základního a uvedu pouze některé užitečné příklady z plné verze. Základní modul je typický pro obě předchozí vydání jazyka.

Všechny prvky mají některé společné vlastnosti, jsou jimi atributy: `class`, `id`, `title`.

#### Atributy **form**:

- `action` určuje URL adresu skriptu, na kterou se budou data předávat a následně podle ní zpracovávat. Pokud uvedena není, odešlou se data téže stránce.
- `method` specifikuje způsob, kterým se data skriptu předávají. Existují dvě hodnoty:

- `get` – data jsou odeslána pomocí URL ve tvaru `jméno=hodnota`. Blok hodnot je umístěn za znakem „?“ , jednotlivé hodnoty jsou odděleny znakem „&“. Tato hodnota je implicitní.
- `post` – data jsou začleněna do těla HTTP dotazu.
- `enctype` určuje způsob kódování dat formuláře pomocí MIME typu. Uvádí se pro metodu odesílání `post`. (např. `multipart/form-data`, `text/plain` aj.). U `get` nabývá implicitní hodnoty `application/x-www-form-urlencoded`.

### Atributy `input`:

Hlavním atributem je `type`, jeho hodnotou je typ ovládacího prvku.

- `type="text"` a `type="password"` představují jednořádkové textové pole
- `type="checkbox"` a `type="radio"` reprezentují zatrhávací políčka
- `type="submit"`, `type="reset"`, `type="button"` vytvářejí tlačítka určité funkcionality
- `type="image"` vytváří tlačítko tvořené obrázkem
- `type="hidden"`
- `type="file"` reprezentuje pole určené k připojení souboru

Každý typ má specifické atributy, většina z následujících je však v rámci `input` společná.

- `name` udává jméno ovládacího prvku pro účel odeslání formuláře
- `accesskey` je přístupová klávesa ovládací prvku
- `tabindex` představuje tabulační hodnotu prvku
- `size` nastavuje rozměry prvku
- `value` udává počáteční hodnotu prvku
- `maxlength` zastupuje maximální počet znaků v prvku (`text`, `password`)

- checked a jeho stejnojmenná hodnota zajistí, že zatrhávající políčko bude implicitně zatržené (checkbox, radio)
- src specifikuje URL obrázku, který se má použít při vytvoření tlačítka (submit, reset, button)

### Příklad

Text je odesílán metodou get (implicitní hodnota) s MIME text/plain. Nebyla uvedena hodnota atributu action, stránka se bude odesílat sama na sebe. Je zde textové pole (type="text", také implicitní hodnota), pole pro vložení hesla (type="password"), odesílací tlačítko (type="submit") a obnovovací tlačítko (type="reset"). Po odeslání formuláře se do URL řádky přidají příslušné hodnoty. Tlačítkem reset obnovíte původní konfiguraci ovládacích prvků. Prvky mají nastavenou přístupovou a tabulační klávesu. (Tlačítko type="button" nemá bez použití událostí žádnou funkci.)



form/  
příklad1

```
<form action="" method="get" enctype="text/plain">
<p>
<input type="text" value="Uživatelske jmeno" name="jmeno"
      size="25" accesskey="1" tabindex="3087" />
<input type="password" value="heslo" name="heslo"
      maxlength="6" size="15" accesskey="2" tabindex="4000"/>
</p><p>
<input type="submit" value="Odeslat" />
<input type="reset" value="Obnovit" /></p>
</form>
```

URL: ?jmeno=Uživatelske+jmeno&heslo=heslo

### Ukázky

**Formuláře**

Input type="text",  
type="password",.

Uživatelske jmeno  
\*\*\*\*\*

type="submit", type="reset".

Odeslat Obnovit

Internet Explorer Mobile  
(Windows Mobile 6)

**Formuláře**

Input type="text",  
type="password",.

Uživatelske jmeno  
\*\*\*\*\*

type="submit", type="reset".

Odeslat Obnovit

Opera 9.5 beta  
(Windows Mobile - Touchscreen)

form/  
příklad2

### Příklad

Ukázka zatrhvacích polí `type="checkbox"` a `type="radio"`. Atribut `checked` udává počáteční zatržení. Jméno + hodnota vybraných položek se odešlou jako data formuláře.

```
<p>Co vyhledáváte na internetu?</p>
<p>
<input type="checkbox" name="hudba" value="true" />Hudbu</p>
...
<p>Jaké zvíře je nejtěžší?</p>
<p><input type="radio" name="nejtezsi" value="slon"
checked="checked"/> Slon</p>
```

### Ukázky

Input type="checkbox",  
Co vyhledáváte na internetu?  
 Hudbu  
 Filmy  
 Fóra  
 Souhlasím se zveřejněním údajů.  
type="radio".  
Jaké zvíře je nejtěžší?  
 Slon  
 Medvěd  
 Žirafa  
Odeslat Obnovit

Internet Explorer Mobile  
(Windows Mobile 6)

Co vyhledáváte na internetu?  
 Hudbu  
 Filmy  
 Fóra  
 Souhlasím se zveřejněním údajů.  
type="radio".  
Jaké zvíře je nejtěžší?  
 Slon  
 Medvěd

Opera 9.5 beta  
(Windows Mobile - Touchscreen)

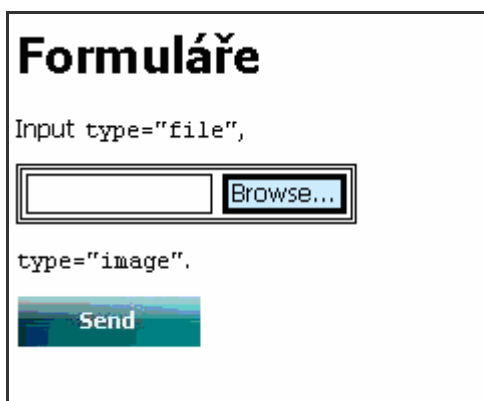
### Příklad

Ukázka ovládacího prvku `file` (`type="file"`), který vyvolá nabídku pro vybrání souboru a obrázkového tlačítka (`type="image"`). U ovládacího prvku `file` se neudává hodnota `value` (z bezpečnostních důvodů) a je potřeba odesílat formulář s `mime multipart/form-data`. Typ `file` je dostupný pouze v XHTML-MP1.2

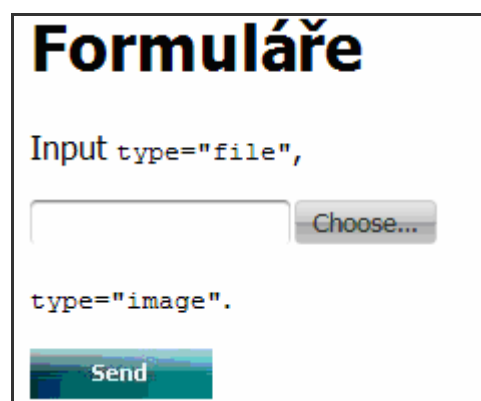
```
<form action="" method="post" enctype="multipart/form-data ">
<p>
<input type="file" name="soubor" accept="text/plain" /></p>
<p>
<input type="image" src="send.jpg" alt="Odeslat" /></p>
</form>
```

form/  
příklad3

## Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

Elementy `button` a `textarea`

Tyto dva elementy se velmi podobají některým typům ovládacího prvků input. Oproti nim však nabízejí větší možnosti nastavení. Jak již bylo řečeno, element `button` představuje pokročilé tlačítko a element `textarea` rozsáhlejší textové pole. Značka `button` je dostupná pouze v XHTML-MP 1.2.

Atributy `button`:

- `accesskey`, `tabindex`
- `type` udává funkci tlačítka, hodnoty: `button`, `submit`, `reset`
- `value` představuje základní hodnotu tlačítka, nezobrazuje se jako popis
- `disabled` znamená vyřazení tlačítka
- `name` udává jméno tlačítka

## Příklad

Uvnitř `button` může být libovolný obsah (např. obrázky i text najednou). Jsou zde zakázány pouze značky `form` a `a`.

```
<p>
<button type="submit" value="odeslat" name="tlacitko">
  <p><strong>Libovolný obsah.</strong></p>
</button>
</p>
```



form/  
příklad4

## Ukázky

**Formuláře**

Element button

Internet Explorer Mobile  
(Windows Mobile 6)

**Formuláře**

Element button

Opera 9.5 beta  
(Windows Mobile - Touchscreen)

Atributy `textarea`:

- `name` je jméno textového pole
- `rows` udává maximální počet viditelných řádků
- `cols` udává šířku pole, šířka se odvozuje od velikosti znaků (např. 5×šířka znaků)
- `accesskey`, `tabindex`

## Příklad

Ukázka elementu `textarea` s viditelnými 6ti řádky (nultý + 5), šířkou (20×šířka znaků) a jménem `text`. Po odeslání bude obsah umístěn za jméno elementu.

```
<textarea rows="4" cols="20" name="text">
  Lorem ipsum dolor sit amet ...
</textarea>
```



form/  
příklad5

## Ukázky

**Formuláře**

Element textarea

Internet Explorer Mobile  
(Windows Mobile 6)

**Formuláře**

Element textarea

Opera 9.5 beta  
(Windows Mobile - Touchscreen)

**Elementy `select`, `option` a `optgroup`**

Tyto tři značky spoluutvářejí tzv. vysouvací pole. `select` ohraničuje jednotlivé prvky pole `option`, které element `optgroup` sdružuje do skupin.

**Atributy `select`:**

- `name` je opět jméno pole
- `size` udává počet položek, které budou viditelné bez skrolování
- `multiple` určuje výběr buď jedné, nebo více položek najednou (implicitně jedné)
- `tabindex` znamená procházení klávesnicí

**Atributy `option`:**

- `selected` říká, jaká položka má být při načtení vybrána
- `value` specifikuje hodnotu, která se bude posílat s hodnotou `selected`. Pokud jí neuvedete, odešle se textový obsah položky.

**Atributy `optgroup`:**

- `disabled` vyřazuje položku
- `label` určuje popisek skupiny položek

**Příklad**

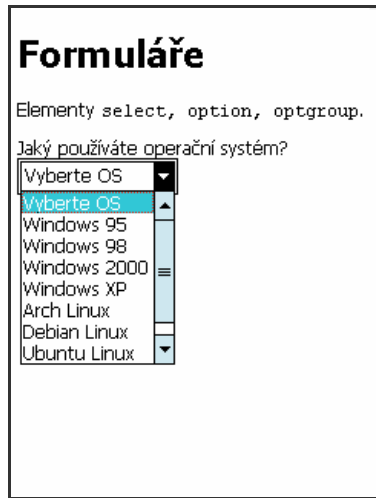
Definice elementů `select`, `option` a `optgroup`. Odesílaná hodnota bude mít tvar: OS=hodnota vybrané položky.



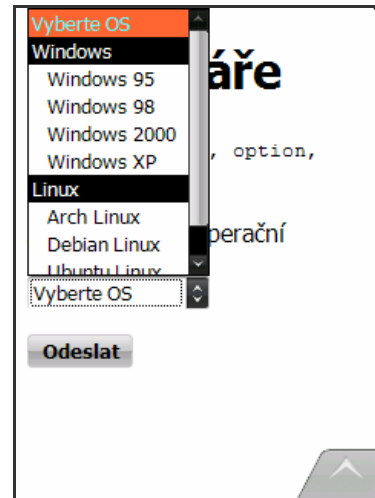
```
<p> Jaký používáte operační systém? <br />
<select name="OS">
<option selected="selected" value="0">Vyberte OS</option>
  <optgroup label="Windows">
    <option value="Win95">Windows 95</option>
    <option value="Win98">Windows 98</option>
    <option value="Win2000">Windows 2000</option>
    <option value="WinXP">Windows XP</option>
  </optgroup>
  <optgroup label="Linux">
    <option value="Arch">Arch Linux</option>
    <option value="Debian">Debian Linux</option>
    <option value="Ubuntu">Ubuntu Linux</option>
    <option value="Mandriva">Mandriva Linux</option>
  </optgroup>
```



## Ukázky



**Internet Explorer Mobile  
(Windows Mobile 6)**



**Opera 9.5 beta  
(Windows Mobile - Touchscreen)**

### Elementy `label`, `fieldset` a `legend`

Tyto značky byly vytvořeny z důvodu lepšího strukturování formulářů. Element `label` představuje popis (doprovodný text) ovládacích prvků, `fieldset` seskupuje jednotlivé prvky (včetně popisu) a `legend` nastavuje nadpis seskupeným prvkům.

#### Atributy `label`:

- `for` slouží k definování ovládacího prvku, ke kterému daný `label` patří. Jméno musí být v atributu `id`.
- `accesskey` představuje přístupovou klávesu

#### Atributy `legend`:

- `accesskey` představuje přístupovou klávesu

form/  
příklad7

## Příklad

Ukázka definice elementů fieldset, label a legend.

```
<fieldset>
  <legend>Osobní údaje</legend>
  <p><label>Příjmení: <input name="prijmeni" /></label></p>
  <p><label>Jméno: <input name="jmeno" /></label></p>
  <p><label>E-mail: <input name="email" /></label></p>
  <p>
    <label><input type="radio" name="pohlavi" value="muz"
      checked="checked" /> Muž</label>
    <label><input type="radio" name="pohlavi" value="zena" />
      Žena</label></p>
</fieldset>
```

## Ukázky

Internet Explorer Mobile  
(Windows Mobile 6)

Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## 2.9.17 Společné atributy

### Atribut xml:lang

Díky kódování Unicode (UTF) odpadají problémy se zobrazováním písma různých světových jazyků. Na stránce se tedy mohou najednou objevit znaky české, japonské, ruské atd. Takovýto obsah je potřeba identifikovat. Jazyky se mohou lišit ve směru zápisu znaků, významu a výslovnosti, stránka může být případně klientovi předčítána.

K tomuto účelu slouží atribut `xml:lang`, jeho hodnotou je kód jazyka, ve kterém je obsah elementu (i všech atributů) napsán. Atribut je dědičný, tj. platí na

všechny své potomky do dalšího výskytu téhož atributu, který ho přepíše. Může být použit u kteréhokoli elementu.

### Kódy zemí lze popsat několika způsoby:

- dvoupísmenným kódem (ISO 639) – např. en, cs, ru.
- dvoupísmenný kód následován subkódem země (ISO 3166) - např. en-us, en-gb, cs-cz.

### Příklad

Deklarace různých jazyků v textu.

```
<h1 xml:lang="cs">Atribut <code>xml:lang</code>.</h1>
<p xml:lang="cs">Ahoj, jak se máš?</p>
<p xml:lang="en">Hello, how are you?</p>
<p xml:lang="de">Hallo, wie geht es dir? </p>
<p xml:lang="ru">Пока, как здоровится ?</p>
```



### Atribut xml:space

Bílé znaky se v kódu používají za účelem vizuální strukturalizace. Jejich zpracování je plně v režii klientských aplikací. Může se však objevit požadavek na přesné zachování bílých znaků uvnitř elementů. Pro vyznačení důležitosti takového formátování používáme atribut `xml:space`. Ten může nabývat dvou hodnot: `preserve` (zachování bílých znaků v nezměněné podobě) a `default` (aplikace používá pro zpracování své vlastní nastavení).

### 2.9.18 XHTML MP vs WML

Předchůdcem jazyka XHTML MP byl jazyk WML, ten po dlouhá léta vládl mobilním zařízením. V čem je nová koncepce lepší, v čem vyniká a kde naopak ztrácí, se dozvíte z následujícího srovnání.

Většina chybějících vlastností jazyka WML jde v XHTML MP realizovat jiným způsobem. WAM 2.0 je zpětně kompatibilní, lze zobrazovat stránky napsané ve WML. Pokud budete některou chybějící vlastnost přímo vyžadovat, je tady možnost zvolit jako programovací jazyk staršího předchůdce.

### Stavba stránky

Ve WML se dokument (deck) skládá z několika podčástí (card), takovou strukturu již XHTML nemá. Výhodou bylo hromadné načtení podstrán. WAP prohlížeč zobrazí

vždy jednu podstránku a na ostatní přechází pomocí odkazů. Minimalizovaly se dotazy mířené na server a zvýšila se rychlost zobrazování obsahu.

V XHTML MP lze využít multipart messages, tj. více dokumentů se odešle najednou klientovi. Nevýhoda je málo podporovaných zařízení. Další možností je použít jeden dlouhý dokument a odkazovat se pomocí kotev (viz odkazy).

### **Časovač**

HXTML MP nepodporuje element `timer`, ten se ve WML používá pro automatický přechod na další podstránku.

Shodnou funkci lze zajistit pomocí hodnoty `refresh` v META elementu.

### **Proměnné**

Ve WML je možné vytvářet proměnné a přiřazovat jim hodnoty. Ty následně platí pro celý dokument i všechny podstránky. XHTML MP toto nepodporuje. Jedinou možností je řešit podobné záležitosti na serveru.

### **Funkční tlačítka**

WML jazyk umožňuje nastavit levou a pravou funkční klávesu (softkey) za pomoci elementu `<do>`.

XHTML MP naopak využívá atribut `accesskey`, kterým umožní rychlý přístup přes jakoukoli klávesu zařízení.

### **Podtržení**

Značka `<u>` slouží ve WML k podtržení písma. Nic takového v XHTML MP nenajdete.

Je zde však více než výhodná možnost využít kaskádové styly, stvořené právě pro estetické úpravy webu.

### **Formuláře**

V XHTML MP neexistuje atribut `format`, ten slouží pro nastavení `input` jazyka WML (omezoval počet a druh znaků).

Náhradou je např. `maxlength`, nebo vlastnosti WCSS.

XHTML MP nepodporuje odesílání formulářů za pomoci odkazů.

### III. Wireless profile Cascading Style Sheet

Kaskádové styly jsou mechanismem, který slouží pro oddělení struktury od prezentace dokumentů. Popisují vzhled stránek, aniž by jakkoli zasahovaly do jejich sémantického obsahu. Bezdrátový profil (Wireless CSS nebo Wireless Profile CSS) je implementován v protokolu WAP 2.0. Spolu s jazykem XHTML-MP tvoří dvojici silných nástrojů pro vytváření stránek na displej mobilních telefonů.

#### 3.1 WCSS verze

WCSS je z hlediska syntaxe podmnožinou jazyka CSS 2.0 (známého z WWW), a navíc obsahuje jistá rozšíření, jedná se o specifické vlastnosti mobilních zařízení. Naopak části jazyka, nepotřebné pro mobilní aplikace, byly odstraněny.

Specifikaci WCSS vytvořila organizace Wap fórum, další specifikací podobného jazyka je CSS Mobile Profile od společnosti W3C. Jedná se o dva rozdílné jazyky, které se však v mnohém podobají.

Veškeré příklady z této kapitoly lze nalézt v sekci „Příklady/WCSS“ na příloženém CD, nebo na internetové adrese <http://www.webmobile.comuv.com/>.

#### 3.2 Výhody kaskádových stylů

- oddělujeme prezentační část od části strukturální, na dokumentu může pracovat vícero vývojových skupin
- styly lze umístit do externího souboru a následně využít pro více stránek najednou. Pokud soubor upravíme, změny se projeví ve všech dokumentech bez jakékoli manipulace s jejich obsahem.
- externí soubor se styly načítáme pouze jednou, uloží se do paměti cache a při každé další návštěvě se automaticky použije. Minimalizují se tak dotazy na server, velikost samotných dokumentů bude minimální.
- je možné vytvářet soubory stylů pro různé výstupní zařízení i pro různé obrazovky mobilních zařízení
- kaskádové styl přináší mnohem větší prezentační možnosti dokumentů srovnatelné s vývojem ve WWW.

### 3.3 Nevýhody kaskádových stylů

- největší nevýhodou je rozdílná podpora stylů v různých mobilních prohlížečích, tento problém existuje dlouhá léta i v technologii WWW
- pokud máte v prohlížeči zakázáno ukládání do cache paměti, soubor stylů se bude při každé návštěvě opakovaně stahovat. To zvyšuje dobu a velikost přenášených dat.

### 3.4 Syntaxe jazyka

Definice stylu sestává z vlastnosti a její hodnoty, má tvar `vlastnost: hodnota`. Jednotlivé definice se mohou sdružovat do seznamu definice, od sebe jsou odděleny znakem „;“ (středník).

#### 3.4.1 MIME typ

Jako každý obsah i kaskádové styly mají svůj specifický typ, dle kterého se klient dozví, s čím má co dočinění, a podle toho data zpracuje. WCSS mají MIME jako klasické kaskádové styly tedy „text/css“.

#### 3.4.2 Umístění v dokumentu

##### Přímé styly

jsou tvořeny pouze **seznamem definic**, zapisují se do atributu `style` a přiřazují vlastnosti konkrétním prvkům uvnitř dokumentu.

Mají tvar: `style="seznam definic"`

##### Příklad

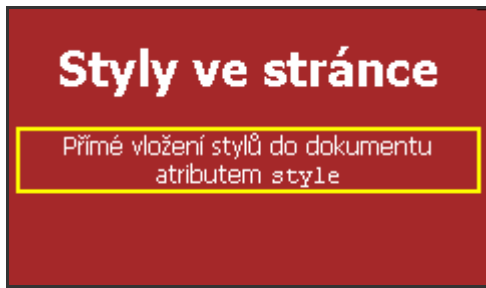
Definice přímých stylů uvnitř dokumentu.

```
<body style="background-color:brown; color:white; font-family:
Arial, sans-serif; text-align:center">
<h1>Styly ve stránce</h1>
<p style="border: 2px solid yellow">
Přímé vložení stylů ... </p>
</body>
```



pripojeni/  
priklad1

## Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Tabulky stylů

představují posloupnost tzv. pravidel a @pravidel. Pravidlo se skládá ze selektoru za nímž následuje seznam definic stylu. Selektor označuje prvek nebo skupinu prvků, pro které platí uvedené definice.

Pravidla mají tvar: `selector {seznam_definic}`

Kromě obyčejných pravidel mohou tabulky stylů obsahovat i tzv. @pravidla. Jde o speciální příkazy, které rozšiřují základní funkce WCSS. Jsou jimi `@import` a `@media`.

Výslednou tabulku lze umístit do **externího souboru**, na který následně v dokumentu odkážeme. Odkazujeme pomocí značky `link` umístěné v hlavičce stránky.

## Příklad

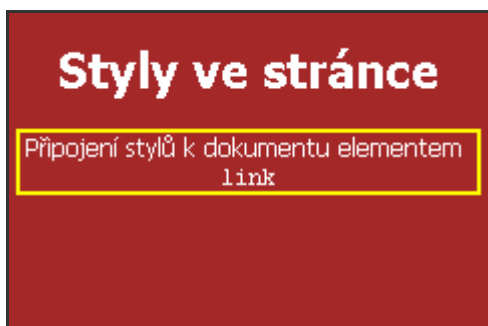
V `link` je uvedena adresa, MIME typ a druh dat.

```
<head>
...
<link href="style.css" rel="stylesheet" type="text/css"/>
...
</head>
```



pripojeni/  
priklad2

## Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

Celou tabulku můžeme také umístit přímo do stránky, na kterou se má vztahovat. V tom případě ji ale musíme uzavřít do značky `<style>` a umístit taktéž do hlavičky dokumentu.

## Příklad

V XHTML musí být veškerý cizí obsah umístěn v sekci CDATA. Element `style` udává MIME typ dat, které obsahuje.



připojení/  
příklad3

```
<head>
...
<style type="text/css">
  /* *]
  body {
    background-color:brown;
    color:white;
    font-family: Arial, sans-serif; text-align:center}
  p {border: 2px solid yellow}
  /* ]]&gt; */
&lt;/style&gt;
...
&lt;/head&gt;</pre>
</div>
<div data-bbox="162 694 233 711" data-label="Section-Header">Ukázky</div>
<div data-bbox="179 718 485 860" data-label="Image">
<img alt="Screenshot of Internet Explorer Mobile (Windows Mobile 6) showing a red background with white text 'Styly ve stránce'. A yellow box highlights the text 'Připojení stylů k dokumentu elementem style'."/>
</div>
<div data-bbox="238 860 427 890" data-label="Caption">
<p>Internet Explorer Mobile<br/>(Windows Mobile 6)</p>
</div>
<div data-bbox="532 725 834 860" data-label="Image">
<img alt="Screenshot of Opera 9.5 beta (Windows Mobile - Touchscreen) showing a red background with white text 'Styly ve stránce'. A yellow box highlights the text 'Připojení stylů k dokumentu elementem style'."/>
</div>
<div data-bbox="563 860 810 890" data-label="Caption">
<p>Opera 9.5 beta<br/>(Windows Mobile - Touchscreen)</p>
</div>
<div data-bbox="826 937 859 955" data-label="Page-Footer">89</div>
```



### 3.4.3 Komentáře

V CSS se komentáře umísťují mezi znaky `/* komentář */`. Ten platí v rámci řádek i bloků textu.

### 3.4.4 Selektory

Selektor je symbolický popis prvku nebo skupiny prvků, na které se vztahuje přiřazená definice stylu. Mohou nabývat několika tvarů:



**Obecný selektor** zastupuje všechny prvky dokumentu. Vyjadřuje ho znak `*`.

Např.: `*{font-size:large}`

**Selektor typu** vyhovuje všem elementům daného typu. Zapisuje se názvem příslušné značky.

Např. `body {background-color:white; ...}`

**Selektor třídy** pro svou funkci využívá atributu `class` jazyka XHTML MP. Jedná se o atribut, který zařazuje vybrané elementy do konkrétních tříd. Jednotlivým třídám lze následně přiřadit definice stylů. K tomuto účelu slouží speciální znak `.` (tečka), uvádíme ho před názvem obsaženém v atributu `class`.

Např.: `.styl{text-align:center}`

```
<h1 class="styl1">Styly ve stránce</h1>
```

Definice vycentruje obsah všech značek, které mají definován atribut `class="styl1"`. Pokud bude určen u vícero rozdílných prvků, lze dodatečně upřesnit pouze některou konkrétní skupinu. Provedeme to vepsáním elementu před deklaraci třídy.

Např.: `h1.styl1{background-color:black; color:white;}`

Pravidlo platí na všechny elementy `h1` s atributem `class="styl1"`.

**Selektor ID** neboli identifikátor je zvláštním druhem selektoru. Jazyk XHTML-MP využívá pro jednoznačné určení prvků stránky atributy `id`. Zápis pravidla je podobný jako u definice tříd, používá však znaku `#`. Před ten můžeme opět umístit název prvku, u kterého je identifikátor uveden. Tento údaj však slouží pouze pro lepší orientaci v kódu, jiné uplatnění nemá. **Identifikátor musí být**

v rámci dokumentu jedinečný, tj. nesmí existovat dva se shodným názvem a jeden prvek může mít maximálně jeden atribut `id`.

Např.: `p#pata {font-size:small}`

```
<p id="pata">Leo justo fermentum...</p>
```

### 3.4.5 Operátory

Selektory je možné kombinovat a vytvářet tak selektory složitější. Používá se k tomu operátorů „mezera“ a „čárka“.

**Operátor mezera** slouží pro vytvoření následovnického vztahu. Pokud element B je následovníkem (někde uvnitř) elementu A. Můžeme napsat: A B.

#### Příklad

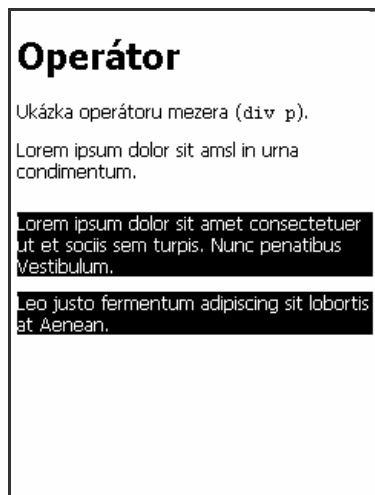
Následující pravidlo platí na všechny elementy `p` umístěné někde uvnitř `div`.

```
div p {font-size:small; color:white; background-color:black}
<p>Lorem ipsum ...</p>
<div>
  <p>Lorem ipsum dolor ...</p>
  <p>Leo justo ...</p>
</div>
```

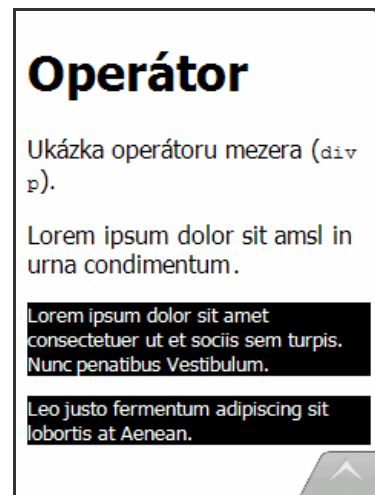


operator/  
příklad1

#### Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

**Operátor čárka** značí konjunkci selektorů. Můžeme definovat stejné vlastnosti více elementům najednou. Zapisuje se: A,B,C

**Příklad**

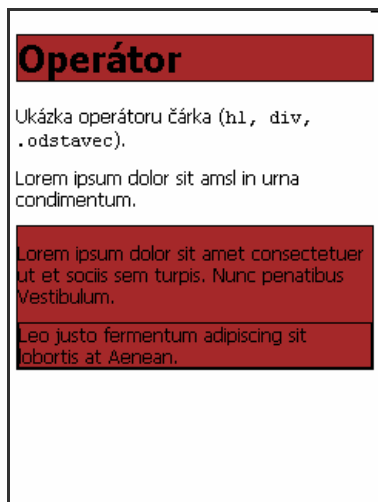
Pravidlo platí zároveň pro elementy h1, div a třídu odstavec.

```
h1, div, p.odstavec {
border: 1px solid black;
background-color:brown}

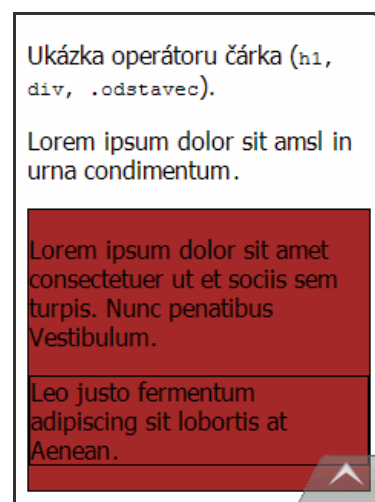
<h1>Operátory</h1>
<p>Ukázka operátoru mezera.</p>
<p>Lorem ipsum dolor sit ...</p>
<div>
  <p>Lorem ipsum dolor sit ...</p>
  <p class="odstavec">Leo justo ...</p>
</div>
```



operator/  
priklad2

**Ukázky**

Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

**3.4.6 Kaskáda**

Kaskáda je způsob zpracování totožných definic stylů, platných pro stejné elementy stránky. V dokumentu se při přiřazování hodnot vlastnostem schází více tabulek najednou. Výchozí tabulku stylů má prohlížeč (podle ní formátuje nedefinované vlastnosti), uživatel může mít své vlastní styly, navíc lze připojit k dokumentu více vlastních tabulek. Prvky tedy mohou mít jednu vlastnost určenou více způsoby. Jakou hodnotu vlastnosti ve výsledku použít, vybírá právě mechanismus kaskády.

Při zpracovávání stránky přiřazuje klient elementům hodnoty všech vlastností, které lze na daném médium použít. Vytváří při tom hierarchický strom dokumentu. Každá vlastnost má určenu svou výchozí hodnotu, pokud se za pomoci mechanismu

kaskády nepřihodí hodnota jiná (a vlastnost není dědičná – viz Dědičnost), použije se právě výchozí.

### Pravidla kaskády

**Váha selektorů** – je-li definováno více pravidel s hodnotami stejných vlastností, použij to, které má největší váhu selektoru. Největší váhu mají pravidla !important, dále tabulky autora, tabulky uživatele a tabulka klienta. Výraz !important nalezne uplatnění v uživatelských stylech, kde umožňuje měnit styly autora. Uvádí se až za hodnotou požadované vlastnosti.

**Specifičnost selektorů** – pravidla se stejnou váhou dále hodnotíme dle specifičnosti selektorů. Nejvíce specifické pravidlo zvítězí. Obecné selektory (\*, selektor typu) jsou nejméně specifické, nejkonkrétnější jsou naopak nejvíce specifické (#identifikátor).

### Příklad

Specifičnost selectorů. Platí pořekadlo: Bližší košile než-li kabát!

```
* {
font-size:normal;
background-color:white;
color:black}

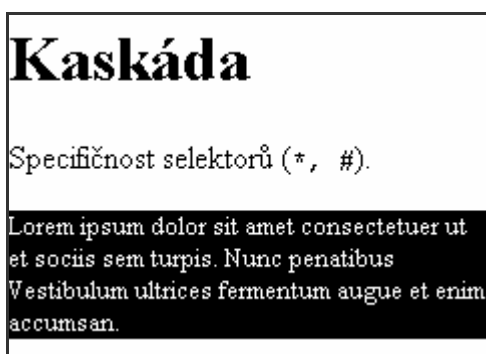
p#odstavec {
font-size:small;
background-color:black;
color:white}

<p id="odstavec">Lorem ipsum dolor sit ... </p>
```

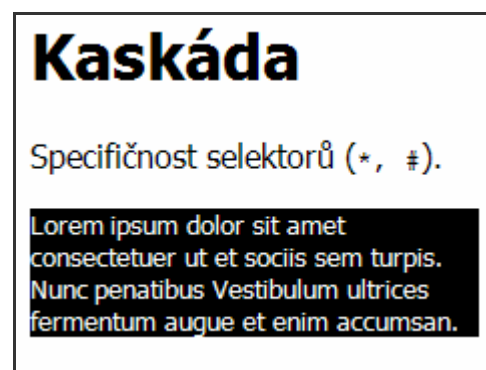


kaskada/  
příklad1

### Ukázky



Skyfire 0.9.333  
(Windows Mobile)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

**Pořadí selektorů** – jsou li dva selektory stejně specifické, použij hodnotu toho, který je uveden v kódu později

### Příklad

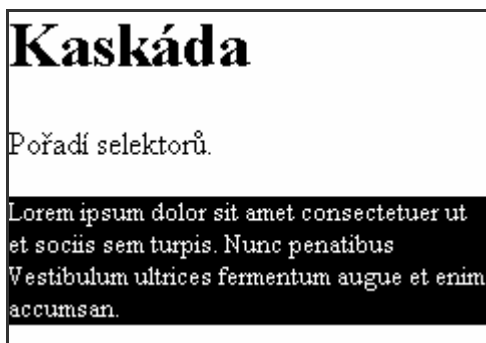
První pravidlo určuje bílé pozadí a modré písmo. Druhé mění barvu pozadí a přidává velikost písma. Třetí změní barvu písma i pozadí, velikost není uvedena, zůstane tedy zachována z předchozího pravidla.



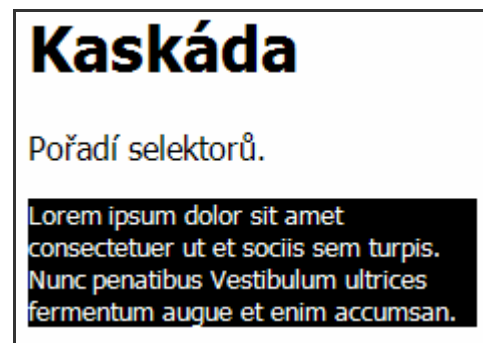
kaskada/  
priklad2

```
p.odstavec {background-color:white; color:blue}
p.odstavec {background-color:yellow; font-size:small}
p.odstavec {background-color:black; color:white}
```

### Ukázky



Skyfire 0.9.333  
(Windows Mobile)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

Přiřazované hodnoty procházejí několika fázemi. Nejprve se jedná o definovanou hodnotu. Ta může být tvořena absolutními jednotkami, relativními jednotkami a nějakou ze zkratkových hodnot. Absolutní většinou žádnou úpravu nepotřebují, relativní a zkratkové hodnoty (např. procenta, zástupná hodnota) však vyžadují přepočítání. Další fází je tedy vypočítaná hodnota, které se již přiřadí konkrétní vlastnosti.

### 3.4.7 Dědičnost

Některé vlastnosti mohou zdědit hodnoty od svých rodičů. U každé vlastnosti je uvedeno, zda dědičná je či není. Pokud dědičná není, není jí přiřazena žádná hodnota, nebo se jedná o kořenový prvek dokumentu, použije se hodnota výchozí.

## Příklad

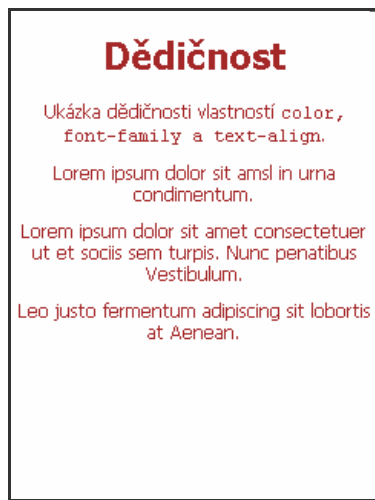
Uvedené vlastnosti zdědí všechny elementy uvnitř `html` a `body`, tedy celý viditelný dokument.



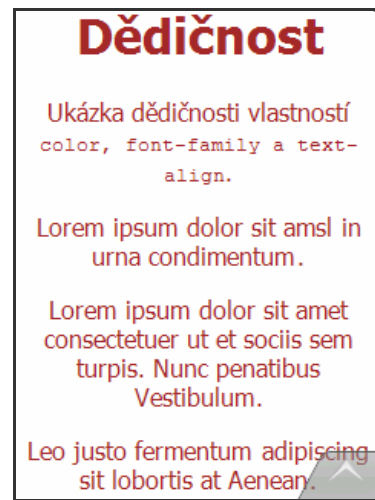
dedicnost/  
priklad1

```
html, body {
color: brown;
font-family: Arial, Helvetica, sans-serif;
text-align:center}
```

## Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

Dědí se většinou hodnoty vypočítané. Každá vlastnost může mít hodnotu `inherit`. Ta vynucuje dědění od svého rodičovského prvku.

## Příklad

Vynucené dědění od rodičovského prvku..

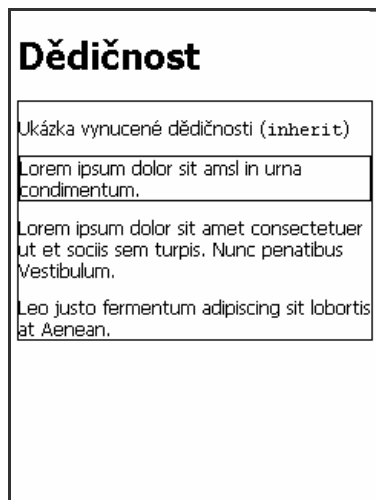
```
div {border: 1px solid black}
p.ramecek {border:inherit}

<div>
...
  <p class="ramecek">Lorem ipsum ...</p>
...
</div>
```

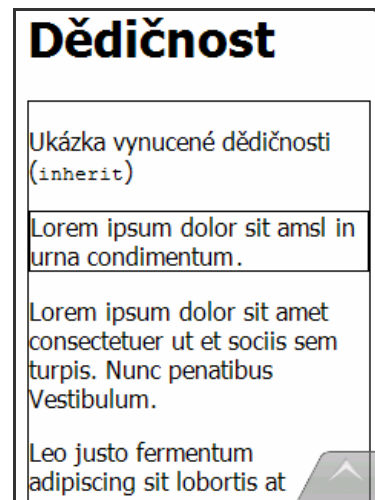


dedicnost/  
priklad2

## Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

### 3.4.8 Datové typy

Vlastnostem můžeme přiřadit několik různých typů hodnot a k nim celou řadu jednotek. U každé vlastnosti je definován seznam povolených hodnot.

#### Klíčová slova

Některé hodnoty mohou zastupovat hodnoty složitější, zbavíme se tak často používaných složitých zápisů, které nás při práci zpomalují. Ty se v procesu přiřazování převedou do své skutečné podoby.

Např.: `rgb(0, 128, 0)` zastupuje slovo `green`

Klíčové hodnoty zapisuje bez uvozovek.

Např.: `display:none` a `color:black`

#### Čísla

V WCSS existují reálná a celá čísla. Celé číslo zapisujeme sekvencí číslic 0-9, před kterými mohou být ve výsledku znaménka „+“ a „-“. Reálná čísla jsou čísla celá nebo desetinná.

Např.: celá čísla `10`, `15`, `-45`, `+2`

reálná čísla `-5.4`, `10`, `+3.564`

**Velikost**

Velikosti se zapisují ve tvaru *číslojednotka*. Jednotky jsou buď relativní nebo absolutní.

**Relativní jednotky:**

Tyto jednotky se vždy vztahují k nějaká další hodnotě. Jsou jimi:

- % - procentní hodnota z nějaké další velikosti
- px – relativní jednotka bodu obrazovky, vztahuje se k použitému výstupnímu zařízení
- em– velikost právě používaného písma v prvku
- ex – velikost písma x, právě používaného písma

**Absolutní jednotky:**

- cm, mm (centimetr, milimetr)
- in – palec (25,4mm)
- pt – typografický bod (0.353 mm)
- pc – typografická jednotka pica 1 pc = 12 pt

**Barvy**

Barvy lze zapisovat pomocí zástupných hodnot, nebo pomocí číselného zastoupení RGB (Red Green Blue).

**Poměr RGB můžeme určit několika způsoby:**

- Pomocí číselných nebo procentních hodnot zápisem `rgb (R, G, B)`.  
Používá se rozsah od 0-255, s tím korespondují hodnoty 0-100%.  
Tvar např. `rgb (100%, 100%, 100%)` je shodný  
s `rgb (255, 255, 255)`.
- Pomocí hexadecimálních hodnot. Zapisují se ve tvaru `#RRGGBB`, nebo zkráceném `#RGB`. Tvoří hexadecimální vyjádření barvy.  
Např. `black = #000000 = #000`

Tabulku zástupných barev naleznete v **Příloze B** na konci publikace.



**URI adresa**

Za pomocí URI (Uniform Resource Identifier) lze popsat jednoznačnou adresu souborů na internetu. Zapisujeme ji do funkce `url()`.

Všechny tyto zápisy jsou možné: `url(adresa)`, `url("adresa")`, `url('adresa')`.

**Zkratky**

Některé vlastnosti kaskádových stylů jsou deklarovány jako tzv. zkratkové. Jejich hodnotou je soupis dílčích hodnot více vlastností, které oddělujeme mezerami.

```
Např.: margin: 10px 15px 20px 25px;
        border: 1px solid black;
```

Kdybychom chtěli dokumentu přiřadit hodnoty okrajů (`margin`) bez tohoto zkráceného zápisu, museli bychom deklarovat 4 dílčí vlastnosti.

**3.4.9 Média**

Jednou z velkých výhod technologie kaskádových stylů je definování různých vzhledů pro různá výstupní zařízení. Výstupním zařízením říkáme média a řadíme je do několika skupin.

**Druhy médií:**

**all** – všechny druhy zařízení

**aural** – zvukový výstup hlasových zařízení

**braille** – braillová dotyková zařízení

**embossed** – plastnický tisk nebo braillové zařízení

**handheld** – obrazovky mobilních přenosných zařízení

**print** – stránky na tiskárně nebo náhledy tisku

**projection** – plošná projekce

**screen** – obrazovky počítačů

**tty** – znakový výstup (dálnopisy aj.)

**tv** – televizní obrazovky

### 3.4.10 Média v tabulkách stylů

Výstupní zařízení rozlišujeme atributem `media` ve značkách `link` a `style`. Hodnotou atributu může být seznam médií oddělených čárkou.

Např.: `media="handheld, screen"`

```
media="all "
```

Na mobilní prohlížeče se vztahují profily `all` a `handheld`. Implicitní hodnotou je `all`.

#### Pravidla `@media` a `@import`

K určení typů médií přímo v tabulkách stylů používáme příkaz `@media`. Označíme tím skupinu pravidel platných pouze na určité výstupní zařízení.

Má tvar: `@media seznam_medií {pravidla}`

#### Příklad

Ukázka definic stylů pro různá výstupní zařízení.

```
<style type="text/css">
/* */
@<b>media handheld</b> {<b>body</b> {font-size:small; color:blue}}
@<b>media screen</b> {<b>body</b> {font-size:large; color:brown}}
@<b>media print</b> {<b>body</b> {font-size:medium; color:black}}
/* ]]&gt; */
&lt;/style&gt;</pre>
</div>
<div data-bbox="880 437 969 489" data-label="Image">
<img alt="Globe icon representing media examples."/>
</div>
<div data-bbox="897 490 962 516" data-label="Caption">
<p>media/<br/>příklad1</p>
</div>
<div data-bbox="161 603 857 646" data-label="Text">
<p>Pravidlem <code>@import</code> vkládáme do stávajících tabulek tabulky jiné. Umístíme ho před všechny pravidla v dané tabulce.</p>
</div>
<div data-bbox="185 661 653 679" data-label="Text">
<p>Zapíše se ve tvaru: <code>@import URL seznam_medií</code></p>
</div>
<div data-bbox="161 702 233 719" data-label="Section-Header">
<h4>Příklad</h4>
</div>
<div data-bbox="185 727 594 744" data-label="Text">
<p>Vložení tabulky stylů pomocí příkazu <code>@import</code>.</p>
</div>
<div data-bbox="161 757 609 890" data-label="Text">
<pre>&lt;style type="text/css"&gt;
/* <![CDATA[*/
@<b>import</b> url(styly.css) handheld, screen;
<b>p#pata</b> {
font-size:small;
background-color:black;
color:white}
/* ]]&gt; */
&lt;/style&gt;</pre>
</div>
<div data-bbox="880 703 969 755" data-label="Image">
<img alt="Globe icon representing media examples."/>
</div>
<div data-bbox="897 756 962 782" data-label="Caption">
<p>media/<br/>příklad2</p>
</div>
<div data-bbox="829 936 862 955" data-label="Page-Footer">99</div>
```

### 3.4.11 Pseudo třídy

Selektory pseudotříd dokáží rozlišovat prvky dokumentu pomocí interakce uživatele. Mohou se definovat samostatně, nebo je lze použít jako doplněk selektorů jiných. Ve WCSS existují pseudotřídy:

#### Odkazy

- `:link` - prvek je v pseudotřídě `link`, pokud ještě nebyla navštívena adresa jeho odkazu
- `:visited` - prvek je v pseudotřídě `visited`, pokud již byla navštívena adresa jeho odkazu

#### Všechny prvky

- `:active` - do pseudotřídě `active` spadá prvek, pokud byl aktivován (kliknutím, klávesou `enter` aj.)
- `:focus` - tuto pseudotřídu má prvek, který právě získal zaměření (např. tabulátorem aj.)

#### Příklad

Všechny nenavštívené odkazy na stránce mají oranžovou barvu, navštívené odkazy mají barvu šedou, pokud některý zaměříte klávesou tabulátor, tak zezelená a pokud se kterýkoliv odstavec aktivuje (kliknete na něj), tak zoranžoví.



pseudotridy/  
priklad1

```
:link {color:orange}
:visited {color:gray}
p:active {color:orange}
a:focus {color:green}
```

#### Ukázky

## Pseudotřídy

Ukázka pseudotříd `:link`, `:visited`,  
`:active`, `:focus`.

[Kotva1](#) [Kotva2](#) [Kotva3](#) [Kotva4](#) [Kotva5](#)

---

## Nějaký text

Lorem ipsum dolor sit amet  
consectetur euismod turpis lacinia  
Morbi Curabitur. Consectetur eu  
hendrerit nunc Nam dui tincidunt

Internet Explorer Mobile  
(Windows Mobile 6)

[Kotva1](#) [Kotva2](#) [Kotva3](#) [Kotva4](#)  
[Kotva5](#)

---

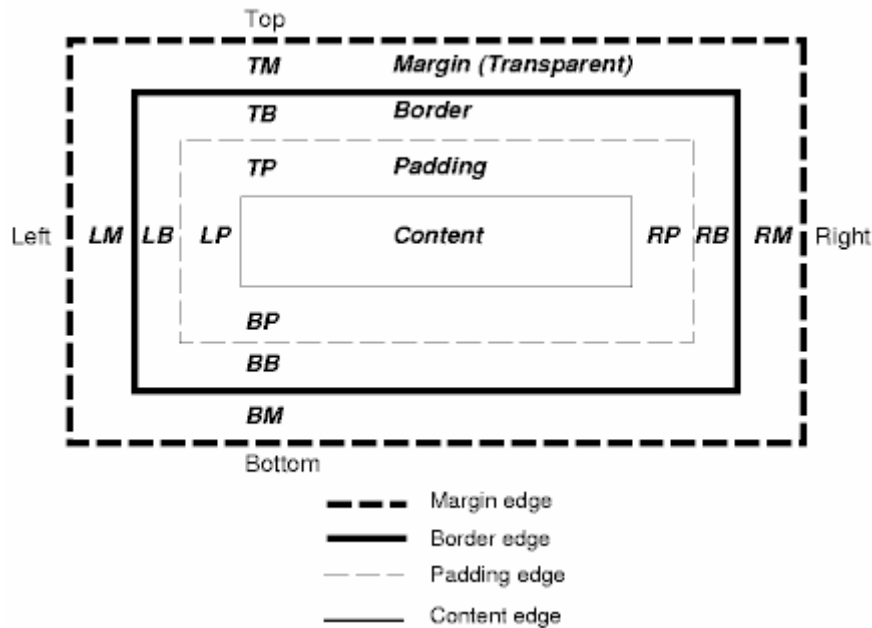
## Nějaký text

Lorem ipsum dolor sit amet  
consectetur euismod turpis  
lacinia Morbi Curabitur.  
Consectetur eu hendrerit nunc

Opera 9.5 beta  
(Windows Mobile - Touchscreen)

### 3.4.12 Box model

Boxový (rámový) model představuje obdélníkové rámy, generované pro všechny elementy dokumentu viz obr.14. Základní částí je oblast obsahu, zde se nachází samotný obsah prvku (obrázek, text aj.). Kolem mohou (ale nemusí) být další oblasti - oblast výplně (Padding), oblast rámečku (Border) a nakonec oblast okraje (Margin).



**Obrázek 14 – Rámy prvků**

**Zdroj:** W3C. *Cascading Style Sheets, level 2. [s.l.] : [s.n.], 1998. Dostupný z WWW: <<http://www.w3.org/TR/1998/REC-CSS2-19980512/box.html>>. Box dimensions, s. 8.1.*

Každou oblast udávají čtyři rozměry – horní (Top), pravý (Right), dolní (Bottom) a levý (Left). Lze je shrnout v některé ze zkratkových vlastností.

Celkové rozměry prvku tvoří součet všech jeho dílčích oblastí:

$$\text{Celková šířka} = LM + LB + LP + \text{width} + RP + RB + RM$$

$$\text{Celková výška} = TM + TB + TP + \text{height} + BP + BB + BM$$

Width a height jsou vnitřní rozměry prvku a určují **omezující oblok** pro jeho obsah. Elementy respektují rozměry svých omezujících bloků.

Generované rámy jsou dvojího typu: **blokové a řádkové**. Blokové rámy se generují pro blokové elementy, řádkové pro řádkové elementy. Neoznačený obsah se umístí do (neviditelných) anonymních bloků. Anonymní blok dědí hodnoty vlastností od svých nadřazených prvků.

## Margin (okraj)

Oblast okraje představuje průhledný odstup elementu od ostatních prvků dokumentu. Jeho rozměry definují vlastnosti `margin-top` (horní okraj), `margin-right` (pravý okraj), `margin-bottom` (dolní okraj), `margin-left` (levý okraj). Hodnotou může být buď konkrétní velikost, procenta, nebo hodnota `auto`. Procentní údaj se vztahuje vždy k šířce/výšce svého omezující bloku. Hodnota `auto` se vypočte v závislosti na použitém pozičním schématu (viz kapitola Rozměry rámu). Všechny čtyři vlastnosti lze shrnout do definice jediné, tou je zkratka `margin`.

Má tvar: `margin: Top Right Bottom Left;`  
`margin: All;`  
`margin: TopBottom RightLeft`  
`margin: Top RightLeft Bottom`

Za normálních podmínek se okraje sousedních prvků spojují a jejich velikost je dána větším z nich. Výchozí hodnotou je 0. Jsou možná i záporná čísla.

## Příklad

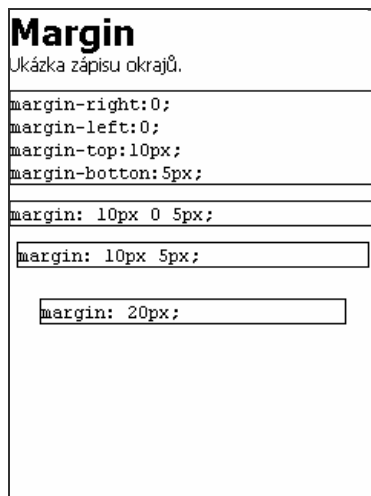
Deklarace okrajů. První a druhý zápis jsou totožné, mezera mezi nimi bude odpovídat 10 px. Mezera mezi druhým a třetím odstavcem bude také 10 px a mezi třetím a čtvrtým 20px. Třetí a čtvrtý prvek jsou odsazeny zleva o 5 a 20 px.

```
* {margin:0}
p.prvni, p.druhy, p.treti, p.ctvrty {border:1px solid black}
p.prvni {
margin-right:0;
margin-left:0;
margin-top:10px;
margin-bottom:5px}
p.druhy {margin: 10px 0 5px}
p.treti {margin: 10px 5px}
p.ctvrty {margin: 20px}
```

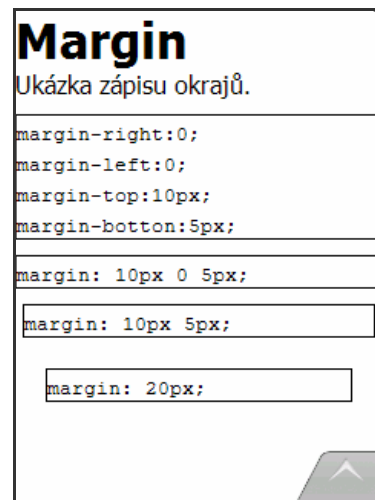


boxmodel/  
příklad1

## Ukázky



**Internet Explorer Mobile  
(Windows Mobile 6)**



**Opera 9.5 beta  
(Windows Mobile - Touchscreen)**

Je výhodné na začátku vynulovat všechny okraje. Můžou být nastaveny v tabulkách prohlížečů, a to v každém prohlížeči různě.

## Border (rámeček)

Rámeček tvoří viditelné orámování elementů. Rozlišujeme vlastnosti pro určení šířky, stylu a barvy rámečků.

### Šířka

Tloušťku rámečku určuje skupina vlastností:

- border-top-width (horní)
- border-right-width (pravá)
- border-bottom-width (dolní)
- border-left-width (levá)

Mohou nabývat hodnot *thin* (tenký rámeček), *medium* (střední), *thick* (silný). Výchozí je *medium*. Přesnou hodnotu lze určit zadáním přesné velikosti.

Všechny velikosti je také možné zadat najednou, a to za pomoci vlastnosti *border-width*. Její zápis je totožný s vlastností *margin*. Šířka rámečku nemůže být záporná.

**Barva**

Barvu rámečku zadáváme vlastnostmi:

- `border-top-color` (horní)
- `border-right-color` (pravá)
- `border-bottom-color` (dolní)
- `border-left-color` (levá)

Hodnotu zadáváme pomocí některého možného zápisu barev.

Opět je možnost zkrácené deklarace všech vlastností zároveň, využívá se vlastnosti `border-color`. Jako výchozí se použije hodnota vlastnosti `color`.

**Styl**

Styl rámečku řídí čtveřice vlastností:

- `border-top-style` (horní)
- `border-right-style` (pravá)
- `border-bottom-style` (dolní)
- `border-left-style` (left)

Hodnotou může být `none` (žádný rámeček), `solid` (plná čára), `hidden` (stejně jako `none`), `dotted` (tečkovaný), `dashed` (čárkovaný), `double` (dvojitý), `groove` (vyryt do pozadí), `ridge` (vystupuje nad pozadí), `inset` (prvek zapuštěn do pozadí), `outset` (prvek vystupuje z pozadí).

**Sdružené vlastnosti**

Následující vlastnostmi definujeme šířku, barvu a styl zároveň:

- `border-top` (horní)
- `border-right` (pravá)
- `border-bottom` (dolní)
- `border-left` (levá)

Hodnoty mohou být v libovolném pořadí. Pokud některou neuvedeme, použije se její výchozí hodnota.

Vlastností `border` určíme šířku, barvu a styl všech čtyřek stran najednou.

### Příklad

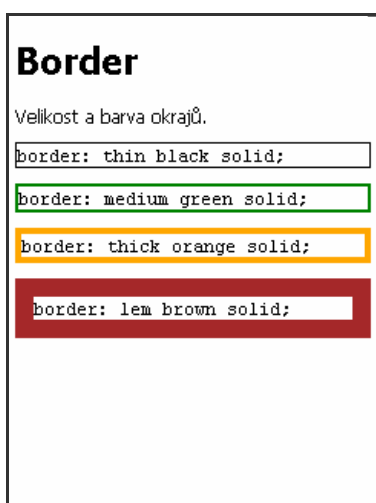
Definice různých velikostí a barev rámečků.

```
p.prvni {border: thin black solid}
p.druhy {border: medium green solid}
p.treti {border: thick orange solid}
p.ctvrty {border: 1em brown solid}
```



boxmodel/  
priklad2

### Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

### Příklad

Nastavení různých stylů rámečku

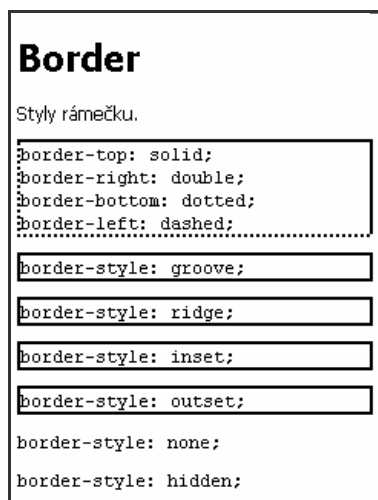
```
p.prvni {
border-top-style: solid;
border-right-style: double;
border-bottom-style: dotted;
border-left-style: dashed}
p.druhy {border-style: groove}
p.treti {border-style: ridge}
p.ctvrty {border-style: inset}
p.paty {border-style: outset}
p.sesty {border-style: none}
p.sedmy {border-style: hidden}
```



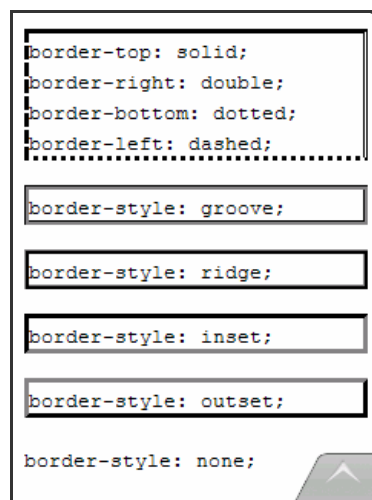
boxmodel/  
priklad3



## Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Padding (výplň)

Padding představuje oblast výplně, umístěnou mezi obsahem prvku a jeho rámečkem. Určíme ho vlastnostmi:

- padding-top (horní)
- padding-right (pravá)
- padding-bottom (doplní)
- padding-left (levé)

Hodnotu můžeme zadat v procentech (vztahují se k šířce omezujícího bloku), nebo uvést konkrétní velikost. Výchozí hodnotou je 0.

Zápis lze zkrátit vlastností padding.

## Příklad

Deklarace oblastí výplně s počátečním vynulováním.

```
* {padding:0}
p.prvni, p.druhy, p.treti {border:1px solid black}
p.prvni {
padding-top:10px;
padding-right:15px;
padding-botton:5px;
padding-left:20px}
p.druhy {padding: 10px}
p.treti {padding-left:20px}
```



boxmodel/  
priklad4

## Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

### 3.4.13 Vizuální formátování

Všem vlastnostem všech prvků jsou již přiřazeny příslušné hodnoty, dalším krokem prohlížečů je jejich zobrazení. Zobrazení se provádí podle postupů určených formátovacím modelem.

#### Display

Vlastnost `display` určuje typ generovaných rámců elementů. Může nabývat několika hodnot:

- `inline` – řádkový prvek, generuje se jeden nebo více řádkových rámců
- `block` – blokový prvek, generuje blokové rámy
- `list-item` – položka seznamu, generuje jeden hlavní, případně i doplňkový rám
- `none` – negeneruje žádný rám, element se chová jako by v dokumentu nebyl

Výchozí hodnotou je `inline`. V tabulkách klientů jsou vlastnosti `display` předdefinované.



## Příklad

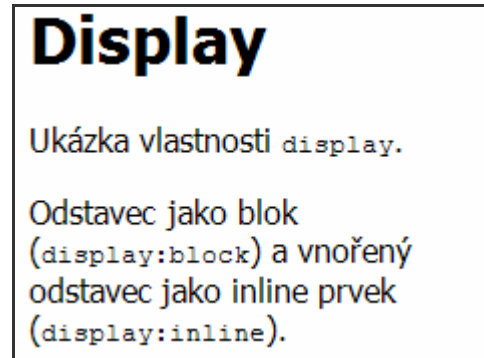
Různé formátovací definice. Hodnota `list-item` je předvedena v kapitole Seznamy.

```
<p style="display:block">Odstavec jako blok ...
<p style="display:inline">vnořený odstavec ...</p>.</p>
<p style="display:none">Neviditelný rám ...</p>
```

## Ukázky



Skyfire 0.9.333  
(Windows Mobile)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Poziční schéma

Umístění generovaných rámců v dokumentu závisí na použitém pozičním schématu.

Ve WCSS existují dva druhy pozičního schéma:

- normální tok dokumentu
- plovoucí prvky (float)

V **normálním toku** jdou po sobě prvky tak, jak jsou definovány v kódu dokumentu. Okraje horní a dolní okraje (margin) sousedních prvků se slučují.

### Plovoucí prvky (float)

Vlastnost float zajišťuje obtékání elementů. Hodnotou může být `left` (obtéká vlevo), `right` (obtéká vpravo) a `none` (neobtéká). `None` je také výchozí hodnotou. Plovoucí prvky musí mít zadánu šířku!

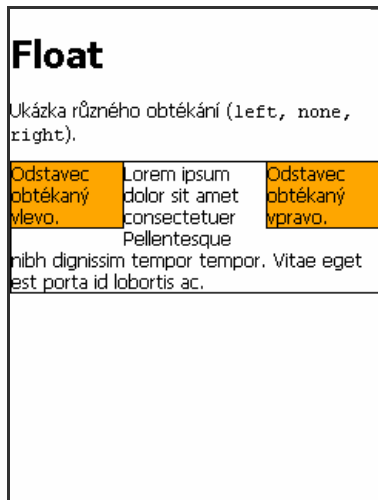
## Příklad

Obtékání odstavců dokumentu s nastavenou šířkou 20% stránky.

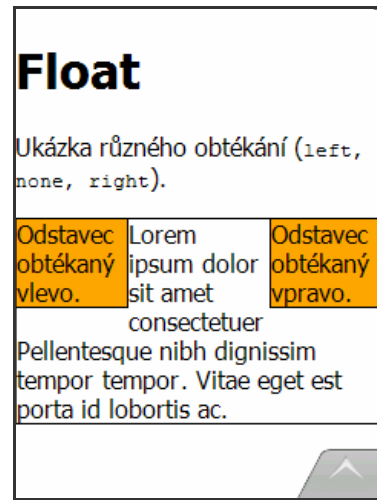
```
p.prvni {float:left; width:30%}
p.druhy {float:right; width:30%}
p.treti {float:none}
```



## Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Zákaz obtékání

Zatímco `float` určuje jak bude prvek obtékán, vlastnost `clear` říká, zda může být vůbec obtékán. Hodnoty: `none` (bez omezení), `left` (zákaz obtékání vlevo), `right` (zákaz obtékání vpravo), `both` (zákaz obtékání)

## Příklad

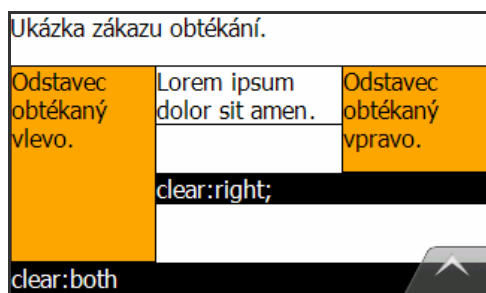
Zákaz obtékání vlevo je analogií `clear:right`, proto ho v ukázce neuvádím.

```
p.ctvrty {clear:right}
p.paty {clear:both}
```



formatovani/  
příklad3

## Ukázky



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## 3.4.14 Rozměry elementů

Vlastnosti `width` (šířka) a `height` (výška) definují rozměry obsahu prvku dokumentu. Jejich hodnotu lze uvést v procentech (relativně k šířce/výšce omezujícího bloku), `auto`, nebo konkrétní velikostí. Výchozí hodnotou je `auto`, tzn. vypočítaný údaj z rovnice (viz kapitola Rozměry rámců).

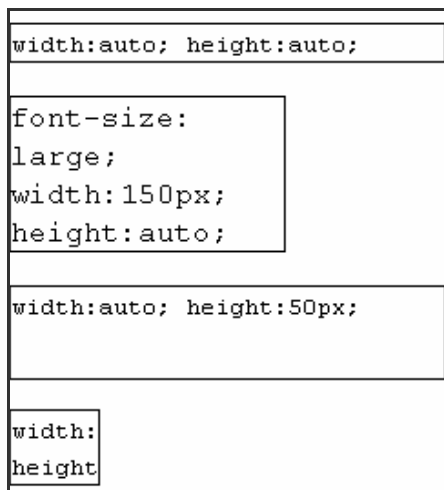


## Příklad

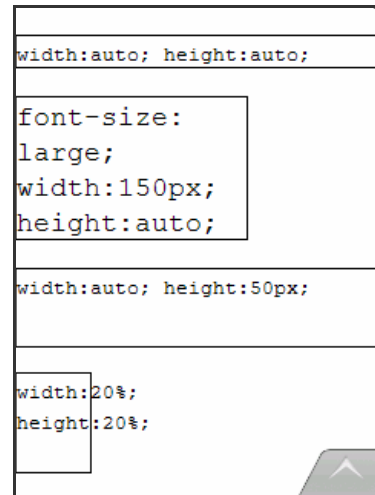
Určení různých rozměrů obsahu.

```
html, body {width:100%; height:100%}
p.prvni {width:auto; height:auto}
p.druhy {width:150px; height:auto; font-size: large}
p.treti {width:auto; height:50px}
p.ctvrty {width:20%; height:20%}
```

## Ukázky



Skyfire 0.9.333  
(Windows Mobile)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

### 3.4.15 Rozměry rámců

Další rozdělení prvků je na **nahrazované** a **nenahrazované**. Nahrazované prvky jsou ty, jejichž obsah nevychází přímo z dokumentu (např. obrázky). Nenahrazované jsou všechny ostatní.

#### Řádkové nenahrazované prvky

Nepoužívají se zde vlastnosti `width` a `height`. Vypočítané hodnoty `auto` okrajů `margin` se rovnají nule. Výšku řádky tvoří horní hrana nejvyššího prvku a spodní hrana nejnižšího prvku.

#### Řádkové nahrazované prvky

Vypočítané hodnoty `auto` okrajů `margin` se rovnají nule. Pokud je zadán jeden z rozměrů `width` a `height`, druhý se vypočte ze skutečných rozměrů prvku (dle poměru stran).

### Blokové nenahrazované prvky

Vypočítané hodnoty `auto` okrajů `margin` se rovnají nule. Při výpočtu šířky musí být zachována rovnost:

`Šířka omezujícího bloku = margin-left + border-left + padding-left + width + padding-right + border-right + margin-right`

Pokud jsou všechny hodnoty definovány, `margin-right` se ignoruje. Pokud je `width auto`, ostatní hodnoty `auto` jsou nulové. Pokud je `margin-left` a `margin-right auto` a je určená šířka, prvek bude vystředěn.

Je-li `height auto`, výška odpovídá výšce obsahu prvku.

### Blokové a plovoucí nahrazované prvky

Jsou shodné s řádkovými nahrazovanými prvky.

### Plovoucí nenahrazované prvky

Vypočítané hodnoty `auto` okrajů `margin` se rovnají nule. Výška udává součet výšek obsahu.

## 3.4.16 Barvy a pozadí dokumentu

Následujícími vlastnostmi přiřazujeme barvy všem prvkům dokumentu.

### Barva popředí

Barva popředí elementů závisí na vlastnosti `color`. Většinou se jedná o barvu písma. Vlastnost je dědičná. Hodnotou je některá definice barvy.

Výchozí hodnotu určuje prohlížeč.

### Příklad

Definice barev pro celé odstavce a úseky textu.

```
span {color:brown}
p.prvni {color:green}
p.druhy {color:orange}
p.treti {width:auto; height:50px}
```

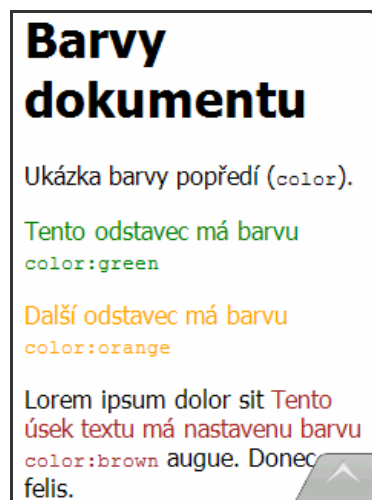


barvy/  
příklad1

## Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Barva pozadí

Podbarvení elementů definujeme vlastností background-color. Hodnotou může být některý zápis barvy, nebo transparent (průhledné pozadí). Transparent je i výchozí hodnotou, u kořenového elementu ji definuje prohlížeč. Pozadí zasahuje do oblasti obsahu, výplně a rámečku.

## Příklad

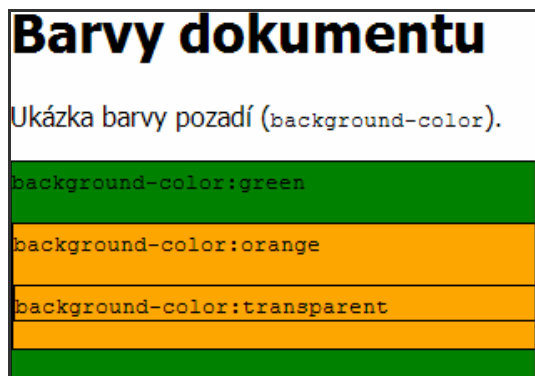
Nastavení barev v pozadí prvků a jejich vzájemné překrývání.

```
body {background-color:white}
p.prvni {background-color:green}
p.druhy {background-color:orange}
p.treti {background-color:transparent}
```



barvy/  
priklad2

## Ukázky



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Obrázek na pozadí

Na pozadí prvku lze nechat vykreslit obrázek. Provádí se to vlastností `background-image`. Pokud obrázek nepokrývá celou plochu prvku, zbytek se vyplní barvou z vlastnosti `background-color`. Pokud obrázek přesahuje, je oříznut. Do hodnoty uvádíme URL adresu souboru, výchozí hodnota je `none`.

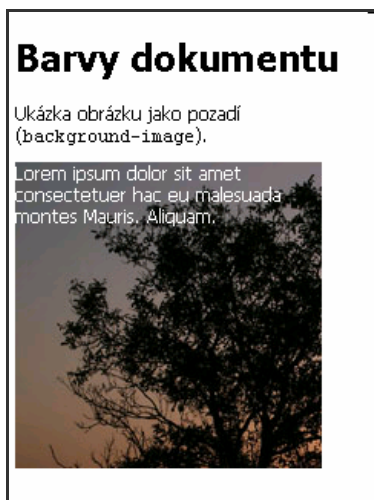
### Příklad

Odstavec má oba rozměry nastaveny na 200px. Rozměry obrázku jsou 420×315px, je tedy oříznut. Pokud by se z nějakého důvodu nenačetl, použila by se barva pozadí (`black`).



```
p.prvni {
width: 200px;
height: 200px;
background-image: url(krajina.jpg);
background-color:black;
color:white}
```

### Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

### Opakování obrázku

Vlastnost `background-repeat` má za následek opakování obrázku na celé ploše pozadí prvku. Hodnoty mohou být `repeat` (opakování v obou směrech), `repeat-x` (opakování po ose x), `repeat-y` (opakování po ose y), `no-repeat` (zákaz opakování). Výchozí hodnotou je `repeat`.



## Příklad

Definice opakování obrázku v obou směrech.

```

p.prvni {
...
background-image: url(krajina1.jpg);
background-repeat: repeat}

```

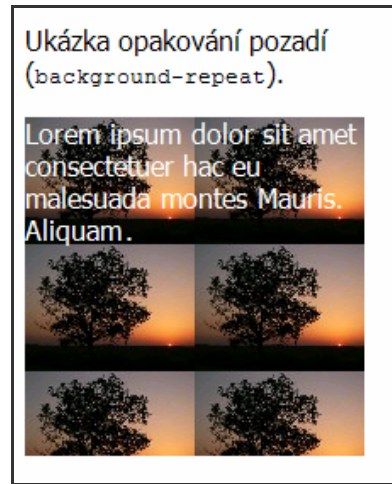


barvy/  
příklad4

## Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Ukotvení obrázku

Vlastnost `background-attachment` určuje, zda se má obrázek při posouvání obsahu posouvat také, nebo zůstat na místě. Výchozí hodnotou je `scroll`, obrázek se posouvá společně s obsahem. Pro zajištění opačného efektu používáme klíčové slovo `fixed`.

## Příklad

Uchycení obrázku v pozadí stránky.

```

body {
color:white;
background-color:black;
background-image: url(krajina.jpg);
background-repeat: no-repeat;
background-attachment:fixed}

```



barvy/  
příklad5

## Ukázky



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Pozice obrázku

Obrázku lze zadat přesnou pozici v pozadí elementu, na kterou má být vykreslen, provádí to vlastnost `background-position`. Pozice se vztahuje k levému hornímu rohu padding oblasti. Obrázek následně pozicujeme od jeho levé horní hrany. Hodnotou je dvojice čísel, udávající souřadnice *x* a *y*.

Lze je zadat několika způsoby:

- procenta – zápis 0% 0% (výchozí hodnota) bude znamenat umístění levého rohu obrázku do levého rohu oblasti padding.
- velikost – zápis 10px 10px posune obrázek o 10px dolů a doprava.
- klíčová slova - `top`, `center`, `bottom`, `left`, `right`.

## Příklad

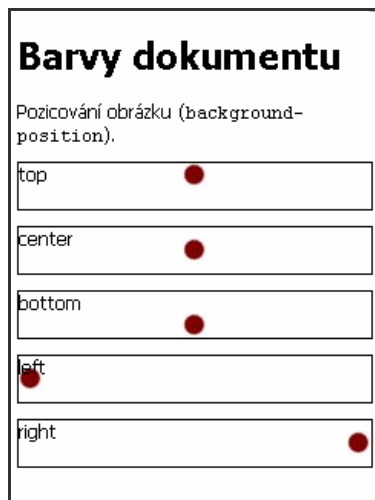
Pozicování pozadí pomocí klíčových slov.

```
<p style="background-position:top">top</p>
<p style="background-position:center">center</p>
<p style="background-position:bottom">bottom</p>
<p style="background-position:left">left</p>
<p style="background-position:right">right</p>
```

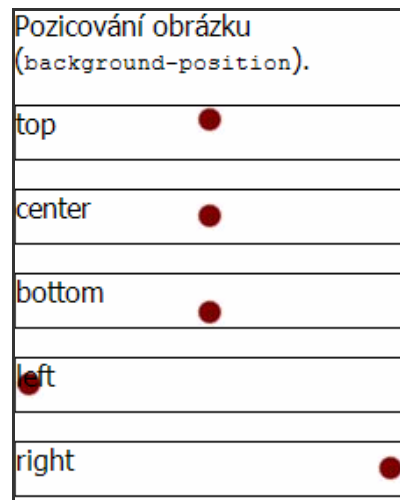


barvy/  
příklad6

## Ukázky



**Internet Explorer Mobile  
(Windows Mobile 6)**



**Opera 9.5 beta  
(Windows Mobile - Touchscreen)**

## Vlastnost pozadí

Vlastností `background` definujeme všechny dílčí vlastnosti pozadí najednou.

Hodnoty je možné uvést v jakémkoliv pořadí, oddělují se mezerou. Výchozí hodnoty viz dílčí vlastnosti.

## Příklad

Definice všech vlastností pozadí současně.

```
body {
background: url(krajina2.jpg) black 10% 20% fixed no-repeat }
```



barvy/  
příklad7

## Ukázky



**Internet Explorer Mobile  
(Windows Mobile 6)**



**Opera 9.5 beta  
(Windows Mobile - Touchscreen)**

### 3.4.17 Vlastnosti písma

Následující vlastnosti slouží pro formátování písma řádkového obsahu elementů.

#### Typ písma

Vlastností `font-family` určujeme typ písma zobrazovaného textu. Hodnotou je seznam názvů rodin písma, oddělený čárkami. Názvy, které obsahují mezeru, musí být v uvozovkách. Rodina písma může být buď konkrétní název písma, nebo některý generický (obecný) typ.

Rozeznáváme pět generických typů písma:

`serif`, `sans-serif`, `monospace`, `fantasy`, `cursive`,

každý z nich zastřešuje určitou skupinu fontů. Prohlížeč z nich vyberete ten, který je na klientské straně dostupný. Vlastnost je dědičná. Výchozí hodnota závisí na prohlížeči.

#### Příklad

Definice generických typů písma.

```
<p style="font-family:serif">Serif</p>
<p style="font-family:sans-serif">Sans-serif</p>
<p style="font-family:monospace">Monospace</p>
<p style="font-family:fantasy">Fantasy</p>
<p style="font-family:cursive">Cursive</p>
```



pismo/  
příklad1

#### Ukázky

##### Vlastnosti písma

Ukázka typů písma (`font-family`).

Serif

Sans-serif

Monospace

Fantasy

Cursive

Internet Explorer Mobile  
(Windows Mobile 6)

Ukázka typů písma (`font-family`).

Serif

Sans-serif

Monospace

Fantasy

Cursive

Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Styl písma

Vlastnost `font-style` definuje sklon písma. Hodnota `normal` (výchozí hodnota) značí neskloněné písmo, `italic` a `oblique` skloněné. Rozdíl mezi `italic` a `oblique` je v druhu použitého písma. Italic použije písmo označené `italic` a naopak. Vlastnost je dědičná.

### Příklad

Nastavení různých stylů písma.

```
<p style="font-style:normal">Normal</p>
<p style="font-style:italic">Italic</p>
<p style="font-style:oblique">Oblique</p>
```



pismo/  
příklad2

### Ukázky

**Vlastnosti písma**

Ukázka stylu písma (`font-style`).

Normal

*Italic*

*Oblique*

Internet Explorer Mobile  
(Windows Mobile 6)

Ukázka stylu písma (`font-style`).

Normal

*Italic*

*Oblique*

Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Varianta písma

Vlastnost `font-variant` určuje, zda je písmo nastaveno na `normal` (výchozí hodnota – základní písmo), nebo `small-caps` (kapitálky). Vlastnost je dědičná.

### Příklad

Definice varianty písma.

```
<p style="font-variant:normal">Normální styl ... </p>
<p style="font-variant:small-caps">Kapitálky ... </p>
```



pismo/  
příklad3

### Ukázky

**Vlastnosti písma**

Ukázka variant písma (`font-variant` ).

Normální styl písma (`font-variant:normal`)

KAPITÁLKY (`FONT-VARIANT:SMALL-CAPS`)

Internet Explorer Mobile  
(Windows Mobile 6)

Ukázka variant písma (`font-variant` ).

Normální styl písma (`font-variant:normal`)

KAPITÁLKY (`FONT-VARIANT:SMALL-CAPS`)

Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Tučnost písma

Vlastnost `font-weight` nastavuje sílu písma. Hodnotou může být některé z klíčových slov `normal` (normální), `bold` (tučné), `bolder` (tučnější), `lighter` (slabší), nebo čísel 100, 200, 300, 400, 500, 600, 700, 800, 900 (od nejslabšího po nejsilnější). Vlastnost je dědičná. Výchozí hodnotou je `normal`.

### Příklad

Definice tučností písma.

```
<h2>Klíčová slova</h2>
<p>
  <span style="font-weight:normal">normal</span>,
  <span style="font-weight:bold">bold</span>,
  <span style="font-weight:bolder">bolder</span>,
  <span style="font-weight:lighter">lighter</span>
</p>

<h2>Hodnoty</h2>
<p>
  <span style="font-weight:100">100</span>,
  <span style="font-weight:200">200</span>,
  <span style="font-weight:300">300</span>,
  <span style="font-weight:400">400</span>,
  <span style="font-weight:500">500</span>,
  <span style="font-weight:600">600</span>,
  <span style="font-weight:700">700</span>,
  <span style="font-weight:800">800</span>,
  <span style="font-weight:900">900</span>
</p>
```



pismo/  
příklad4

### Ukázky

**Vlastnosti písma**

Ukázka tučnosti písma (`font-weight`).

**Klíčová slova**

normal, **bold**, **bolder**, lighter

**Hodnoty**

100, 200, 300, 400, 500, **600**, **700**,  
**800**, **900**

Internet Explorer Mobile  
(Windows Mobile 6)

Ukázka tučnosti písma (`font-weight`).

**Klíčová slova**

normal, **bold**, **bolder**, lighter

**Hodnoty**

100, 200, 300, 400, 500, **600**,  
**700**, **800**, **900**

Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Velikost písma

Vlastností `font-size` definujeme velikost písma. Hodnotu můžeme určit v relativních i absolutních jednotkách, nebo za pomoci klíčových slov. Klíčová slova také dělíme na relativní (vztahují se k velikosti aktuálního písma rodičovského prvku) a absolutní.

**Absolutní:** `xx-small`, `x-small`, `small`, `medium`, `large`, `x-large`, `xx-large` a **relativní** `larger`, `smaller`.

Koeficient pro převod mezi jednotlivými stupni = 1,2. Výchozí hodnotou je `medium`. Vlastnost je dědičná.

### Příklad

Deklarace různých velikostí písem.

```
<h2>Klíčová slova</h2>
<h3>Absolutní</h3>
<p>
  <span style="font-size:xx-small">xx-small</span>,
  <span style="font-size:x-small">x-small</span>,
  <span style="font-size:small">small</span>,
  <span style="font-size:medium">medium</span>
  <span style="font-size:large">large</span>
  <span style="font-size:x-large">x-large</span>
  <span style="font-size:xx-large">xx-large</span>
</p>
<h3>Relativní</h3>
<p>
  <span style="font-size:larger">larger</span>
  <span style="font-size:smaller">smaller</span>
</p>
```



pismo/  
příklad5

### Ukázky

**Vlastnosti písma**

Ukázka velikosti písma (`font-size`).

**Klíčová slova**

**Absolutní**

`xx-small`, `x-small`, `small`, `medium` `large` `x-large` `xx-large`

**Relativní**

`larger` `smaller`

Internet Explorer Mobile  
(Windows Mobile 6)

**Absolutní**

`xx-small`, `x-small`, `small`, `medium`

`large` `x-large` `XX-large`

**Relativní**

`larger` `smaller`

Opera 9.5 beta  
(Windows Mobile - Touchscreen)



## Vlastnost písma

Vlastnost `font` nastavuje současně všechny dílčí vlastnosti písma. Při definování hodnoty je potřeba dodržet jistá pravidla:

- hodnoty `font-size` a `font-family` jsou povinné
- před nimi jsem nepovinně umístěny hodnoty `font-style`, `font-variant`, `font-weight` v libovolném pořadí

Jako oddělovač slouží znak mezera. Vlastnost je dědičná. Výchozí hodnoty záleží na dílčích vlastnostech.

## Příklad

Definice všech dílčích vlastností písma současně.

```
span.pismo1 {font:12pt Arial,sans-serif}
span.pismo2
{font:small-caps italic bold 12pt Arial,sans-serif}
```



## Ukázky

### Vlastnosti písma

Ukázka vlastnosti `font`.

Velikost písma 12pt, styl písma Arial,sans-serif,  
**VELIKOST PÍSMÁ 12PT, TUČNÁ  
 KURZÍVA, KAPITÁLKY, STYL PÍSMÁ  
 ARIAL,SANS-SERIF.**

Internet Explorer Mobile  
(Windows Mobile 6)

### Vlastnosti písma

Ukázka vlastnosti `font`.

Velikost písma 12pt, styl písma Arial,sans-serif,  
**VELIKOST PÍSMÁ 12PT, TUČNÁ  
 KURZÍVA, KAPITÁLKY, STYL PÍSMÁ  
 ARIAL,SANS-SERIF.**

Opera 9.5 beta  
(Windows Mobile - Touchscreen)



### 3.4.18 Seznamy

Následující vlastnosti mají smysl pouze pro elementy, které jsou označené jako položky seznamu (`display:list-item`)

#### Typ seznamu

Vlastnost `list-style-type` slouží pro definici úvodních značek u položek seznamů. Použijte se, pokud není uvedena vlastnost `list-style-image` (o tom dále).

Hodnota je tvořena některým z klíčových slov:

- `disc` (kruh), `circle` (kružnice), `square` (čtverec) – podoba záleží na interpretaci prohlížeče
- `decimal` – arabské číslice (1,2,3,4....)
- `lower-roman` – malé římské číslice (i, ii, iii, iv...)
- `upper-roman` – velké římské číslice (I, II, III, IV...)
- `lower-alpha` – malé ASCII znaky (a, b, c, d ...)
- `upper-alpha` – velké ASCII znaky (A, B, C, D ...)
- `none` – žádná značka

Výchozí hodnotou je `disc`. Vlastnost je dědičná.

#### Příklad

Definice odrážek seznamu.

```
<ul>
  <li style="list-style-type:disc">disc</li>
  <li style="list-style-type:circle">circle</li>
  <li style="list-style-type:square">square</li>
  <li style="list-style-type:decimal">decimal</li>
  <li style="list-style-type:lower-roman">lower-roman</li>
  <li style="list-style-type:upper-roman">upper-roman</li>
  <li style="list-style-type:lower-alpha">lower-alpha</li>
  <li style="list-style-type:upper-alpha">upper-alpha</li>
  <li style="list-style-type:none">none</li>
</ul>
```

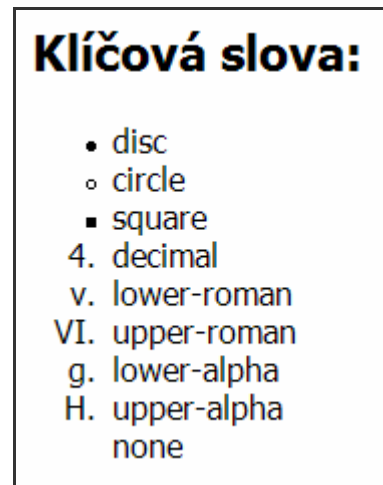


seznamy/  
příklad1

## Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Obrázkové značení

Vlastnost `list-style-image` definuje obrázkové značky, zobrazované u položek seznamů. Hodnotou je URL obrázku. Výchozí hodnotou je potom `none` (žádná značka). Vlastnost je dědičná.

## Příklad

Obrázek jako odrážka u položek seznamu.

```
ul {list-style-image: url(kytka.jpg)}
```



seznamy/  
priklad2

## Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Pozice značení

Kam umístit značku seznamu udává vlastnost `list-style-position`. Přesněji definuje umístění generovaného doplňkového rámu vůči rámu hlavnímu. Hodnoty zastupují klíčová slova `inside` (uvnitř hlavního rámu) a `outside` (vně hlavního rámu). Výchozí hodnotou je `outside`. Vlastnost je dědičná.

### Příklad

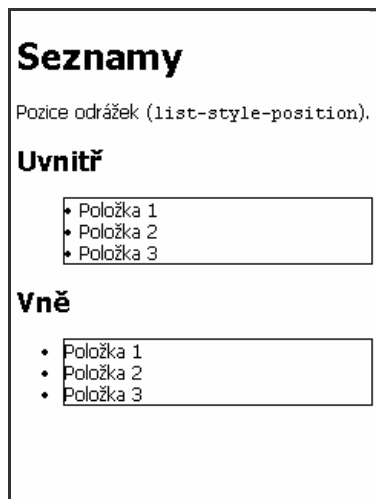
Umístění odrážek u seznamů.

```
ul.prvni {list-style-position:inside}
ul.druhy {list-style-position:outside}
```

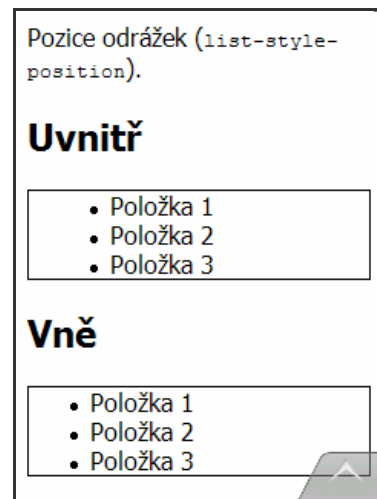


seznamy/  
priklad3

### Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Vlastnost seznamů

Pomocí vlastnosti `list-style` uvádíme v jedné definici všechny dílčí vlastnosti seznamů. Ty lze uvést v libovolném pořadí. Výchozí hodnoty záleží na dílčích vlastnostech. `List-style` je dědičná.

### Příklad

Styly pro jednotlivé položky seznamu.

```
li.prvni {list-style: url(kytka.jpg) inside square}
li.druhy {list-style: outside lower-roman}
li.treti {list-style: inside upper-alpha}
```

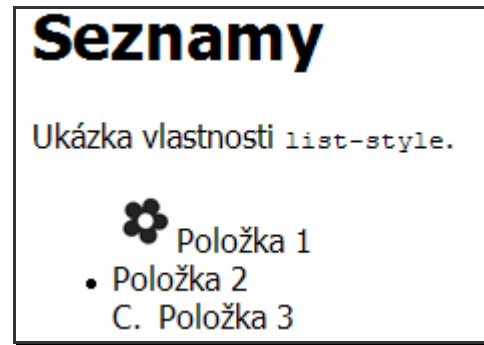


seznamy/  
priklad4

## Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## 3.4.19 Vlastnosti textu

Následující vlastnosti slouží k formátování textu.

## Odsazení první řádky

Odsazení první řádky textu od levého okraje omezujícího bloku zajišťuje vlastnost `text-indent`. Její hodnotu lze zadat konkrétní velikostí, nebo ji uvést v procentech (vztahuje se k šířce omezujícího bloku). Výchozí hodnotou je 0. Vlastnost je dědičná.

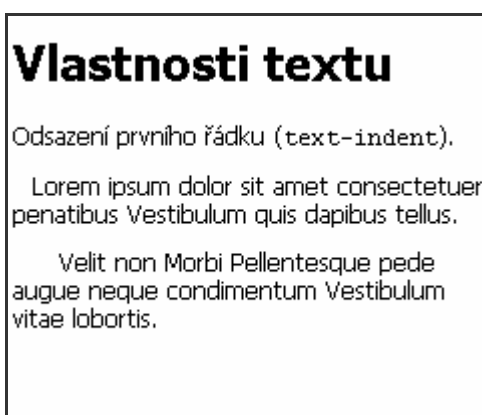
## Příklad

In-line definice odsazení prvních řádků odstavců.

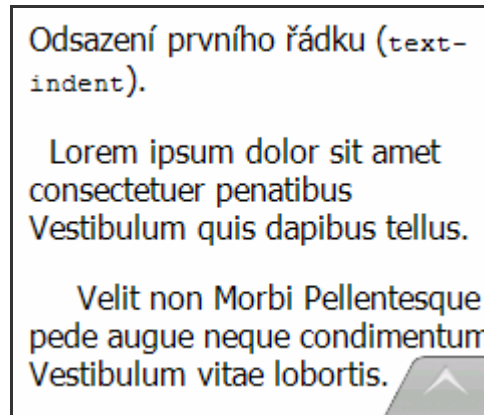
```
<p style="text-indent:10px">Lorem ipsum ...</p>
<p style="text-indent:10%">Velit non Morbi ...</p>
```



## Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Vodorovné zarovnání textu

Řádkový obsah bloků je možné vertikálně zarovnat vlastností `text-align`. Hodnotu zadáváme pomocí klíčových slov `left` (zarovnání k levé hraně rodičovského bloku), `center` (zarovnání uprostřed rodičovského bloku), `right` (zarovnání k pravé hraně rodičovského bloku), `justify` (zarovnání do bloku).

Výchozí hodnota závisí na prohlížeči. Vlastnost je dědičná.

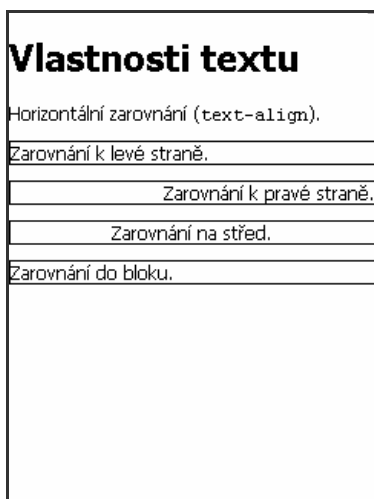
### Příklad

Zarovnání obsahu odstavců uvnitř dokumentu.

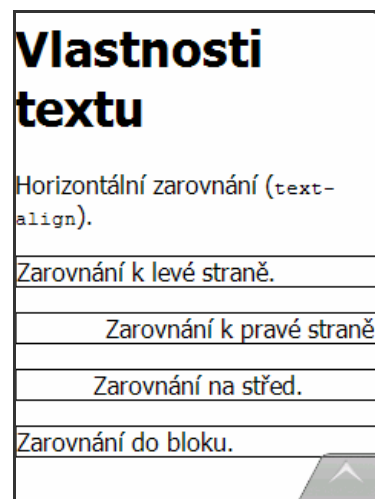
```
<p style="border-style:none">Horizontální ...</p>
<p style="text-align:left">Zarovnání k levé straně.</p>
<p style="text-align:right">Zarovnání k pravé straně.</p>
<p style="text-align:center">Zarovnání na střed.</p>
<p style="text-align:justify">Zarovnání do bloku.</p>
```



### Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Dekorace textu

Vlastnost `text-decoration` přidává dekorace k formátovanému textu. U jakéhokoliv jiného obsahu je ignorována. Lze ji uvést u blokových (platí na celý obsah prvku) i řádkových elementů. Hodnotu tvoří klíčová slova `underline` (podtržení), `none` (vynulování dekorace), `blink` (blikání textu). Výchozí hodnotou je `none`.

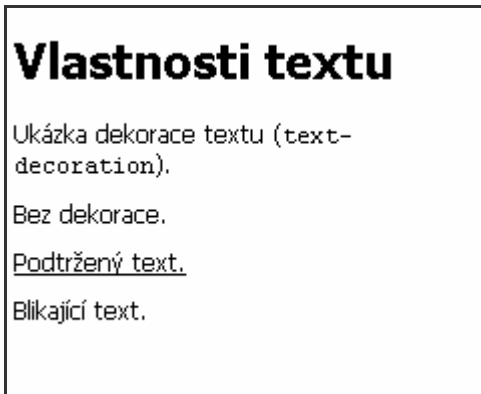
text/  
příklad3

## Příklad

Nastavení dekorací písma odstavců.

```
<p style="text-decoration:none">Bez dekorace.</p>
<p style="text-decoration:underline">Podtržený text.</p>
<p style="text-decoration:blink">Blikající text.</p>
```

## Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Transformace textu

Pro transformaci mezi malými a velkými znaky slouží vlastnost `text-transform`.

Hodnotu udávají klíčová slova:

- `capitalize` – převod prvních písmen slov na velké
- `uppercase` – převod všech znaků na velké (verzálky)
- `lowercase` – převod všech znaků na malé

Výchozí hodnotou je `none`, tedy žádná transformace. Vlastnost je dědičná.

## Příklad

Transformace písma.

```
<p style="text-transform:capitalize">
Všechna počáteční písmena slov transformována na velká.</p>
<p style="text-transform:uppercase">
Všechny znaky v textu transformovány na velké.</p>
<p style="text-transform:lowercase">
VŠECHNY ZNAKY V TEXTU TRANSFORMOVÁNY NA MALÉ.</p>
```

text/  
příklad4

## Ukázky

### Vlastnosti textu

Ukázka transformace textu (`text-transform`).

Všechna Počáteční Písma Sloz Transformována Na Velká.

VŠECHNY ZNAKY V TEXTU TRANSFORMOVÁNY NA VELKÉ.

všechny znaky v textu transformovány na malé.

Internet Explorer Mobile  
(Windows Mobile 6)

## Vertikální zarovnání

Vlastnost `vertical-align` ovlivňuje vertikální zarovnání řádkového obsahu uvnitř bloku. Obsah je členěn do bloků řádek, jejich výška se řídí podle největší prvku v řádku. S jednotlivými prvky je možné vertikálně manipulovat. Hodnoty vlastnosti jsou následující:

- `baseline` – zarovnání na účaří
- `sub` – dolní index
- `super` – horní index
- `top` – zarovnání k hornímu rámu
- `middle` – zarovnání na střed
- `bottom` – zarovnání k dolnímu rámu

Výchozí hodnotou je `baseline`.

## Příklad

Definice různých zarovnání obsahu elementů.

```
<span style="vertical-align:baseline">baseline</span>
<span style="vertical-align:sub">sub</span>
<span style="vertical-align:super">super</span>
<span style="vertical-align:top">top</span>
<span style="vertical-align:middle">middle</span>
<span style="vertical-align:bottom">bottom</span>
```

## Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)

## Bílé znaky

Zpracování bílých znaků řídí vlastnost `white-space`. Za hodnotu zvolíme jednu z klíčových slov:

- `normal` (výchozí hodnota) – všechny mezery mohou být zalomeny, více bílých znaků je považováno za jednu mezeru
- `nowrap` – řádky mohou být zalomeny pouze na definovaných místech (sekvence `\A`)
- `pre` – zachováno formátování z kódu dokumentu

## Příklad

Formátování bílých znaků.

```
<p style="white-space:normal">
Všechny mezery v tomto textu mohou být zalomeny.</p>
<p style="white-space:nowrap">
Text může být zalomen na <br />předem určených místech.</p>
<p style="white-space:pre">
Veškeré bílé znaky
z kódu zachovány.
</p>
```





## Ukázky

Zalamování (white-space).

Všechny mezery v tomto textu mohou

Text může být zalomen na  
předem určených místech.

Veškeré bílé znaky  
z kódu zachovány.

Emulátor Fennec/Firefox 1.0a1

Zalamování (white-space).

Všechny mezery v tomto textu  
mohou být zalomeny.

Text může být zalomen na  
předem určených místech.

Veškeré bílé znaky  
z kódu zachovány.

Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## 3.4.20 Viditelnost

Vlastnost `visibility` nastavuje vizuální viditelnost rámců elementu. Může nabývat hodnot:

- `visible` - prvek je viditelný, výchozí hodnota
- `hidden` - prvek je průhledný
- `collapse` - stejné jako `hidden`, používá se v tabulkách pro zneviditelnění buněk.

## Příklad

Nastavení viditelnosti prvků.

```
<p style="visibility:visible">
Viditelný prvek (visible).</p>
<p style="visibility:hidden">
Průhledný prvek (hidden).</p>
<p style="visibility:collapse">
Průhledný prvek (collapse).</p>
```

viditelnost/  
příklad1

## Ukázky

**Viditelnost  
prvků**

Ukázka viditelnosti rámců  
elementu(`visibility`).

Viditelný prvek (`visible`).

Skyfire 0.9.333  
(Windows Mobile)

**Viditelnost  
prvků**

Ukázka viditelnosti rámců  
elementu(`visibility`).

Viditelný prvek (`visible`).

Opera 9.5 beta  
(Windows Mobile - Touchscreen)

### 3.5 WAP CSS rozšíření

Následující vlastnosti jsou rozšířením probrané podmnožiny kaskádových stylů.

#### 3.5.1 Rolování obsahu

Pokud u elementů uvedeme vlastnost `display` s hodnotou `-wap-marquee`, veškerý jejich obsah bude rolován napříč obrazovkou mobilního zařízení. Obsahem může být prostý text, obrázek atd. Výsledkem je rozšířený blokový rám.

Jak bude obsah rolován nastavují vlastnosti:

- `-wap-marquee-style`
- `-wap-marquee-loop`
- `-wap-marquee-dir`
- `-wap-marquee-speed`

#### Příklad

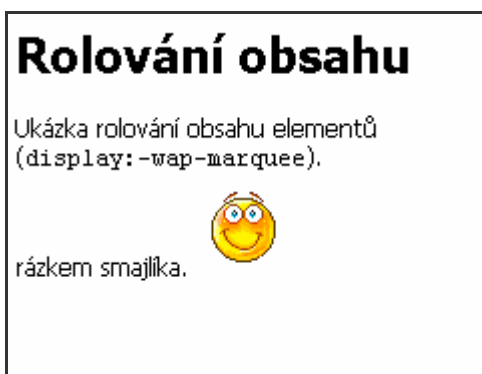
Definice rolování obsahu elementu.

```
<p style="display:-wap-marquee">
Text v odstavci s obrázkem smajlíka.
</p>
```

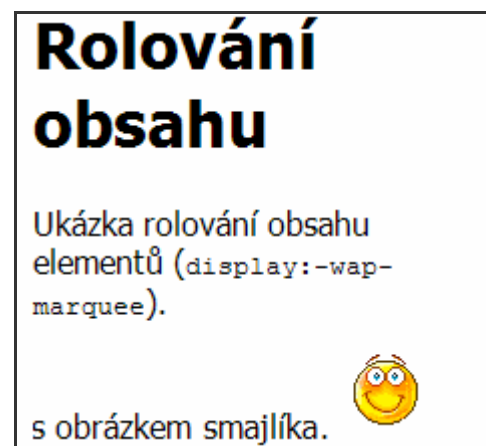


rozsireni/  
priklad1

#### Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Směr rolování obsahu (`-wap-marquee-dir`)

Vlastnost `-wap-marquee-dir` určuje směr rolování obsahu prvku. Hodnota `ltr` (left to right) způsobí rolování zleva doprava, naopak `rtl` (right to left) zprava doleva. Výchozí je rolování `rtl`.

### Příklad

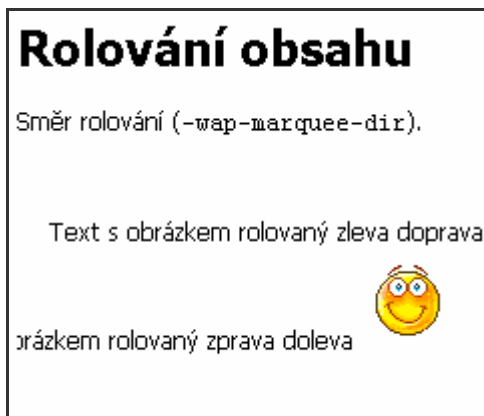
Definice směru rolování obsahu.

```
p.prvni, p.druhy {display:-wap-marquee}
p.prvni  {-wap-marquee-dir:rtl}
p.druhy  {-wap-marquee-dir:ltr}
```

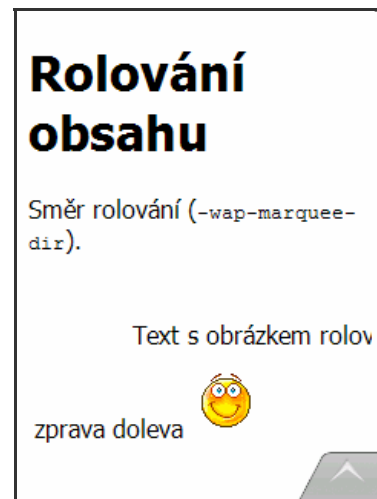


rozsireni/  
priklad2

### Ukázky



Internet Explorer Mobile  
(Windows Mobile 6)



Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## Počet opakování (`-wap-marquee-loop`)

Vlastnost `-wap-marquee-loop` řídí opakované rolování obsahu prvku. Výchozí hodnotou je 1, tj. obsah bude přenesen přes obrazovku pouze jednou. Lze nastavit jakékoliv celé číslo, nebo klíčové slovo `infinite` (nekonečné rolování).

### Příklad

Počet opakování rolování obsahu.

```
p.prvni, p.druhy {display:-wap-marquee}
p.prvni  {-wap-marquee-loop:2}
p.druhy  {-wap-marquee-loop:infinite}
```



rozsireni/  
priklad3

### Rychlost rolování (`-wap-marquee-speed`)

Vlastnost `-wap-marquee-speed` nastavuje rychlost přechodu obsahu přes obrazovku zařízení. Hodnotou je některé z klíčových slov `slow` (pomalé rolování), `normal` (běžné rolování), `fast` (rychlé rolování). Výchozí hodnotou je `normal`.

#### Příklad

Definice rychlosti rolování obsahu.

```
p.prvni {-wap-marquee-speed:slow}
p.druhy {-wap-marquee-speed:normal}
p.treti {-wap-marquee-speed:fast}
```



rozsireni/  
priklad4

### Styl rolování (`-wap-marquee-style`)

Vlastnost `-wap-marquee-style` definuje způsob rolování obsahu prvků.

Může nabývat hodnot:

- `scroll` – Nejprve je obsah prvku schován za levou resp. pravou hranou omezujícího bloku, poté je plynule přesunut **za** hranu protější.
- `slide` – Nejprve je obsah prvku schován za levou resp. pravou hranou omezujícího bloku, poté je plynule přesunut **před** hranou protější.
- `alternate` – Nejprve je obsah prvku umístěn **před** levou resp. pravou hranou omezujícího bloku, poté je plynule přesunut **před** hranu protější.

Výchozí hodnotou je `scroll`.

#### Příklad

Způsob rolování obsahu.

```
p.prvni {-wap-marquee-style:scroll}
p.druhy {-wap-marquee-style:slide}
p.treti {-wap-marquee-style:alternate}
```



rozsireni/  
priklad5

## 3.5.2 Přístupové klávesy

Vlastnost `-wap-accesskey` definuje přístupové klávesy, díky kterým lze přistupovat k jednotlivým prvkům v dokumentu. Hodnotami mohou být klávesy 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \*, #, nebo klíčové slovo `none` (žádná klávesa – výchozí hodnota). Uvádíme-li více hodnot najednou, oddělujeme je znakem „,” (čárka).

## Příklad

Definice přístupových kláves mobilního zařízení.

```
a.prvni {-wap-accesskey:0,1,2,3,4}
a.druhy {-wap-accesskey:5,6,7,8,9}
a.treti {-wap-accesskey:*}
a.ctvrty {-wap-accesskey:#}
a.paty {-wap-accesskey:none}
```



rozsireni/  
priklad6

### 3.5.3 Rozšíření input

Vlastnosti `-wap-input-format` a `-wap-input-required` slouží pro upřesnění vstupních hodnot elementu `input`.

#### Formát vstupu (`-wap-input-format`)

Typ a množství znaků v textovém poli určuje vlastnost `-wap-input-format`, lze ji použít u textových vstupů: `<input type="text">`, `<input type="password">`, `<textarea>`. U ostatních elementů nemá význam.

Hodnotou je omezující vstupní maska:

Formátovací znaky	Použití
<b>a</b>	Zastupuje malá písmena, nebo symbolické znaky.
<b>A</b>	Zastupuje velká písmena, nebo symbolické znaky.
<b>n</b>	Zastupuje numerickou hodnotu, nebo symbolické znaky.
<b>N</b>	Zastupuje numerickou hodnotu.
<b>x</b>	Zastupuje malá písmena, numerickou hodnotu, nebo symbolické znaky.
<b>X</b>	Zastupuje velká písmena, numerickou hodnotu, nebo symbolické znaky.
<b>m</b>	Zastupuje všechny znaky. Očekávána malá písmena, není to však podmínkou.
<b>M</b>	Zastupuje všechny znaky. Očekávána velká písmena, není to však podmínkou.

Výchozí hodnotou je „\*M“, hvězdičku interpretujeme jako neomezený počet (M) znaků. Před libovolné písmeno můžeme také vložit číslo, to reprezentuje počet příslušných znaků v textu. Sestavení masky má svá omezení, pokud přidáme číselnou

hodnotu nebo znak „\*“ před některý formátovací znak, je možné danou kombinaci uvést v masce pouze jednou, a to na samém konci řetězce.

Např.: špatně 10Aaa10A – 10A musí být v řetězci pouze jednou - na konci

špatně \*N10m\*Naa – \*N musí být v řetězci pouze jednou - na konci

Masku lze doplnit o tzv. escape sekvence. Ty poslouží lepšímu formátování vstupu. Zapisují se za znaky „\“ (např. deklarace datumu aj.).

### Příklad

Definice vstupní masky pole.

```
input.inicialy {-wap-input-format:AA}
input.jmeno, input.prijmeni {-wap-input-format:A*a}
input.datum {-wap-input-format:NN\\ / NN\\ / NNNN}
input.email {-wap-input-format:*\\ @*\\ .*}
input.zprava {-wap-input-format:*M}
```



rozsireni/  
priklad7

### Prázdná hodnota (-wap-input-required)

Vlastnost `-wap-input-required` povoluje, resp. zakazuje textovému poli prázdný obsah. Hodnota `true` znamená, že uživatel musí zadat nějaký textový vstup, naopak `false` prázdný vstup umožňuje. Výchozí hodnotou je `none` (nedefinováno).

### Příklad

Povolení/zakázání prázdných hodnot.

```
input.prazdne {-wap-input-required:false}
input.neprazdne {-wap-input-required:true}
```



rozsireni/  
priklad8

### Ukázky

Prázdná vstupní hodnota (-wap-input-required).

`-wap-input-required:false`

Titul

`-wap-input-required:true`

Vaše jméno

The input is not valid.

input

Static

The input is not valid.

OK Cancel

vase jméno

Opera 9.5 beta  
(Windows Mobile - Touchscreen)

## IV. Závěr

Problematika protokolu WAP a tvorby dokumentů pro mobilní zařízení je velmi rozsáhlá. Jedním z hlavních důvodů je kompatibilita s technologií WWW, tj. platí zde stejné principy, využívají se zde i stejné prostředky.

V práci jsem se snažil podat ucelený pohled na danou problematiku. Byla podrobně rozebrána technologie WAP, její architektura a fungování. Bylo provedeno porovnání se staršími verzemi WAP 1.x a uvedeno závěrečné zhodnocení.

Dále byly zpracovány všechny aktuální programovací jazyky, určené pro obrazovky mobilních zařízení. Jejich syntax jsem shrnul do sady PSPad šablon, díky kterým lze snadno a rychle vytvářet WAP stránky všech příslušných verzí jazyků.

Zvláštní pozornost je věnována jazykům XHTML-MP, ty jsou v průběhu práce rozebírány a následně prezentovány na praktických příkladech. Závěrem druhé kapitoly uvádím rozdíly oproti staršímu, ale hojně využívanému jazyku WML.

Poslední kapitola patří bezdrátovému profilu kaskádových stylů, zde je podobně jako v XHTML-MP rozebrán syntax jazyka a následně prezentován na praktických příkladech.

Příklady jsou vytvářeny formou mobilní webové prezentace a umístěny na příloženém CD, nebo na internetové adrese <http://www.webmobile.comuv.com/>.

Při jejich vytváření jsem se setkal s řadou problému, které od počátku provázejí tvůrce klasických WWW stránek. Jedná se v první řadě o otázku kompatibility s nejnovějšími standardy. Jejich podpora je v mobilních prohlížečích velmi rozmanitá, a to jak v technologii XHTML-MP, tak především ve WCSS.

Pro lepší názornost jsou příklady doplněny o ukázky v mobilních prohlížečích. Nalézt vhodný prohlížeč bylo mnohdy velmi obtížné. Nakonec jsem si vystačil se třemi základními (Internet Mobile Explorer, SkyFire, Opera Mobile) a jedním emulátorem (Fennec).

Věřím, že cíle práce byly splněny v plném rozsahu.

## V. Literatura

- [1] Open Mobile Alliance. *Wireless Application Protocol* [online]. c2009 [cit. 2009-01-01]. Text v angličtině. Dostupný z WWW: <<http://www.openmobilealliance.org/Technical/wapindex.aspx>>.
- [2] WAP Forum. *WAP Architecture*. [s.l.] : [s.n.], 2001. 24 s. Dostupný z WWW: <<http://www.openmobilealliance.org/tech/affiliates/wap/wap-210-waparch-20010712-a.pdf>>.
- [3] WAP Forum. *XHTML Mobile Profile*. [s.l.] : [s.n.], 2001. 17 s. Dostupný z WWW: <<http://www.openmobilealliance.org/tech/affiliates/wap/wap-277-xhtmlmp-20011029-a.pdf>>.
- [4] WAP Forum. *WAP CSS Specification*. [s.l.] : [s.n.], 2001. 64 s. Dostupný z WWW: <<http://www.openmobilealliance.org/tech/affiliates/wap/wap-239-wcss-20011026-a.pdf>>.
- [5] WAP Forum. *WAG UAProf*. [s.l.] : [s.n.], 2001. 86 s. Dostupný z WWW: <<http://www.openmobilealliance.org/tech/affiliates/wap/wap-248-uaprof-20011020-a.pdf>>.
- [6] WAP Forum. *WAP Architecture*. [s.l.] : [s.n.], 1998. 20 s. Dostupný z WWW: <[http://www.openmobilealliance.org/tech/affiliates/LicenseAgreement.asp?DocName=/wap/Technical\\_June2000-20021106%5B1%5D.zip](http://www.openmobilealliance.org/tech/affiliates/LicenseAgreement.asp?DocName=/wap/Technical_June2000-20021106%5B1%5D.zip)>.
- [7] *Developers\' Home* [online]. c2008 [cit. 2009-01-01]. Anglický text. Dostupný z WWW: <<http://www.developershome.com/>>.
- [8] KOKOŠKA , Lukáš . *Mobilní web .info* [online]. [2007] [cit. 2009-04-01]. Dostupný z WWW: <<http://www.mobilniweb.info/>>.
- [9] PETERKA, Jiří. *Archiv článků a přednášek Jiřího Peterky* [online]. [2000] [cit. 2009-04-01]. Dostupný z WWW: <<http://www.earchiv.cz/>>.
- [10] STANÍČEK, Petr. *Kompletní průvodce CSS*. [s.l.] : [s.n.], 2003. 178 s. ISBN 80-7226-872-4.
- [11] *Interval.cz* [online]. [1998] [cit. 2009-04-01]. Dostupný z WWW: <<http://interval.cz/>>. ISSN 1212-8651.
- [12] JANOVSKEÝ, Lukáš. *Jak psát web* [online]. [2003] , 31.3.2009 [cit. 2009-04-01]. Dostupný z WWW: <<http://www.jakpsatweb.cz/>>.
- [13] *XHTML*. 2002-2003. Dostupný z WWW: <<http://interval.cz/clanky/xhtml-vyvoj-x-html-a-jeho-moznosti/>>.



- [14] W3C. *XHTML™ Modularization 1.1* [online]. 2008 [cit. 2009-04-01]. Dostupný z WWW: <<http://www.w3.org/TR/xhtml-modularization/>>.
- [15] *Modularization of XHTML™* [online]. 2000 [cit. 2009-04-01]. Dostupný z WWW: <<http://www.w3.org/TR/2000/CR-xhtml-modularization-20001020/>>.
- [16] User Datagram Protocol. *RFC 768* [online]. 1980 [cit. 2009-03-11]. Dostupný z WWW: <<http://tools.ietf.org/html/rfc768>>.
- [17] Transmission Control Protocol . *RFC793* [online]. 1981 [cit. 2009-03-11]. Dostupný z WWW: <<http://www.faqs.org/rfcs/rfc793.html>>.
- [18] FIELDING, R., et al. Hypertext Transfer Protocol -- HTTP/1.1. *RFC2616* [online]. 1999 [cit. 2009-03-11]. Dostupný z WWW: <<http://tools.ietf.org/html/rfc2616>>.
- [19] KRISTOL, D., et al. HTTP State Management Mechanism. *RFC2965* [online]. 2000 [cit. 2009-03-11]. Dostupný z WWW: <<http://tools.ietf.org/html/rfc2965>>.
- [20] *XHTML™ Basic* [online]. 2000 [cit. 2009-04-01]. Dostupný z WWW: <<http://www.w3.org/TR/2000/REC-xhtml-basic-20001219/>>.
- [21] *XHTML™ Basic 1.1* [online]. 2008 [cit. 2009-04-01]. Dostupný z WWW: <<http://www.w3.org/TR/xhtml-basic/>>.
- [22] Open Mobile Alliance. *XHTML Mobile Profile : Draft Version 1.2 – 22 Dec 2004*. [s.l.] : [s.n.], 2004. 39 s. Dostupný z WWW: <[http://users.jyu.fi/~mweber/teaching/docs/xhtml-mp/OMA-XHTMLMP-V1\\_2-20041222-D.pdf](http://users.jyu.fi/~mweber/teaching/docs/xhtml-mp/OMA-XHTMLMP-V1_2-20041222-D.pdf)>.
- [23] Open Mobile Alliance. *XHTML Mobile Profile : Approved Version 1.1 – 20 Oct 2006*. [s.l.] : [s.n.], 2006. 36 s. Dostupný z WWW: <[http://www.openmobilealliance.org/technical/release\\_program/docs/Browsing/V2\\_1-20061020-A/OMA-WAP-XHTMLMP-V1\\_1-20061020-A.pdf](http://www.openmobilealliance.org/technical/release_program/docs/Browsing/V2_1-20061020-A/OMA-WAP-XHTMLMP-V1_1-20061020-A.pdf)>.

## VI. Přílohy

Příloha A – Tabulka syntaxe jazyků pro mobilní zařízení

Modul	XHTML Basic 1.0	XHTML MP1.1	XHTML MP1.2	XHTML Basic 1.1
Structure	body, head, html, title	<i>stejné</i>	<i>stejné</i>	<i>stejné</i>
Text	abbr, acronym, address, block quote, br, cite, code, dfn, div, em, h1, h2, h3, h4, h5, h6, kbd, p, pre, q, samp, span, strong, var	<i>stejné</i>	<i>stejné</i>	<i>stejné</i>
Hypertext	a	<i>stejné</i>	<i>stejné</i>	<i>stejné</i>
List	dl, dt, dd, ol, ul, li	<i>stejné</i>	<i>stejné</i>	<i>stejné</i>
Basic Forms	form, input, label, select, option, textarea	<i>stejné</i>	<i>stejné</i>	<i>není</i>
Forms	<i>není</i>	fieldset, optgroup	<i>stejné</i>	button, fieldset, form, input, label, legend, select, optgroup, option, textarea
Basic Tables	caption, table, td, th, tr	<i>stejné</i>	<i>stejné</i>	<i>stejné</i>
Image	img	<i>stejné</i>	<i>stejné</i>	<i>stejné</i>
Object	object, param	<i>stejné</i>	<i>stejné</i>	<i>stejné</i>
Meta information	meta	<i>stejné</i>	<i>stejné</i>	<i>stejné</i>
Link	link	<i>stejné</i>	<i>stejné</i>	<i>stejné</i>
Base	base	<i>stejné</i>	<i>stejné</i>	<i>stejné</i>
Legacy	<i>není</i>	start attribute on ol, value attribute on li	<i>stejné</i>	odebrán start atribut
Presentation	<i>není</i>	b, big, hr, i, small	<i>stejné</i>	b, big, hr, i, small, sub, sup, tt

Style	<i>není</i>	style element, style atribut	<i>stejné</i>	<i>stejné</i>
Text Input Modes (OMA)	<i>není</i>	<i>není</i>	inputmode attribute v input, textarea	<i>stejné</i>
Intrinsic Events	<i>není</i>	<i>není</i>	onblur, onfocus, onreset, onsubmit, onload, onunload, onchange, onselect, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup	<i>stejné</i>

**Příloha B** - Tabulka předdefinovaných barev

Zástupná hodnota barev	RGB hodnota	RGB hexadecimální hodnota
Aqua	rgb(0, 255, 255)	#00ffff
Black	rgb(0, 0, 0)	#000000
Blue	rgb(0, 0, 255)	#0000ff
Fuchsia	rgb(255, 0, 255)	#ff00ff
Gray	rgb(128, 128, 128)	#808080
Green	rgb(0, 128, 0)	#008000
Lime	rgb(0, 255, 0)	#00ff00
Maroon	rgb(128, 0, 0)	#800000
Navy	rgb(0, 0, 128)	#000080
Olive	rgb(128, 128, 0)	#808000
Purple	rgb(128, 0, 128)	#800080
Red	rgb(255, 0, 0)	#ff0000
Silver	rgb(192, 192, 192)	#c0c0c0
Teal	rgb(0, 128, 128)	#008080
White	rgb(255, 255, 255)	#ffffff
Yellow	rgb(255, 255, 0)	#ffff00

**Zdroj:**

*Common Types of Property Value : Color.* [s.l.] : [s.n.], c2008. Dostupný z WWW: <[http://www.developershome.com/wap/wcss/wcss\\_tutorial.asp?page=commonPropertyValues](http://www.developershome.com/wap/wcss/wcss_tutorial.asp?page=commonPropertyValues)>. s. 14.

**Příloha C** – Příložené CD obsahující ukázkové příklady probírané problematiky aj.