

**Tvorba prezentací pomocí \LaTeX u ve třídě
BEAMER**

**Creation of presentations by \LaTeX in class
BEAMER**

Bakalářská práce

Josef Stern

Vedoucí bakalářské práce: Mgr. Jiří Pech, Ph.D.

Jihočeská univerzita v Českých Budějovicích

Pedagogická fakulta

Katedra informatiky

2011

Prohlášení

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

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

V Českých Budějovicích dne

Anotace

Tato bakalářská práce seznamuje uživatele systému \LaTeX s tvorbou prezentací pomocí třídy BEAMER. V této práci se čtenář seznámí s tím, jak by měl postupovat při tvorbě, získá informace o pravidlech tvorby prezentací a zejména podrobný popis příkazů této třídy a jejich vlastností. V textu jsou teoretické části doplněny o názorné ukázky zdrojového kódu.

Abstract

This bachelor thesis introduces creating of a presentation in a BEAMER class to \LaTeX system users. At this thesis the reader will learn how to proceed with the creation, he will get information about rules of the creation of presentations and especially will get a detailed description of the commands of this class and their properties. Theoretical parts are supplemented by demonstration of a source code in the text.

Poděkování

Rád bych poděkoval panu Mgr. Jiřímu Pechovi, Ph.D. za to, že mne během studia seznámil s filosofií opensource a public domain programů, díky čemuž je, i když práce s nimi nemusí být vždy pohodlná, dokáží nyní ocenit, stejně tak jako jejich autory. Dále bych chtěl poděkovat svým rodičům za podporu a neobyčejnou dávku trpělivosti, kterou se mnou během mé práce měli a také přátelům, kteří mi s touto prací pomohli.

Obsah

1	Úvod a cíl práce	6
1.1	Úvod do problematiky	6
1.2	Cíl práce	8
2	Metodika řešení	9
3	Výchozí stav problematiky	10
4	Pojem třída a prezentační možnosti v \LaTeXu	11
4.1	Třída	11
4.2	Možnosti prezentací pomocí \LaTeX u	11
4.2.1	Třída slides	12
4.2.2	Třída foiltex	12
4.2.3	Třída prosper	12
4.2.4	Třída texpower	13
5	Třída BEAMER	14
5.1	Instalace	14
5.2	Rady a pravidla pro tvorbu prezentace	14
5.2.1	Struktura prezentace	15
5.2.1.1	Časové rozpětí	15
5.2.1.2	Globální struktura	15
5.2.1.3	Struktura rámce	18
5.2.1.4	Interaktivní prvky	20
5.2.2	Použití grafiky	20
5.2.3	Výběr vhodných témat	21
5.2.4	Výběr vhodných barev	21
5.2.5	Výběr vhodných fontů a jejich vlastností	22
5.2.5.1	Velikost fontu	22
5.2.5.2	Rodiny fontů	24
5.2.5.3	Tvary fontů - kurzíva a kapitálky	26

5.2.5.4	Tloušťka fontu	27
5.3	Příkazy třídy BEAMER	27
5.3.1	Vytvoření rámců	28
5.3.1.1	Prostředí s rámci	28
5.3.1.2	Součásti rámce	36
5.3.1.2.1	Záhlaví a zápatí	36
5.3.1.2.2	Postranní panely	42
5.3.1.2.3	Navigační lišty	45
5.3.1.2.4	Navigační symboly	51
5.3.1.2.5	Logo	53
5.3.1.2.6	Titul rámce	54
5.3.1.2.7	Pozadí	56
5.3.1.3	Velikost okrajů	58
5.3.1.4	Omezení počtu snímků v rámci	59
5.3.2	Vytvoření překrytí	60
5.3.2.1	Pauzovací příkazy	60
5.3.2.2	Obecný koncept specifikací překrytí	61
5.3.2.3	Příkazy se specifikacemi překrytí	63
5.3.2.4	Prostředí se specifikacemi překrytí	68
5.3.2.5	Dynamická změna textu nebo obrázků	69
5.3.2.6	Pokročilé specifikace překrytí	72
5.3.2.6.1	Vytváření příkazů a prostředí, které reagují na specifikace překrytí	72
5.3.2.6.2	Specifikace režimů	75
5.3.2.6.3	Specifikace akcí	76
5.3.2.6.4	Přírůstkové specifikace	79
5.3.3	Struktura prezentace: statická globální struktura	81
5.3.3.1	Přidání titulu stránky	81
5.3.3.2	Přidání oddílů a pododdílů	84
5.3.3.3	Přidání částí	89
5.3.3.4	Rozdělení kurzu do přednášek	91
5.3.3.5	Přidání přehledu kapitol	92
5.3.3.6	Přidání bibliografie	96
5.3.3.7	Přidání dodatku	99
5.3.4	Strukturujeme prezentaci: Interaktivní globální struktura	100
5.3.4.1	Přidání odkazů a tlačítek	100
5.3.4.2	Pozdější opakování rámce	105
5.3.4.3	Přidání předpokládaného zvětšení	106

Literatura	111
Přílohy	113

Kapitola 1

Úvod a cíl práce

1.1 Úvod do problematiky

Můj oblíbený citát od legendárního swingového klarinetisty Bennyho Godmana praví, že "nejlepší věci v životě jsou zdarma". A ne jinak tomu je i u nabídky softwaru pro zpracování textu. Tedy abych byl přesný, pro zpracování textu k publikačním účelům.

Řeč je o sázecím programu $\text{T}_{\text{E}}\text{X}$, respektive o nejznámější sadě maker pro tento program - $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u. Historie vzniku softwaru nebývá, až na některé výjimky (Windows, Linux apod.), v pracích příliš zajímavou pasáží. Troufám si tvrdit, že vznik programu $\text{T}_{\text{E}}\text{X}$ však takovou výjimkou je.

Zasloužil se o ní zakladatel $\text{T}_{\text{E}}\text{X}$ u Donald Ervin Knuth ze Standfordské univerzity. Po svých studiích na Kalifornském technickém institutu (California Institute of Technology), kde obržel titul Ph.D., začal pracovat na své knize The Art of Computer Programming. V té době také odmítl pracovní nabídku od Národní bezpečnostní agentury (National Security Agency) a dal přednost práci právě na Standfordské univerzitě, kde pracuje do současnosti. Nutno podotknout, že práce na knize mu šla velmi dobře. A to dokonce tak, že po několika letech práce měla ručně psaná předloha okolo 3000 stran. Z toho důvodu se rozhodl knihu rozdělit.

Na konci 70. let 20. století, když měl být vydán druhý svazek jeho knihy (z plánovaných sedmi), vydavatel Knuthovi oznámil, že od vydání prvního svazku vyměnili původní sázecí systém za nový a ten že budou muset použít. Při shlednutí vzorů se svérázný Knuth zhrozil, protože nová verze byla ještě horší, než ta původní a jeho text obsahoval chyby a mnoho nedokonalostí, zejména v zápisu matematických znaků.

Protože si uvědomoval, že je nejen matematik, ale i počítačový vědec, rozhodl se, že celou situaci vyřešit následovně: nastuduje tehdejší typogra-

fická pravidla, dále pravidla pro sazbu matematických vztahů a protože fonty symbolů, které potřeboval, prakticky ještě vůbec neexistovaly, zjistí co nejvíce informací také o designu a řezu písem. Poté vědomosti zapracuje do nového formátovacího programu, který pojmenoval $\text{T}_{\text{E}}\text{X}$ (Název odvodil od starořeckého slova, které bylo výrazem pro umění či dovednost. Z důvodu použití řeckého písma proto "ortodoxní" uživatelé vyslovují [tech], jinak se také užívá z angličtiny přejatý [tek]. Zápis výchozího starořeckého slova vypadá takto: $\tau \varepsilon \chi \nu \eta$).

Knuth odhadl, že za 6 měsíců by práce měla být hotova. Dnes víme, že práce trvala přibližně 10 let a podílelo se na ní více lidí. Spolupracovníky získal rok po zahájení své práce na výročním zasedání Americké matematické společnosti (American Mathematical Society), na kterou byl pozván a kde se rozhodl tento projekt představit. Setkal se s velkým ohlasem, protože útrapy se sazbou matematických vztahů zažívali i ostatní matematikové.

V čem tedy spočívaly zásadní výhody $\text{T}_{\text{E}}\text{X}$ u: mohli ho používat sami autoři textů, tedy ti, co rozuměli tomu co píší, což se podstatně promítlo do výsledné kvality sazby. Vznikl z akademických zdrojů a měl být k dispozici zdarma. Obrovské výhody program získal během svého vývoje - stal nezávislým na všech zařízeních (operační systém, platforma, výstupní zařízení). Také se stal nezávislým na vstupních souborech. Tyto soubory byly přenositelné a obsahovaly značkovací instrukce, podle kterých byl výstup vždy stejný.

$\text{T}_{\text{E}}\text{X}$ rozumí přibližně 300 základním příkazům, tzv. primitivům, které dovolují jednoduché sázecí operace a programovací funkce. Protože příkazy jsou příliš elementární, práce v samotném $\text{T}_{\text{E}}\text{X}$ u by byla prakticky nemožná. Elementárnost příkazů je na druhou stranu výhodou, protože na jejich základě tak lze definovat komplexnější příkazy na vyšší úrovni – makra. Souhrn (balík) maker se nazývá formát.

Nejjednodušší formát pojmenovaný Plain $\text{T}_{\text{E}}\text{X}$ sestavil sám Knuth, uživatel má ale funkci spíše jako sazeč textu, než jeho autor a práce byla stále programátorsky náročná, zdlouhavá a uživatel musel vykazovat určité znalosti typografie.

To byl důvod, proč Leslie Lamport v roce 1985 na základech Plain $\text{T}_{\text{E}}\text{X}$ u sestavil formát $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. Název vznikl sloučením a zkrácením příjmení zakladatele Lamporta a názvem programu $\text{T}_{\text{E}}\text{X}$ - dohromady tedy $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. Tento americký počítačový specialista, který v současné době pracuje ve společnosti Microsoft, vytvořil sadu maker, pro vytváření komplexních dokumentů. Formát je sestaven tak, aby byl uživatelsky přívětivější a umožnil práci v programu $\text{T}_{\text{E}}\text{X}$ mnohem širšímu okruhu uživatelů. Ti, aniž by měli znalosti sazby nebo programování, mohou využívat rozsáhlých výhod, které program $\text{T}_{\text{E}}\text{X}$ nabízí a to během relativně krátkého času. $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ obsahuje třídy, tedy soubory, na který odkazuje vstupní text a v které definují volby pro specifické druhy

textů. Příkazy pro tyto texty jsou uloženy v doplňkových souborech třídy. Některá potřebná data k této kapitole jsem čerpal ze serveru *tug.org*[9] a *Podrobného průvodce*[2].

Jednu z těchto tříd systému \LaTeX , třídu BEAMER, sestavil Till Tantau a umožňuje v \LaTeX u vytvářet prezentace. Stejně jako celý \TeX funguje na principu WYSIWYM (What you see is what you mean). A právě popis této relativně málo rozšířené, ale podle mého názoru velmi užitečné třídy, je předmětem mé bakalářské práce.

1.2 Cíl práce

Prezentace jsou velmi užitečným nástrojem ať už pro předávání vědomostí a informací, či představování prací, produktů i lidí. Díky své užitečnosti se tak jejich použití rozšířilo nejen ve školách a konferencích, ale zejména i do běžného fungování firem a společností. Velké rozšíření využívání prezentací znamená i poptávku po nástrojích, jejichž pomocí lze tyto prezentace vytvářet. Stejně jako v ekonomice, i zde platí zákonitosti volného trhu a zvyšující se poptávka po softwaru měla za následek rostoucí nabídku. Proto má dnes uživatel možnost využít relativně široké nabídky programů k tomuto účelu. Většina jich je ale komerčních, což se jeví pro mne, studenta, jako zásadní problém v hlavním kritériu takového výběru. Druhým zásadním kritériem je kvalita výstupu, tedy samotné prezentace, které je program schopen dopomoci. Vzhledem k těmto kritériím a mým sympatiím k filosofii Open source a freeware licencím, jsem se během studia díky panu doktoru Pechovi seznámil s velmi užitečným systémem \LaTeX , potažmo její třídou BEAMER, která, jak jsem následně zjistil, však u nás nebyla nikým podrobněji popsána. Zde se mi otevřela možnost se o to pokusit.

Cílem práce je tedy vyčerpávajícím způsobem seznámit uživatele \LaTeX u s touto třídou, důkladně ho obeznámit s vytvářením prezentací v ní, popsat její příkazy a vlastnosti těchto příkazů a navíc vše doplnit i užitečnými obecnými radami pro tvorbu prezentací nejen v BEAMERu. Pro základní představu uvedu i kód jednoduché prezentace, která bude použita při obhajobě této práce.

Kapitola 2

Metodika řešení

Jak jsem uváděl v úvodu, velkou výhodou \LaTeX u je jeho nezávislost na platformě. Proto ho můžeme nainstalovat na téměř všechny typy počítačů. Druh a průběh instalace se však může v závislosti na operačním systému lišit. Všechny postupy jsou však již vyčerpávajícím způsobem pomocí návodů a rad zdokumentovány a nejsou příliš obtížné (což v minulosti, zejména v OS Windows, nebylo samozřejmostí).

Tuto práci jsem psal a třídu BEAMER používal na počítači s OS Linux, distribucí Ubuntu 10.10. Tuto distribuci používám prakticky z jediného důvodu: Ubuntu je velkým zastáncem používání čistě svobodného softwaru. Samozřejmě má i další výhody, např. je uživatelsky přívětivý a má dobrou technickou podporu. Používání \TeX u včetně jeho nadstaveb umožňuje distribuce \TeX Live, má současná verze je \TeX Live 2009, které je součástí i třída BEAMER. Jako editor pro práci používám \TeX maker, nyní ve verzi 2.0, který mi díky možnosti používání mnou předdefinovaných značek, alespoň částečně urychluje práci.

V úvodu své bakalářské práce jsem uživatele stručně seznámil \TeX em, s jeho zajímavou a ne příliš známou historií, vysvětlil základní pojmy a pokusil se uživateli pomoci získat určitý přehled a orientaci. Následující kapitoly budou věnovány zejména třídě BEAMER. Ne však zcela výhradně, protože mnoho rad a pravidel týkajících se tvorby prezentací lze aplikovat i v jiných systémech, pro tyto účely vyvinutých. Poté s využitím uvedených zdrojů a jejich vlastním zpracováním podrobně popíši a vysvětlím příkazy, které jsou součástí jen třídy BEAMER. Dle jejich funkce se dají rozdělit do kapitol na příkazy k vytváření a úpravy rámců (záhlaví, zápatí, navigační panely, postranní panely apod.), k vytváření a úpravy překrytí (postupné odkrývání, zvýrazňování textu apod.), příkazy upravující statickou strukturu textu (oddíly, pododdíly, části atd.) a i interaktivní globální strukturu (odkazy, talčítka apod.).

Kapitola 3

Výchozí stav problematiky

Stav, z kterého jsem na počátku vycházel, nebyl příliš jednoduchý. Během mé práce totiž nebyla ani jedna tištěná publikace, která by se o třídě BEAMER zmiňovala, natož se jí nějak zaobírala. U stručnějších publikací, jako je např. *LaTeX pro začátečníky* [1], jsem to ani nepředpokládal, ale překvapilo mne, že se žádná informace neobjevila ani v podstatně obsáhlejší knize *LaTeX Podrobný průvodce* [2]. Zřejmě v té době nebyla třída ještě příliš známá (což ostatně ještě zcela není ani dnes). Mezi elektronickými zdroji dat to nebylo o mnoho lepší. Vesměs se jednalo o krátké návody, jak v několika málo krocích vytvořit v BEAMERu jednoduchou prezentaci, např. *BEAMER: LaTeX na prezentace* [6], *Beamer - prezentace v LaTeXu* [10] nebo *Beamer - snadná a profesionální prezentace v (pdf)LaTeXu* [11]. Musím uznat, že pro moje první seznamování s touto třídou byly velmi přínosné, pro další práci však ne příliš využitelné. Zajímavé informace se vyskytly v článku na stránkách Československého sdružení uživatelů TeXu, *Prezentační software pro LaTeX* [12] ve slovenštině objevil krátký report i o BEAMERu. Nutno podotknout, že se v tomto článku stal BEAMER vítězem mezi porovnávanými nástroji.

Potřebné zdroje dat jsem objevil až při prohledávání cizojazyčných webových stránek a dokumentů. Zjistil jsem, že součástí nainstalované třídy BEAMER je i příložený dokument *The Beamer class* [8] (v angličtině), která se stala stěžejním zdrojem mých informací pro tuto práci. Také jsem se dozvěděl mnohem více detailů a zajímavostí z historie celé "rodiny" programů založených na TeXu, např. *LaTeX Podrobný průvodce* [2] nebo *The history of TeX* [9].

Kapitola 4

Pojem třída a prezentační možnosti v L^AT_EXu

4.1 Třída

Pomocí tříd se v L^AT_EXu definuje globální styl sazby pro zpracování celého dokumentu. Na základě toho, jaký typ dokumentu budete chtít vytvořit si vyberete příslušnou třídu. Základní a nejpoužívanější třídy jsou `book`, `article`, `report`, `letter` a nebo také `beamer`. Vždy můžete zvolit pro každý dokument pouze jednu třídu a podle toho, kterou zvolíte, se vám změní nejen výsledný vzhled stránek, ale také jejich rozvržení. Třída `article` umožňuje členit dokument na *části*, *sekce*, *podsekce* apod., zatímco třída `report`, s kterou mimochodem píšete i tuto práci, může mít i kapitoly. Třída `book` obsahuje také kapitoly, ale jiným způsobem zpracovává sudé a liché stránky; na každé stránce tiskne průběžné živé záhlaví s nadpisem kapitoly a sekce.

Jako úplně prvním příkazem v preambuli dokumentu se v L^AT_EXu píše právě příkaz určující třídu. Do verze L^AT_EXu 2.09 (včetně) vypadal zápis kódu takto: `\documentstyle[volby]{jménotřídy}`. Správná syntaxe v současné verzi, tedy v L^AT_EXu 2 ϵ vypadá následovně: `\documentclass[volby]{třída}`. I v nové verzi je však původní příkaz zařazen a to pouze kvůli zajištění kompatibility se staršími dokumenty[2].

4.2 Možnosti prezentací pomocí L^AT_EXu

Pro tvorbu prezentací existuje v L^AT_EXu hned několik tříd. Liší se nejen svým stylem zpracování dokumentu pro specifické účely, ale také určitou mírou "praktičnosti" pro použití uživatelem. Některé z nich se vám nyní pokusím představit:

4.2.1 Třída slides

Třída slides je obsažena v instalaci \LaTeX 2 ϵ , ve starší verzi (2.09) nesla název Sli \TeX . Nabízí pouze základní sadu příkazů pro tvorbu a práci se snímky. Tato třída je určena speciálně pro výrobu transparentních fólií pro zpětný projektor, na formát Letter (tzn. formát označovaný jako "A4").

Balík obsahuje sice jedno užitečné prostředí, to se dá ale najít v celé řadě dalších balíčků. Postředí se jmenuje `overlays` a funguje s kombinací příkazů `\visible` a `\invisible`. Jejich pomocí se dá upravovat, na kterém slidu bude položka viditelná a na kterém ne. Příprava snímků je však v balíku slides trochu zbytečně zdlouhavá: je třeba vytvořit celý slide, ten pak nakopírovat tolikrát, kolik chceme mít snímků (je třeba dbát na jejich rozumný počet, protože každá vrstva totiž snižuje propustnost světla skrz fólii a tím i čitelnost snímku). Pokročilejší balíčky ale přichází s efektivnějším řešením a proto se tato třída příliš nepoužívá, ale pro úplnost ji zde uvádím.

4.2.2 Třída foiltex

Tento balík je velmi podobný třídě `slides`. Třída `foils` poskytuje množství volitelných přepínačů, např. na změnu orientace papíru, nebo na ovlivnění velikosti či poměru stran. Snímky se na rozdíl od třídy `slides` nekládají do prostředí, ale oddělují se pomocí příkazu `\foilhead{nadpis snímku}`

Příkaz `\MyLogo{}` vkládá do zápatí každého snímku uživatelem zvolené logo. Dále tento balíček nabízí příkazy k ovlivnění obsahu v rozích prezentace (levé a pravé záhlaví a zápatí). Narozdíl od třídy `slides`, `foiltex` obsahuje prostředí na vysazování matematických vět, důkazů a definic. Na druhou stranu ale této třídě chybí příkazy na práci s překrytím, což dovoluje praktické použití pouze k tvorbě průhledných fólií.

4.2.3 Třída prosper

Třída Prosper vznikla původně za účelem možnosti přípravy transparentních fólií. Postupně se ale zdokonalovala a proto není problém vytvořit pomocí této třídy prezentaci pro digitální projektor, obohatit ji o multi-mediální prvky, jako je video, postupné odkrývání, animace či zvuk. Balík `prosper` vznikl na základech třídy `seminar` od Timothy Van Zandtha. Výstupy tohoto balíčku již velmi připomínají prezentace v PowerPointu, tzn. odrážky jsou tvořeny grafickými symboly, mezi snímky je možné vložit přechodové efekty, k dispozici je sada předdefinovaných šablon, je možné vložit např. video ve formátu MPEG a pod.

4.2.4 Třída `texpower`

Překrytí u této třídy jsou řešeny komplikovaněji, než u předchozích balíčků. Umožňují tak uživateli větší kontrolu (např. zobrazování vzorce po částech). Zajímavostí je možnost vložit do prezentace navigační tlačítka, pomocí kterých lze přepínat mezi snímky či překrývat. K dispozici jsou také příkazy pro přidávání přechodových efektů mezi snímky nebo automatický posun.

Zřejmě nejdokonalejší třída je oproti ostatním třída `BEAMER`. Umožňuje využít mnoho vlastních šablon, funguje v ní naprostá většina standardních příkazů \LaTeX u a dovoluje snadné používání funkce překrytí.
du tříd jsem čerpal ze *Zpravodaje serveru CSTUG.cz*[12]

Kapitola 5

Třída BEAMER

5.1 Instalace

Samotná instalace třídy BEAMER je velmi jednoduchá. Není v praxi totiž vůbec potřeba. Distribuce T_EXu, potažmo L^AT_EXu pro Windows MiK_TE_X a T_EXlive v sobě mají BEAMER nainstalovaný, stejně tak v případě distribucí T_EXLive a t_EXpro operační systém Linux a většiny jeho distribucí.

5.2 Rady a pravidla pro tvorbu prezentace

Tato část obsahuje shrnutí různých rad a pravidel, při jejichž používání si může uživatel celou tvorbu prezentace jednak zpřehlednit, ale zejména výslednou práci zkvalitnit (vhodný počet stran, správná struktura, promyšlené použití barev, fontů, obrázků, animací atd.). Kvalitně sepsaná práce poté samotné prezentování značně ulehčí a promítne se na celkovém výsledku naší snahy publiku, sdělit nějakou informaci či něco vysvětlit a nemusí se jednat pouze o prezentace vytvořené v BEAMERu. Použití těchto rad pochopitelně není povinné a i bez jejich znalosti lze vytvořit kvalitní prezentaci, k mnohým radám uživatel totiž sám nevědomky dospěje za použití svého "selského rozumu". Níže uvedené rady jsou částečně sestaveny jednak z vlastních zkušeností, které jsem nabyl během tvorby prezentací, ale zejména volným překladem a zpracováním dokumentu přiloženého ke třídě BEAMER[8].

5.2.1 Struktura prezentace

5.2.1.1 Časové rozpětí

Jakmile se pustíte do tvorby určité prezentace, první věcí, již byste měli zvážit, je čas, který pro svou prezentaci budete mít k dispozici. V závislosti na konkrétní situaci to mohou být 2 minuty nebo také 2 hodiny.

- Jednoduchým pravidlem pro určení počtu jednotlivých snímků je, že během jedné minuty byste měli prezentovat maximálně jeden snímek.
- Většinou budete mít na prezentaci méně času, než kolik by vám vyhovovalo.
- Nesnažte se do prezentace vměstnat více informací, než kolik dovoluje přidělený čas. Ačkoliv vám některé detaily mohou osobně připadat velmi důležité, je lepší je vynechat a řádně představit hlavní myšlenku. Jinak se může stát, že nestihnete odprezentovat ani hlavní myšlenku, ani detaily.

V mnohých situacích rychle odhadnete, že na prezentaci určitých podrobností již nezbude čas. To vám může ušetřit spoustu námahy – nebude totiž nutné připravovat snímky, jež byste během prezentace stejně vynechali.

5.2.1.2 Globální struktura

Chcete-li vytvořit globální strukturu určité prezentace s ohledem na časové omezení, můžete postupovat takto:

- Vytvořte si inventář bodů prezentace, o nichž můžete během přiděleného času v rozumné míře promluvit.
- Inventář následně rozdělte podle oddílů a pododdílů.
- V případě velmi dlouhých prezentací (například přednáška na devadesát minut) můžete příkazem `\part` svoji řeč rozdělit do nezávislých celků (například „shrnutí předchozí přednášky“ či „hlavní část“).
- Během pozdější práce na prezentaci se nebojte její strukturu změnit.

Části, oddíly a pododdíly.

- V jednotlivých částech nepoužívejte více než čtyři a ne méně než dva oddíly. Dokonce i čtyři oddíly bývají obvykle příliš mnoho, nejsou-li uspořádány do velmi jednoduché struktury. Pět a více oddílů si již

posluchači nejspíše nezapamatuji. Při úvodním představení přehledu celé prezentace nejsou konec konců posluchači ještě schopni pochopit důležitost a relevanci jednotlivých oddílů a než se dostanete k jejich podrobné prezentaci, nejspíše již jednotlivé body zapomenou.

- V ideálním případě by měl být přehled obsahu prezentace jasně srozumitelný již podle názvů jednotlivých oddílů. Obzvláště by měl být srozumitelný ještě předtím, než vůbec začnete mluvit.
- Názvy oddílů a pododdílů by měly být jasně srozumitelné.
- Oddíly a pododdíly by měly tvořit logickou strukturu.
- Začněte vysvětlením, o čem vaše prezentace vlastně pojednává. Nepředpokládejte, že to každý bude vědět.
- Následně vysvětlete, na co jste v souvislosti s prezentovanými skutečnostmi přišli vy, nebo ostatní lidé.
- Svoji řeč vždy zakončete souhrnem, v němž krátce a jednoduše zopakujete hlavní body prezentace. Lidé věnují největší pozornost prezentacím na jejich začátku a pak až na konci. Souhrn tak představuje vaši „druhou šanci“, jak předat určitou myšlenku.
- Příkazem `\appendix` můžete přidat i dodatek. Do dodatku vložte vše, co v podstatě ani neplánujete prezentovat, avšak mohlo by se to hodit při zodpovídání dotazů.
- Nepoužívejte podpododdíly, komplikovanost a spleť struktura prezentaci vždy uškodí.

Abstrakty

Ve vědeckých pracích udává abstrakt v přibližně 100 slovech krátký souhrn celého dokumentu. Na základě tohoto souhrnu se čtenáři mohou rozhodnout, zda si přečtou celou práci, anebo ne.

- Jelikož se nemusíte obávat toho, že by publikum po prvním snímku odešlo, nebývá většinou potřeba, uvádět před samotnou prezentací její abstrakt.
- Chcete-li však v několika slovech uvést velmi přesně popis přednášky, můžete v tom případě zahrnout i abstrakt.
- Zahrnete-li abstrakt, neměl by být jeho text příliš dlouhý – spíše by mělo jít o velmi stručný souhrn hlavní myšlenky.

Číslování matematických vět a definic

Obvyklým způsobem globálního členění (matematických) článků a knih je postupné uvádění očíslovaných definic a matematických vět. V případě prezentací je však situace bohužel poněkud komplikovanější a užívání očíslovaných matematických vět se v tomto případě nedoporučuje, protože publikum nemá šanci si tato čísla zapamatovat. Nikdy neříkejte věty typu „Podle teorému 2.5, který jsem vysvětlil dříve, dostaneme...“. V tomto případě by bylo lepší se na matematickou větu odkázat jako na Kummerův teorém než jako na teorém 2.5. Je-li teorém 2.5 jakousi obskurní matematickou větou, jež nenes vlastní název (na rozdíl od Kummerova teorému, Hlavního teorému apod.), publikum ji zapomene dřív, než se na ni opětovně odkážete.

Jediným případem, kdy mají číslované matematické věty v prezentacích smysl, by mohla být přednáška, při níž si studenti mohou zároveň číst podklady, v nichž jsou jednotlivé teorémy očíslovány stejně jako v prezentaci. Pokud teorémy a definice očíslovujete, číslování je v určitém pořadí a to tak, že při přiřazování čísel neberte ohled na to, zda-li se jedná o teorém či definici a číslování informací tak, jak jdou za sebou. Jeden příklad vše jasně vysvětlí: Pokud máte jeden teorém a dvě definice, nebudete číslovat teorém 1, definice 1, definice 2, ale teorém 1, definice 2, definice 3. Tento systém poslouží pro podstatně snazší orientaci posluchačů v prezentaci, protože jedna informace má právě jedno číslo nezaměnitelné s jiným.

Bibliografie

Na konci své prezentace byste třeba rádi uvedli i nějakou bibliografii tak, aby ostatní viděli, co je možné si k tématu přečíst. Přidáváte-li do prezentace bibliografii, některé následující rady by Vám mohli pomoci:

- Není dobré uvádět v prezentaci dlouhou bibliografii. Uveďte jen několik referencí. (To platí samozřejmě pouze pro prezentaci. Pokud publiku rozdáváte nějaké podklady, není třeba být tolik stručný.)
- Neuvádějte více referencí, než se vejde na jeden snímek, protože si nikdo více referencí většinou nezapamatuje.
- Reference uvádějte pouze v případě, že jsou zamýšleny jako materiál k dalšímu studiu dané problematiky. Neuvádějte seznam všech použitých zdrojů – to se hodí jen do písemného dokumentu.
- Používání příkazu `\cite` není v prezentaci ideální. Posluchači s jen stěží zapamatují citace tak, aby mohli na konci prezentace identifikovat jejího autora podle čísla odkazu. Vhodnější je jistě citovat zdroj ihned za danou informaci v částečně zkrácené podobě např. [autor, rok].

5.2.1.3 Struktura rámce

Stejně tak jako celá prezentace, i jednotlivé rámce by měly mít určitou strukturu. Vyplníte-li určitý rámec pouze textem, bude obtížné se v něm orientovat. Je vaším úkolem strukturovat obsah jednotlivých rámců tak, aby publiku v ideálním případě bylo jasné, které informace jsou důležité, co jsou jen detaily a jak spolu informace souvisí.

Titulek rámce

- Do každého rámce vložte titulek. Titulek vysvětluje obsah daného rámce těm účastníkům přednášky, kteří nevěnovali pozornost všem podrobnostem na snímku.
- Titulek by měl problematiku vysvětlovat – neměl by to být žádný tajemný souhrn informací, jemuž nelze porozumět, dokud si nepročtete celý snímek.
- V ideálním případě by tituly následných snímků mohly vysvětlovat problematiku samy o sobě.

Množství informací v rámci

- Rámec s menším objemem informací je lepší než rámec, v němž je informací příliš mnoho. Obvykle by měl mít rámec mezi 20 až 40 slovy. Maximum by mělo být asi 80 slov.
- Nepředpokládejte, že v publiku jsou všichni experty na dané téma. I v případě, že naslouchající vaší prezentace jsou odborníci, je možné, že o daném tématu slyšeli již před lety. I když vám osobně informace připadají jasné, je třeba vždy zvážit, zda-li je není třeba alespoň stručně vysvětlit.
- Do snímku nikdy nevkládejte informace, jež nehodláte během prezentace vysvětlovat. Nezkoušejte na nikoho udělat dojem komplikovaností daného tématu. Naopak je běžné, vysvětlit i skutečnosti, které na snímku nejsou.
- V jednoduchosti je síla. Obvykle má publikum na přečtení jednoho snímku méně než 50 sekund. Nebudou mít čas se probírat dlouhými větami či komplikovanými vzorci.

Struktura rámce

- Používejte bloková prostředí, například `block`, `theorem`, `proof`, `example` atd.
- Spíše než prostý text upřednostňujte výčty a rozepisování do jednotlivých položek.
- Definujete-li několik skutečností, použijte příkaz `description`.
- Ve vytvářené struktuře je důležité zdůraznit vhodné pasáže. Příkazem `\alert` zdůrazněte důležité skutečnosti. Může to být jediné slovo či celá věta. Zvýrazňování však nepoužívejte přespříliš, jelikož by bylo kontraproduktivní.
- Používejte sloupce.
- Nikdy nepoužívejte zápatí. Narušuje tok textu, aniž by mělo nějaký valný význam. Zápatí buď obsahuje důležité informace, které byste měli umístit do běžného textu, nebo žádné důležité informace neobsahuje, a měli byste je proto vynechat (obzvláště v prezentacích).
- K vysázení citací používejte příkazy `quote` či `quotation`
- S výjimkou dlouhých bibliografií nepoužívejte volbu `allowframebreaks`.
- Nepoužívejte dlouhé bibliografie.

Zápis textu

- Používejte krátké věty.
- Upřednostňujte fráze před úplnými větami. Například místo věty „Obrázek nahoře zobrazuje zadání, obrázek vpravo ukazuje správný výsledek.“, zkuste „Vlevo: zadání. Vpravo: správný výsledek.“ Nebo ještě lépe – запиšte to jako výčet nebo popis.
- Používejte správnou interpunkci: žádné tečky na konci frází, úplná interpunkce v případě ucelených vět.
- Nikdy nepoužívejte menší velikost písma z důvodu, že byste chtěli do rámce vložit více textu.
- Snažte se nepoužívat dělení slov. Je-li to opravdu nezbytné, rozdělte slova „ručně“ příkazem `\-`.
- Zalamujte řádky „ručně“ pomocí příkazu `\\`. Nespoléhejte na automatické zalamování řádků.

- Text a čísla v obrázcích by měly mít stejnou velikost jako běžný text, zabráníte tak případné nečitelnosti.

5.2.1.4 Interaktivní prvky

V ideálním případě se snažíte prezentovat snímky v lineárním pořadí, nejspíše stisknutím klávesy PgDn u každého snímku. Z různých důvodů (příprava na dotazy publika apod.) však takovou lineární posloupnost můžete narušit:

- Ve třídě BEAMER můžete použít příkaz `\againframe` pro pokračování rámců, které jste v určitém místě zahájili, avšak určité podrobnosti jste v nich potlačili. Příkaz `\againframe` můžete použít až mnohem později, například pouze v dodatku, v němž zobrazíte dodatečné snímky.
- Ve třídě BEAMER můžete použít příkaz `\framezoom`, jímž vytvoříte odkazy na zmenšené části komplikovaného snímku.

5.2.2 Použití grafiky

Známé přísloví říká, že obrázek je lepší než tisíc slov. Ne jinak tomu je i v prezentacích.

- Kdykoliv to bude možné, umístěte do každého snímku alespoň jeden obrázek. Vizualizace mohou publiku velmi pomoci.
- Obvykle umísťujte grafiku vlevo od textu. (Použijte prostředí `columns`.) Přirozeně si totiž nejdříve prohlédneme snímky zleva (stejně, jako když čteme)
- Grafika by měla mít stejné typografické parametry jako text: V obrázcích používejte stejné fonty (a stejnou velikost) jako v hlavním textu. Tečka v obrázku by měla mít úplně stejnou velikost jako tečka v textu.
- Zatímco bitmapová grafika, například fotografie, může ve srovnání se zbytkem textu hýřit barvami, vektorové obrázky by se měly řídit stejnou logikou barev, jako hlavní text (například černá = běžné čáry, červená = zvýrazněné části, zelená = příklady).
- Stejně tak jako v případě textu byste měli vysvětlit i obsah jednotlivých obrázků. Pokud určité podrobnosti nevysvětlíte, publikum může být poněkud zmateno, protože neví, jestli jim neuniklo něco důležitého. Při importu obrázků z papíru či jiných zdrojů postupujte opatrně. Většinou

se v nich totiž nachází mnohem více detailů, než kolik budete moci vysvětlit, a měli byste je proto částečně zjednodušit.

- V některých případech je však grafika úmyslně velmi komplikovaná, protože chcete obrázkům věnovat spoustu času a jejich jednotlivé podrobnosti obsírně vysvětlit. Často se však budete potýkat s tím, že jemné detaily obrázku nelze před publikem zobrazit v dostatečném rozlišení. V tom případě byste měli použít například příkaz `\framezoom`, jímž u těchto obrázků vytvoříte zvětšené výřezy potřebných částí.

5.2.3 Výběr vhodných témat

Třída BEAMER se dodává s řadou různých témat. Pro výběr té vhodné se můžou hodit následující rady:

- Různá témata se hodí pro různé situace. Není vhodné zůstat u jednoho oblíbeného tématu – vyberte si téma, které se pro danou situaci hodí.
- Navigační ovládací prvky patrně využijete spíše v delší přednášce než v krátké prezentaci. Presentujete-li studentům 90minutovou přednášku, měli byste zvolit téma, které vždy zobrazí postranní panel se zvýrazněným aktuálním předmětem diskuze. Tak bude každý posluchač vědět, v jakém „stavu“ se vaše přednáška právě nachází. Přednášíte-li pouze desetiminutovou úvodní řeč, přehled kapitol bude vypadat poněkud hloupě.
- Téma, v němž se zobrazuje jméno autora a jeho členství v určité organizaci, se hodí v situacích, kdy vás posluchači pravděpodobně neznají. Pokud vás každý zná, je uvádění vlastního jména na každém snímku známkou domýšlivosti.
- Nejdříve vyberte téma prezentace, jež svým rozvržením odpovídá námětu vaší přednášky.
- Poté můžete instalaci jiného barevného tématu případně změnit barvy. To může výrazně změnit vzhled celé prezentace.

5.2.4 Výběr vhodných barev

- Barvy používejte s mírou. Připravená témata jsou již sama o sobě docela barevná. Chcete-li přidat další barvy, například pro počítačový kód či matematický text atd., ujistěte se, že je to opravdu nutné.

- Jasně barvy na bílém pozadí používejte opatrně, obzvláště pak v případě zelené. Vypadá-li určitá barevná kombinace na monitoru skvěle, může stejná kombinace působit nevzhledně během prezentace, což je způsobeno tím, že monitory, projektory a tiskárny zobrazují barvy často trochu odlišně. Používáte-li jasné barvy na jasném pozadí, bude nejspíš třeba přidat i mnoho černé.
- Maximalizujte míru kontrastu. Běžný text by měl být zobrazen černě na bílém pozadí, nebo alespoň velmi tmavou barvou na velmi jasném pozadí.
- Stínované pozadí snižuje čitelnost a nikterak nezvyšuje informační hodnotu.
- Inverzní zobrazení (jasný text na tmavém pozadí) může během prezentací v jasných prostředích působit potíže, protože jen malou část prostoru s prezentací osvětluje projektor. Inverzní zobrazení se obtížně rozšiřuje do tištěných materiálů a na průhledné fólie.

5.2.5 Výběr vhodných fontů a jejich vlastností

Výběr správného fontu pro danou prezentaci není vůbec triviálním úkolem. Vyberete-li si nesprávně, nebude v lepším případě prezentace vizuálně dobře vypadat, v horším případě bude mít publikum potíže se čtením snímků. Tato kapitola tedy obsahuje několik doporučení, jež vám pomohou s výběrem fontů, které při prezentaci prostřednictvím projektoru vypadají dobře. Font má různé vlastnosti (atributy), třeba tloušťku, rodinu či velikost. Všechny tyto vlastnosti mají vliv na použitelnost fontů během prezentace. V několika následujících odstavcích jsem se pokusil přehledně shrnout tyto atributy a naznačit jejich výhody i nevýhody.

5.2.5.1 Velikost fontu

Pravděpodobně nejnápadnějším atributem fontu je jeho velikost. Fonty se tradičně měří v tzv. bodech. Velikost tohoto bodu však není pro všechny systémy stejná. Systém $\text{T}_{\text{E}}\text{X}$. zastává názor, že bod je $1/72,27$ palce. Jeden palec měří 2,54 cm. Na druhou stranu PostScript a Adobe zastávají názor, že bod je $1/72$ palce (v systému $\text{T}_{\text{E}}\text{X}$. se tak označuje tzv. velký bod). Existují i rozdíly mezi americkým a evropským bodem. Pokud je určeno, že text má velikost 11 pt (bodů), popisuje výšku písmen daného fontu, tj. 11 pt. Takové označení výšky však pochází z minulosti a popisuje tak vertikální velikost odlitých olovených písmen. Nemusí se tedy jednat o souvislost například s

vlastní výškou písmene x nebo dokonce M. Písmenko x vysázené fontem Times od společnosti Adobe o velikosti 11 pt bude mít určitou výšku, která se liší od výšky písmene x o velikosti 11 pt ve fontu Times UTC a písmene x o velikosti 11 pt ve fontu Helvetica od Adobe.

Zkráceně tedy: velikost fontu příliš nesouvisí s velikostí vlastních písmen. V současné době je spíše nepsaným pravidlem, že font o velikosti 10 nebo 11 pt by měl být po vtištění určen k běžnému čtení.

Během prezentace pochopitelně ztrácejí klasické velikosti fontů svůj smysl. Nikdo by nebyl schopen přečíst promítaný text, který by měl opravdu velikost 11 pt. Místo toho musí být promítaná písmenka několik centimetrů vysoká. Nemá tedy příliš smysl zadávat při tvorbě prezentace velikost fontů obvyklým způsobem. Lépe určují je tak počet řádků, které se vměstnají na snímek, pokud byste jej teoreticky vyplnili odshora až dolů textem. V závislosti na vzdálenosti publika od promítací plochy a její velikosti by se na každý snímek mělo vejít optimálně něco mezi 10 a 20 řádky textu. Pochopitelně čím méně řádků, tím lepší čitelnost textu.

Ve třídě BEAMER je výchozí velikost fontů zvolena tak, že je obtížné vměstnat na snímek příliš mnoho textu. Díky této velikosti také budete mít jistotu, že publikum snímky přečte i za nepříznivých podmínek, třeba ve velké místnosti s malou projekční plochou. Fonty však můžete poněkud zvětšit či zmenšit, pokud víte, že v daném prezentačním prostředí bude vámi zvolená velikost vhodnější.

Jakmile nastavíte velikost běžného textu, další rozměry se obvykle definují v poměru k vámi nastavené velikosti. Z tohoto důvodu najdete v systému L^AT_EX příkazy jako `\large` či `\small`. Vlastní velikost, kterou nastaví tyto příkazy, závisí na velikosti běžného textu.

V prezentacích používejte velmi malé fonty na text v záhlaví, zápatí či v postranních panelech, protože zde zobrazený text není důležitý. Takový text by však samozřejmě měl být dostatečně velký, aby jej bylo možné přečíst. Text specifikovaný příkazem `\tiny`, by měl pro tento účel vyhovovat.

Na druhou stranu však používání malých fontů může být záludné. Používáte-li fonty PostScript v malých velikostech, jednoduše se tyto fonty zmenší. Použijete-li určitý font v menší než běžné velikosti, jednotlivá písmenka by se měla vykreslit poněkud tenčeji než ta, jež jsou výsledkem pouhého zmenšení. Z tohoto důvodu se pro malá a běžná písmenka používají vysoce kvalitní fonty Multiple Master Fonts nebo Computer Modern v několika druzích. Můžete však použít i běžný font Helvetica či Times, u nichž se jednotlivá písmenka jednoduše zmenší. Podobný problém se objeví, použijete-li světlý font na tmavém pozadí. I když jej vytisknete na papír ve vysokém rozlišení, světlý text tihne k tomu, že se poněkud ztratí v tmavém pozadí. Promítnete-li světlý text na tmavém pozadí během prezentace, může se tento efekt projevit

ještě výrazněji, a text nebude téměř k přečtení. Tomuto negativnímu efektu můžete v případě malého textu zamezit použitím tučného řezu písma.

5.2.5.2 Rodiny fontů

Další ústřední vlastností každého fontu je jeho rodina. Příkladem rodin jsou **Times**, **Helvetica** či **Futura**. Jak již napovídá jejich název, spousta různých fontů může náležet do stejné rodiny. Například rodina Times se dodává v různých velikostech, existuje tučná verze fontu Times, kurzíva atd.

Existují dvě velké třídy rodin fontů: fonty **serif** (patkové) a fonty **sans-serif** (bezpatkové). Font **sans-serif** je fontem, v němž písmena nemají patku (název pochází z francouzského slova „*sans*“, tj. „*bez*“). Patky jsou malé výstupky na koncích tahů jednotlivých čar, z nichž se dané písmenko skládá. Bezpatkové fonty se (obecně) v prezentacích snáze čtou. V nízkém rozlišení snižují patky čitelnost fontu. Na druhou stranu je patkový text stejně dobře čitelný jako bezpatkový text, promítnete-li jej na projektoru o vysokém rozlišení. Prezentace vysázená patkovým fontem působí poněkud konzervativněji, naopak bezpatkové písmo je dle mého názoru v současnosti moderní.

Výchozím fontem v systému T_EX (a třídě BEAMER) je font **Computer Modern**. Jedná se o původní rodinu fontů navrženou samotným Donaldem Knuthem pro systém T_EX. Je to vyspělý font, který se dodává se vším potřebným: rozsáhlé matematické abecedy, obrysové verze PostScript, opravdová čísla ve starém stylu (*oldstyle*), speciálně navržená malá a velká písmena, atd.

Existují však určité důvody, proč používat jiné rodiny fontů než **Computer Modern**:

- Font se může stát „okoukaným“.
- Jiné fonty, zvláště pak Times a Helvetica, se občas vykreslí lépe, protože mají lepší vnitřní zeštíhlení.
- Bezpatková verze fontu **Computer Modern** není tak dobře navržena jako jeho patková verze.
- Font **Computer Modern** potřebuje mnohem více místa než úspornější fonty, jako je třeba **Times**. Zajímavostí je, že font **Times** byl speciálně navržen tak, aby byl úsporný (novinová společnost, která vydává deník The Times, potřebovala robustní, avšak úsporný font).

Alternativy k fontu **Computer Modern**:

- Jako alternativa se často užívá font **Helvetica**. **Helvetica** však vypadá poněkud nudně (protože ji vidíme všude kolem) a má velkou výšku písmene x (výška písmene x ve srovnání s písmenem, jako je například M). Velká výška písmene x je obvykle považována za výhodnou v jazycích (například angličtina), jež používají velká písmena jen zřídka. V jazycích se spoustou velkých písmen (například němčina) se považuje za méně vhodnou.
- Font **Futura** se velmi dobře hodí pro prezentace. Má tlustá písmena, a je proto odolný vůči zmenšování a zvětšování, inverzi a nízkému kontrastu.
- Font **Times** je možnou alternativou fontu **Computer Modern**. Jeho hlavní nevýhoda spočívá v tom, že se jedná o patkové písmo, a je proto nutné používat projektor s vysokým rozlišením. Samozřejmě se také často používá, takže jej všichni velmi dobře znají.

Rodiny, jež byste neměli používat pro běžný text:

- Všechny fonty se stejnou roztečí (například **Courier**).
- Rukopisné fonty (vypadají jako psané rukou). Jejich šířka tahu je příliš tenká pro účely prezentace.
- Jemnější patkové fonty, jako například Stempel
- Fonty **Gothic**. Jen několik málo posluchačů bude schopných tyto fonty plynule číst.

Jedním z nejdůležitějších typografických pravidel je, že v textu byste měli používat co nejméně různých fontů. Obzvláště pak typografická moudrost velí, že na jedné stránce byste neměli použít více než dvě různé rodiny fontů. Sázíte-li však matematický text, je často nezbytné a užitečné použít různé rodiny fontů.

Často se používají bezpatkové fonty pro nadpisy a patkové fonty pro běžný text (např. v časopisech). Můžete použít také dva rozdílné bezpatkové fonty nebo dva rozdílné patkové fonty, avšak v tom případě se ujistěte, že jsou fonty navzájem dostatečně odlišné. Liší-li se jen málo, stránka bude vypadat podivně, aniž by publikum bylo schopno říci, proč tomu tak je. Fonty **Arial** a **Helvetica** jsou si velmi podobné a proto se jejich kombinace k sobě příliš nehodí. Oproti tomu fonty **Futura** a **Optima** jsou odlišné a lze s nimi text dobře členit.

5.2.5.3 Tvary fontů - kurzíva a kapitálky

Systém L^AT_EX má svůj koncept tvaru fontů. Jedinými opravdu důležitými tvary jsou kurzíva a kapitálky. Kurzívou se rozumí font, jehož písmena jsou *poněkud nakloněna vpravo*. Co byste měli o kurzívě vědět:

- Kurzíva se obvykle používá pro vyjádření důrazu. Obzvláště v případě bezpatkových fontů však není kurzíva dostatečně výrazná, a důraz se tak v prezentaci ztratí. Chcete-li v prezentaci vytvořit důraz, zdá se být vhodnějším řešením, použít odlišnou barvu nebo tučný text.
- Podíváte-li se pozorně, všimnete si, že text psaný kurzívou je nejenže nakloněný, ale v podstatě používá i jiná písmena (srovnejte například písmeno *a* s tímto písmenem *a*). To však platí pouze v případě patkového textu, nikoliv bezpatkového. Text, který je pouze nakloněný, avšak nepoužívá jiná písmena, se nazývá skloněný (nakloněný), nikoliv kurzíva. Někteří typografové se údajně na skloněný patkový text dívají s nevolí a považují jej za podřadnou typografii.

Hodláte-li se v některé části prezentace užít patkový font, měli byste používat také kurzívu, nikoliv skloněný text.

- Odlišná písmena používaná v patkové kurzívě se v porovnání s originálními ručně psanými písmeny, z nichž původně vycházejí, změnila mnohem méně než běžný patkový text. Z tohoto důvodu vytváří patková kurzíva dojem rukou psaného textu, což se může hodit, chcete-li, aby prezentace působila poněkud osobněji. Ve srovnání s běžným textem je však obtížněji čitelná, a nepoužívejte ji proto v textech delších než jeden řádek.

Druhým tvarem fontu, jenž je podporován systémem T_EX, jsou kapitálky. Jejich užíváním můžete dosáhnout konzervativního, ba i formálního dojmu, avšak používejte je s rozvahou:

- Kapitálky se liší od textu napsaného pouze velkými písmeny. Text vysázený kapitálkami ponechává běžná velká písmena beze změny a pro vysázení malých písmen používá menší verze velkých písmen. Slovo Česko je kapitálkami vysázeno jako ČESKO, avšak jako ČESKO velkými písmeny.
- Text vysázený kapitálkami se ve srovnání s běžným textem obtížněji čte. Důvodem je to, že čteme zejména pomocí rozeznávání tvarů slov. Například slovo „kapitálka“ většinou rozpoznáme podle počátečního vysokého písmene „k“, dále pak podle normálního písmene „a“, písmene

s nožkou zasahující pod linku „p“ atd. Je mnohem obtížnější zpozorovat překlep ve slově „kapitélka“ nežli v „kapirélka“. Kapitálky naruší tvar slov, jelikož „KAPITÁLKA“, „KAPITÉLKA“ i „KAPIRÉLKA“ budou mít vždy stejný tvar, a proto bude obtížné je rozlišit.

5.2.5.4 Tloušťka fontu

Tloušťkou fontů se myslí tloušťka písmen. Obvykle se fonty v systémech dodávají jako standardní a jako tučné. Často jsou k dispozici i polotučné (semibold), ultratučné (ultrabold nebo black), tenké (thin) či ultratenké (ultrathin) verze. Na snímku prezentace se obvykle nachází málo textu a také další prvky, jež se snaží přitáhnout pozornost publika. Použijete-li pro vyjádření důrazu tradiční kurzívu, publikum ji často přehlédne. Používání tučného textu se tak v prezentacích zdá být dobrou alternativou. Ještě lepší alternativou je však používat jasné barvy, například **červenou**, jež přitahují pozornost.

5.3 Příkazy třídy BEAMER

Příkazů, které lze uplatnit při používání třídy BEAMER, je velmi mnoho a to proto, že při tvorbě můžeme samozřejmě využívat ostatních příkazů systému L^AT_EX (pro vkládání obrázků, animací atd.). Tato kapitola však obsahuje vysvětlení většiny příkazů, které jsou specifické pouze pro třídu BEAMER, tzn. jsou v balíku příkazů nesoucí jméno BEAMER.

V první části jsou sepsány příkazy a jejich vlastnosti, sloužící k vytvoření rámců, tedy základních kamenů každé prezentace. Názorné příklady jsou vždy uvedeny ukázkovými kódy, na kterých je po vložení do L^AT_EXového editoru snadné vidět, jak příkazy fungují.

Ve druhé části je vysvětleno vytváření překrytí a příkazů, které k tomuto účelu slouží. Opět je teoretické vysvětlení doplněno názornými příklady, v této problematice obzvlášť užitečnými, ne-li dokonce nutnými.

V závěrečné části této obsáhlé kapitoly jsou popsány příkazy, které jsou určené ke strukturování prezentace. Při práci jsem volným překladem čerpal z jediného existujícího zdroje, který podrobnější informace o příkazech BEAMERu poskytuje a to dokumentu v angličtině od Tilla Tantaua[8].

5.3.1 Vytvoření rámců

5.3.1.1 Prostředí s rámcí

Prezentace sestává z posloupnosti rámců. Každý rámeček je tvořen sérií snímků. Rámeček vytvoříte příkazem `\begin{\frame}` nebo pro celé prostředí příkazem `\begin{frame}`, který má stejnou funkci. Příkaz očekává jeden parametr, kterým je obsah daného rámečku. Veškerý text, který nemá specifikaci překrytí, se zobrazuje na všech snímcích daného rámečku. Obsahuje-li určitý rámeček příkazy, které mají specifikaci překrytí, budeme mít v rámci několik snímků. V opačném případě takový rámeček obsahuje pouze jeden snímek.

```
\begin{frame}<"overlay specification">
  [<"default overlay specification">]
  ["options"] {"title"} {"subtitle"}
  "environment contents"
\end{frame}
```

Specifikace překrytí (`overlay specification`) určuje snímky, které se v daném rámci zobrazí. Jestliže specifikace vynecháte, počet se vypočítá automaticky. Obsah prostředí (`environment contents`) je běžným textem, který však nesmí obsahovat příkazy `\verb` nebo prostředí `verbatim`. Pochopitelně nelze použít ani jiné prostředí, které by měnilo znakové kódy. K tomu účelu ale slouží volba `fragile`.

Volitelný parametr (`title`) se detekuje podle úvodní závorky, tj. je-li prvním výrazem daného rámečku úvodní závorka, předpokládá se, že následuje název rámečku. Podobným způsobem se detekuje parametr (`subtitle`), tj. podle úvodní závorky následující po parametru (`title`). Titul a podtitul můžete zadat i pomocí příkazů `\frametitle` a `\framesubtitle`.

Běžný příkaz `\frame` systému L^AT_EX je k dispozici uvnitř rámců a má svou obvyklou funkci. Uvnitř i mimo rámeček je vždy k dispozici ve formě `\framelatex`.

Příklad:

```
\begin{frame}{Titul}
  Nějaký obsah.
\end{frame}
% Stejný efekt:
\begin{frame}
\frametitle{Titul}
  Nějaký obsah.
\end{frame}
```

Obvykle se úplný obsah prostředí (`environment contents`) umístí do snímku. Pokud se text kvůli své výšce do snímku nevejde, sám se do snímku celý směstná, objeví se výstražná zpráva a text se roztáhne pod snímek a nevypadá dobře. Pomocí volby `allowframebreaks` můžete text rámce rozdělit do několika snímků, avšak nebude možné používat překrytí.

Výchozí specifikace překrytí (`default overlay specification`) je volitelným argumentem, který se detekuje v souladu s následujícím pravidlem: Začíná-li první volitelný argument v hranaté závorce znakem `<`, je tento argument výchozí specifikací překrytí. V opačném případě se jedná o normální argument `options` (volby). Proto bude výraz `\begin{frame}[<+>][plain]` platný stejně jako výraz `\begin{frame}[plain]`.

Příklad

```
\begin{frame}[<+>]
  \begin{theorem}
     $\tilde{a} = B$ .
  \end{theorem}
  \begin{proof}
    \begin{itemize}
      \item Očividně,  $\tilde{a} = C$ .
      \item Jak jsme ukázali dříve,  $C = B$ .
      \item<3-> Thus  $\tilde{a} = B$ .
    \end{itemize}
  \end{proof}
\end{frame}
```

V tomto rámci se daný teorém zobrazuje od prvního snímku a důkaz od druhého snímku, přičemž se první dva výčtové body zobrazí jeden za druhým. Poslední výčtový bod se zobrazí zároveň s prvním. Celkem obsahuje daný rámec čtyři snímky.

Je možné zadat následující volby (options):

- `allowdisplaybreaks="break desirability"` (vhodné zalomení) způsobí, že se na aktuální rámec použije příkaz `AMSTEX` `\allowdisplaybreaks ["break desirability"]`. `Break desirability` může být hodnota mezi 0 (signalizuje, že vzorce se nesmí nikdy zalomit) a 4 (výchozí nastavení – vzorce je možné zalomit kdekoliv bez jakéhokoliv postihu). Tato volba pouze usnadňuje práci a má význam pouze při použití s volbou `allowsframebreaks`.

- `allowframebreaks="fraction"` (frakce). Zadáte-li tuto volbu, rámec se automaticky rozloží do několika rámců v případě, že se text nevejde do jednoho snímku. Při této možnosti tedy dojde k následujícím akcím:

1. Překrytí nejsou podporována.
2. Veškeré poznámky rámce vytvořeného příkazem `\note` budou vloženy za první stránku daného rámce.
3. Veškeré patičky rámce budou vloženy na poslední stránku daného rámce.
4. Je-li zadán titul rámce, bude tento titul rámce na každé stránce se zvláštní poznámkou, která signalizuje, o kterou stránku se v daném rámci jedná. Ve výchozím nastavení je tato zvláštní poznámka napsána římskou číslicí. To je však možné změnit pomocí níže uvedené šablony.

Beamer-Template/-Color/-Font `frametitle continuation` (pokračování titulu rámce) Text této šablony se vkládá na konec každého titulu daného rámce s nastavenou volbou `allowframebreaks`.

Jsou předdefinovány níže uvedené volby šablony:

- `[default]` Jako šablona se instaluje římská číslice. Číslo signalizuje aktuální stránku daného rámce.
- `[roman]` Alias výchozího nastavení.
- `[from second] ["text"]` Instaluje šablonu, která vloží text do všech stránek rámce počínaje druhou stránkou. Ve výchozím nastavení je vkládaným textem `\insertcontinuationtext`, tj. zkratka (cont.).

K dispozici jsou tyto vsuvky:

- `\insertcontinuationcount` vloží aktuální stránku daného rámce ve formě arabské číslice.
- `\insertcontinuationcountroman` vloží aktuální stránku daného rámce ve formě římské číslice (velkými písmeny).
- `\insertcontinuationtext` vloží pouze text (cont.) nebo případně překlad tohoto textu (například (Forts.) v němčině).

Je-li nutné určitý rámeček rozdělit do několika stránek, zaplní se všechny stránky ve výchozím nastavení z 95% s výjimkou poslední stránky. U horního či spodního okraje proto zůstane určité volné místo a to v závislosti na volbě vertikálního uspořádání daného rámce. Tím získáme ve srovnání se stránkou zaplněnou na 100%, což vypadá celkem natěsnaně. Na pohled lepší výsledek však můžeme změnit volitelným argumentem "`fraction`", přičemž 1 znamená 100% a 0.5 znamená 50%. Toto procentní vyjádření zahrnuje i titul rámce. Chcete-li tedy rozdělit určitý rámeček přibližně na polovinu, měli byste jako argument `fraction` zadat hodnotu 0.6. Většina podrobných detailů obvyklého zalamování stránky v \TeX u platí i pro tuto volbu. Pokud si například přejete, aby se rovnice zalamovaly automaticky, nesmíte zapomenout použít příkaz `\allowdisplaybreaks`. Můžete vložit příkazy `\break`, `\nbreak` a `\penalty`, kterými ovládáte umístění jednotlivých zalomení. Funkční jsou i příkazy `\pagebreak` a `\npagebreak`, a to včetně příslušných voleb. Jelikož obvykle nebudete chtít, aby se zalomení stran daného rámce použilo i v režimu `article`, můžete přidat specifikaci režimu, například `<presentation>`. Tím se tyto příkazy použijí pouze na režimy prezentace. Příkaz `\framebreak` je zkratkou příkazu `\pagebreak<presentation>` a `\noframebreak` je zkratkou příkazu `\npagebreak<presentation>`.

Používání této volby je ale celkem nevhodné. V dobré prezentaci připravujte jednotlivé snímky pečlivě a důkladně si rozmyslete, co na který snímek vložit. Používáním volby `allowframebreaks` může dojít k vytvoření nevzhledné a hlavně příliš dlouhé prezentace, která spíše připomíná papír s textem promítnutý na zeď, než opravdovou prezentaci. Nicméně i tato volba má své využití. Především je možné ji použít pro automatické rozdělení bibliografických položek nebo dlouhých rovnic.

Příklad:

```
\begin{frame}[allowframebreaks]{Reference}
  \begin{bibliografie}{XX}
    \bibliografická položka...
    \bibliografická položka...
    ...
    \bibliografická položka...
  \end{bibliografie}
\end{frame}
```

Příklad:

```
\begin{frame}[allowframebreaks,allowdisplaybreaks]{Dlouhá rovnice}
```

```

\begin{align}
\zeta(2) &= 1 + 1/4 + 1/9 + \cdots \\
&= \dots \\
&\dots \\
&= \pi^2/6.
\end{align}
\end{frame}

```

- `b`, `c`, `t` způsobí, že se rámec svisle zarovná ke spodnímu/hornímu okraji nebo na střed. Tím potlačíme obecné nastavení umístování, již řídí volby třídy `t` a `c`.
- `fragile=singleslide` informuje třídu BEAMER o tom, že obsah rámce je tzv. křehký. To znamená, že rámec obsahuje text, který se nechová obvyklým způsobem. To se například vztahuje na doslovný text, který se očividně interpretuje poněkud odlišně než běžný text. Pokud je určitý rámec tvořen tímto křehkým textem, použijí se k vysázení takového rámce jiné vnitřní mechanismy tak, aby bylo možné kódy znaků daného rámce opětovně vysadit. Cenou za přepnutí do jiného vnitřního mechanismu je to, že buď nemůžeme použít překrytí, nebo je nutné zapsat externí soubor a opětovně jej načíst (což není vždy vhodné). Zadáme-li pro normální (pdf)LaTeX tuto volbu, dojde k následujícím akcím: Obsah daného rámce se podrobně prohlédne a poté zapíše do speciálního souboru s názvem "názevúlohy".vrb nebo v případě, že má rámec přiřazenou popisku, do souboru "názevúlohy"."číslo aktuálního rámce".vrb. Následně se tento rámec opět spustí a obsah souboru se načte zpět. Jelikož je možné po načtení souboru upravit kódy znaků, můžeme použít jak doslovný text, tak překrytí. Konce rámce se určí pomocí následujícího pravidla: První výskyt jednotlivého řádku, který obsahuje výraz `\end{"název prostředí rámce"}`, rámec ukončí. Název prostředí je obvykle rámec (`frame`), což je však možné změnit pomocí volby `environment`. Toto speciální pravidlo je potřeba proto, že obsah rámce se koneckonců neinterpretuje při jeho shromáždění. Můžeme také přidat volitelnou informaci `=singleslide`. Tím třídě BEAMER sdělíme, že obsah rámce obsahuje pouze jediný snímek. V tomto případě se obsah rámce nezapíše do speciálního souboru, ale interpretuje se přímo, což je jednak rychlejší a navíc elegantnější.
- `environment="název prostředí rámce"`. Tato volba je užitečná pouze ve spojení s volbou `fragile` (avšak nepoužívá se pro volbu `fragile=singleslide`, pouze pro obyčejnou volbu `fragile`). Název prostředí rámce se používá k určení konce skenování při načítání obsahu rámce. Obvykle

rámec končí po dosažení řádku s výrazem `\end{frame}`. Použijete-li však výraz `\begin{frame}` uvnitř jiného prostředí, je nutné použít tuto volbu:

Příklad:

```
\newenvironment{slide}[1]
  {\begin{frame}[fragile,environment=slide]
   \frametitle{#1}}
  {\end{frame}}
\begin{slide}{Můj titul}
  Text.
\end{slide}
```

Kdybychom ve výše uvedeném příkladu nezadali volbu `environment=slide`, systému \TeX bude "chybět" konec snímku, protože při shromažďování obsahu rámce neinterpretuje text.

- `label="název"` způsobí, že se obsah rámce uloží se jménem `název` pro pozdější obnovení příkazem `\againframe`. Navíc se na každém snímku daného rámce vytvoří značka "`název<<číslo snímku>>`". Dále se na prvním snímku vytvoří značka `název` (značky "`název`" a "`název<1>`" tedy ukazují na stejný snímek). Za povšimnutí stojí, že všeobecně je možné značky, a obzvláště takovéto značky, používat jako cíl hyperlinkových odkazů. Tuto volbu můžete použít spolu s volbou `fragile`.
- `plain` potlačí záhlaví, patičky a postranní panely. To je užitečné při vytváření jednotlivých rámců s různým záhlavím a patičkou nebo při vytváření rámců s velkými obrázky, jež zcela vyplní rámec.

Příklad: Rámec s obrázkem, který rámec úplně vyplňuje.

```
\begin{frame}[plain]
  \begin{centering}%
    \pgfimage[height=\paperheight]{nějakýsouborsvelkýmob-
rázkem}%
  \par%
  \end{centering}%
\end{frame}
```

Příklad: Titulní stránka, kde jsou záhlaví a zápatí nahrazeny dvěma obrázky.

```

\setbeamertemplate{titulní stránka}
{
  \pgfuseimage{toptitle}
  \vskip0pt plus 1filll

  \begin{centering}
    {\usebeamerfont{title}\usebeamercolor[fg]{title}\in-
serttitle}

    \insertdate
  \end{centering}%
  \vskip0pt plus 1filll
  \pgfuseimage{bottomtitle}
}
\begin{frame}[plain]
  \titlepage
\end{frame}

```

- `shrink="minimální procentní hodnota zmenšení"`. Ta volba zmenší text daného rámce v případě, že je text příliš rozsáhlý a do rámce se nevejde. Třída BEAMER nejdříve běžně vysází celý rámeček. Pak si prohlédne vertikální velikost textu rámce (bez titulu rámce). Je-li tato vertikální velikost větší než výška textu minus výška titulu rámce, třída BEAMER spočítá faktor zmenšení a text rámce zmenší tímto faktorem tak, že text úplně daný rámeček vyplní. Použijete-li tuto volbu, použije se automaticky i volba `squeeze`.

Jelikož ke zmenšení dochází až po vysázení veškerého obsahu, nedojde k úplnému horizontálnímu vyplnění rámce textem. Z tohoto důvodu můžete zadat *minimální procentní hodnotu zmenšení*, například 20. Zadáte-li takovou hodnotu, rámeček se zmenší alespoň touto procentní hodnotou. Jelikož si toho je třída BEAMER vědoma, může doporučeně zvětšit horizontální šířku tak, že zmenšený text opět vyplní celý rámeček. Není-li však procentuální hodnota dostatečná, text se zmenší podle potřeby, avšak objeví se varovná zpráva.

Nejlepším způsobem, jak používat tuto volbu, je identifikovat přeplněné rámce, u nichž je však veškerý text nutné zachovat v jednom jediném rámci. Poté zadejte nejdříve `shrink=5`, poté `shrink=10` a tak dále, dokud nezmizí všechny varovné zprávy (nebo varovné zprávy jednoduše ignorujte, pokud vše vypadá uspokojivě).

Používání této volby je velmi *nevhodné*. Na jednotlivých snímcích totiž dojde k různým změnám velikosti fontu, což je typografická noční můra.

Této volbě se *vždy můžete* vyhnout restrukturalizací a zjednodušením rámců. Výsledkem bude lepší prezentace.

Příklad:

```
\begin{frame}[shrink=5]
  Nějaký nevhodný snímek se spoustou textu, který o 5%
  překračuje vhodnou velikost.
\end{frame}
```

- `squeeze` nahustí veškerý svislý prostor textu co nejvíce k sobě. Aktuálně tato volba zredukuje veškerý svislý prostor ve výčtech a jednotlivě rozepsaných položkách na nulu. Používání této volby není ani vhodné ani nevhodné.

Beamer-Template frame begin (začátek rámce)

Text této šablony se vkládá na začátek každého rámce v režimu `article` (a pouze tam). Můžete to použít například k uvedení prostředí `minipage` na začátku rámce nebo k vložení vodorovné linky či jiných objektů.

Beamer-Template frame end (konec rámce)

Text této šablony se vkládá na konec každého rámce v režimu `article`.

Prostředí rámce můžete použít v jiných prostředích, například takto:

```
\newenvironment{slide}{\begin{frame}}{\end{frame}}
```

nebo takto:

```
\newenvironment{myframe}[1]
{\begin{frame}[fragile,environment=myframe]\frametitle{#1}}
{\end{frame}}
```

Vlastní mechanismus je však poněkud citlivý, jelikož „shromáždění“ obsahu rámce není jednoduché. Nezkoušejte tedy nic příliš komplikovaného. Zpravidla může být začátek prostředí z velké části libovolný, avšak konec musí být ukončen výrazem `\end{frame}` a neměl by obsahovat žádný výraz `\end{xxx}`. Veškeré příliš komplikované výrazy pravděpodobně selžou. Potřebujete-li zadat výraz `\end{xxx}`, definujte nový příkaz, který tento výraz obsahuje, podle níže uvedeného příkladu:

```

\newenvironment{itemizeframe}
  {\begin{frame}\startitemizeframe}
  {\stopitemizeframe\end{frame}}
\newcommand\startitemizeframe{\begin{bfseries}\begin{ite-
mize}}
\newcommand\stopitemizeframe{\end{itemize}\end{bfseries}}

\begin{itemizeframe}
  \item První položka
\end{itemizeframe}

```

5.3.1.2 Součásti rámce

Každý rámec sestává z několika součástí:

1. záhlaví a zápatí,
2. levý a pravý postranní panel,
3. navigační panely,
4. navigační symboly,
5. logo,
6. titul rámce,
7. pozadí a
8. nějaký obsah rámce.

Rámec nemusí mít všechny tyto součásti. Obvykle se první tři součásti nastaví automaticky podle používaného tématu.

5.3.1.2.1 Záhloví a zápatí

Záhlovím rámce je prostor ve vrchní části rámce. Není-li prázdné, mělo by obsahovat nějaké informace, které divákům napomohou se zorientovat ve vaší přednášce. Stejně tak je zápatím prostor ve spodní části rámce.

Třída BEAMER nepoužívá standardní mechanismus L^AT_EX pro vysázení záhlaví a zápatí. Namísto toho se pro vysázení záhlaví a zápatí použijí speciální šablony. Velikost záhlaví a zápatí se určí následovně: Jejich šířka je vždy šířkou papíru. Jejich výška se určí pokusným vysázením záhlaví a zápatí hned po příkazu `\begin{document}`. Hlavička záhlaví a zápatí se v tomto bodě "zmrazí" a použije se napříč celým dokumentem i v případě, že

se výška záhlaví a zápatí později liší (což by se nemělo stát).

Vzhled záhlaví a zápatí je určen následujícími šablonami:

Beamer-Template/-Color/-Font headline

Tato šablona se použije k vysázení záhlaví. Barvy a font záhlaví `BEAMER-color` a `-font headline` se instalují na začátku. Ve výchozím nastavení se nepoužívá pozadí barvy `BEAMER-color`, tj. za záhlavím ani zápatím se nevykreslí žádný obdélník s pozadím (to se může v budoucnu změnit s nástupem plátna (canvas) záhlaví a zápatí).

Šířkou záhlaví je celá šířka papíru. Výška se určuje automaticky podle výše uvedeného popisu. Záhlaví se vysází ve vertikálním režimu s vypnutým nastavením přeskokování vložených řádek (`interline skip`) a přeskokováním odstavců (`paragraph skip`) nastaveným na hodnotu nula.

V této šabloně se příkaz `\` změní tak, že se namísto toho vloží čárka.

Příklad:

```
\setbeamertemplate{headline}
{%
  \begin{beamercolorbox}{section in head/foot}
    \vskip2pt\insertnavigation{\paperwidth}\vskip2pt
  \end{beamercolorbox}%
}
```

Jsou předdefinovány níže uvedené volby šablony:

- Výchozí šablonou je prázdné záhlaví. Chcete-li získat výchozí záhlaví z dřívějších verzí třídy `BEAMER`, použijte téma `compatibility`.
- [`infolines theme`] Tato volba bude k dispozici (a použije se) v případě, že se načte vnější téma `infolines`. Záhlaví zobrazuje aktuální oddíl a pododdíl.
- [`miniframes theme`] Tato volba bude k dispozici (a použije se) v případě, že se načte vnější téma `miniframes`. Záhlaví zobrazí oddíly, pod nimiž jsou malé mini rámce, na něj je možné kliknout.
- [`sidebar theme`] Tato volba bude k dispozici (a použije se) v případě, že se načte vnější téma `sidebar` a výška záhlaví (a volba tématu `sidebar`) není nulová. V tomto případě je záhlaví prázdnou lištou v barvě pozadí `frametitle` s logem v levé či pravé části.

- `[smoothtree theme]` Tato volba bude k dispozici (a použije se) v případě, že se načte vnější téma `smoothtree`. Vyhlazený strom navigace se zobrazí v záhlaví.
- `[smoothbars theme]` Tato volba bude k dispozici (a použije se) v případě, že se načte vnější téma `smoothbars`. Zobrazí se uhlazená verze záhlaví `miniframes`.
- `[tree]` Tato volba bude k dispozici (a použije se) v případě, že se načte vnější téma `tree`. Strom navigace se zobrazí v záhlaví.
- `[split theme]` Tato volba bude k dispozici (a použije se) v případě, že se načte vnější téma `split`. Záhlaví se rozdělí na levou část, v níž se zobrazí oddíly, a na pravou část s vyobrazením pododdílů.
- `[text line]{text}` Záhlaví se víceméně vysází tak, jako by se jednalo o běžnou textovou řádku s obsahem *text*. Levý a pravý okraj se nastaví do stejné polohy, jakou mají okraje běžného textu. *Text* se vysází uvnitř příkazu `\hbox` a záhlaví se normálně vysází ve vertikálním režimu.

Uvnitř šablony je možné použít několik vsuvek:

- `\insertnavigation{width}` Vloží do šablony horizontální lištu dané *šířky*. Lišta zobrazuje oddíly a pod nimi mini rámce jednotlivých rámců daného oddílu.
- `\insertpagenumber` Vloží do šablony číslo aktuální stránky.
- `\insertsection` Vloží do šablony aktuální oddíl.
- `\insertsectionnavigation{width}` Vloží vertikální navigační lištu, která obsahuje všechny oddíly, a zvýrazní aktuální oddíl.
- `\insertsectionnavigationhorizontal{width}{left insert}{right insert}` Vloží horizontální navigační lištu, která obsahuje všechny oddíly, a zvýrazní aktuální oddíl. *Left insert* se vloží do levé části oddílů, *right insert* do pravé části. Vložíte-li trojitou výplň (`fill`), můžete lištu zarovnat vlevo, nebo vpravo.

Příklad:

```
\insertsectionnavigationhorizontal{.5\textwidth}{\hskip0pt
plus1filll}{}
```


- `\insertshortauthor["options"]` Vloží do šablony zkrácenou verzi autora. Text se vytiskne do jednoho dlouhého řádku, zalomení řádků nastavené pomocí příkazu `\\` se potlačí. Je možné zadat následující *volby*:
 - `width="width"` Vloží text do víceřádkové ministránky o dané velikosti. Ve výchozím nastavení je i nadále zalomení řádků potlačeno.
 - `center` Vystředí text uvnitř ministránky vytvořené volbou `width`. Nezarovná text vlevo.
 - `respectlinebreaks` Vynutí zalomení řádků zadaných příkazem `\\`.

Příklad:

```
\insertauthor[width={3cm},center,respectlinebreaks]
```

- `\insertshortdate["options"]` Vloží do šablony krátkou verzi data. Je možné zadat stejné volby jako pro `\insertshortauthor`.
- `\insertshortinstitute["options"]` Vloží do šablony krátkou verzi institutu. Je možné zadat stejné volby jako pro `\insertshortauthor`.
- `\insertshortpart["options"]` Vloží do šablony krátkou verzi názvu části. Je možné zadat stejné volby jako pro `\insertshortauthor`.
- `\insertshorttitle["options"]` Vloží do šablony krátkou verzi titulu dokumentu. Je možné zadat stejné volby jako pro `\insertshortauthor`.
- `\insertshortsubtitle["options"]` Vloží krátkou verzi podtitulu dokumentu. Je možné zadat stejné volby jako pro `\insertshortauthor`.
- `\insertsubsection` Vloží do šablony aktuální pododdíl.
- `\insertsubsubsection` Vloží do šablony aktuální pododdíl.
- `\insertsubsectionnavigation{"width"}` Vloží vertikální navigační panel, který obsahuje všechny pododdíly aktuálního oddílu, a aktuální pododdíl zvýrazní.
- `\insertsubsectionnavigationhorizontal{"width"}{"left insert"}{"right insert"}` Viz `\insertsectionnavigationhorizontal`.
- `\insertverticalnavigation{"width"}` Vloží do šablony vertikální navigační panel dané šířky *width*. Panel zobrazí přehled kapitol prezentace. Jednotlivé řádky se vysází pomocí šablon `section in head/foot` a `subsection in head/foot`.

- `\insertframenumber` Vloží do šablony číslo aktuálního rámce (nikoliv snímku).
- `\inserttotalframenumber` Vloží do šablony celkový počet rámců (nikoliv snímků). Číslo se zobrazí správně až poté, co systém systém \TeX podruhé projde daný dokument.
- `\insertframestartpage` Vloží číslo stránky, které má první stránka aktuálního rámce.
- `\insertframeendpage` Vloží číslo stránky, které má poslední stránka aktuálního rámce.
- `\insertsubsectionstartpage` Vloží číslo stránky, které má první stránka aktuálního pododdílu.
- `\insertsubsectionendpage` Vloží číslo stránky, které má poslední stránka aktuálního pododdílu.
- `\insertsectionstartpage` Vloží číslo stránky, které má první stránka aktuálního oddílu.
- `\insertsectionendpage` Vloží číslo stránky, které má poslední stránka aktuálního oddílu.
- `\insertpartstartpage` Vloží číslo stránky, které má první stránka aktuální části.
- `\insertpartendpage` Vloží číslo stránky, které má poslední stránka aktuální části.
- `\insertpresentationstartpage` Vloží číslo stránky, které má první stránka prezentace.
- `\insertpresentationendpage` Vloží číslo stránky, které má poslední stránka prezentace (bez dodatku).
- `\insertappendixstartpage` Vloží číslo stránky, které má první stránka dodatku. Není-li v dokumentu žádný dodatek, je toto číslo poslední stránkou dokumentu.
- `\insertappendixendpage` Vloží číslo stránky, které má poslední stránka dodatku. Není-li v dokumentu žádný dodatek, je toto číslo poslední stránkou dokumentu.

- `\insertdocumentstartpage` Vloží 1.
- `\insertdocumentendpage` Vloží číslo stránky, které má poslední stránka dokumentu (včetně dodatku).

Beamer-Template/-Color/-Font footline (zápatí)

Tato šablona se chová úplně stejně jako záhlaví. Upozorňujeme, že v současné době třída `BEAMER` občas přidává mezeru o velikosti 4 pt mezi spodní část textu rámce a horní část zápatí, což může poněkud obtěžovat. Jsou předdefinovány níže uvedené volby šablony:

- `[default]` Výchozí hodnotou je prázdné zápatí. Upozorňujeme, že navigační symboly nejsou ve výchozím nastavení součástí zápatí. Místo toho jsou součástí (neviditelného) pravého postranního panelu.
- `[infolines theme]` Tato volba bude k dispozici (a použije se) v případě, že se načte vnější téma `infolines`. Zápatí zobrazuje například informace o jménu autora a tématu prezentace.
- `[miniframes theme]` Tato volba bude k dispozici (a použije se) v případě, že se načte vnější téma `miniframes`. V závislosti na přesném nastavení voleb, které se používají při načtení tématu `miniframes`, se v zápatí mohou zobrazit různé informace.
- `[page number]` Zobrazí v zápatí číslo aktuální stránky.
- `[frame number]` Zobrazí v zápatí číslo aktuálního rámce.
- `[split]` Tato volba bude k dispozici (a použije se) v případě, že se načte vnější téma `split`. Zápatí (stejně tak jako záhlaví) se rozdělí na levou část, v níž se zobrazí jméno autora, a na pravou část s informacemi o názvu prezentace.
- `[text line]{text}` Zápatí se víceméně vysází tak, jako by se jednalo o běžnou textovou řádku s obsahem `text`. Levý a pravý okraj se nastaví do stejné polohy, jakou mají okraje běžného textu. Text se vysází uvnitř příkazu `\hbox` a záhlaví se normálně vysází ve vertikálním režimu. V takovém řádku se může osvědčit použití příkazu `\strut`.

Je možné použít stejné vsuvky jako pro záhlaví.

Beamer-Color/-Font page number in head/foot (číslo stránky v záhlaví/zápatí)

Volby `BEAMER-color` a `-font` se používají k vysázení čísla stránky či rámce v zápatí.

5.3.1.2.2 Postranní panely

Postranní panely jsou vertikálními oblastmi, které se rozprostírají mezi spodní částí záhlaví a vrchní částí zápatí. Jeden postranní panel může být vlevo či vpravo (nebo na obou stranách). V postranních panelech se může zobrazovat přehled kapitol dokumentu nebo můžete panely přidat z čistě estetických důvodů.

Instalujete-li šablonu postranního panelu, je nutné explicitně specifikovat horizontální velikost tohoto panelu příkazem `\setbeamesize` a volbou `sidebar left width` nebo `sidebar right width`. Vertikální velikost se určuje automaticky. Každý postranní panel má své plátno (canvas) s pozadím, které je možné nastavit pomocí šablon plátna postranních panelů.

Přidáte-li postranní panel o velikosti například 1 cm, hlavní text se zúží o 1 cm. Vzdálenost mezi vnitřní stranou postranního panelu a vnější stranou textu, která je určena příkazem `\setbeamersize` s volbou `text margin left` nebo opačnou volbou pro pravý okraj, se při instalaci postranního panelu nezmění.

Interně se postranní panely vysází tak, že se zobrazí jako součást záhlaví. Třída `textscBEAMER` si uchovává informace o šesti dimenzích, tři pro každou stranu: proměnné `\beamer@leftsidebar` a `\beamer@rightsidebar` uchovávají (horizontální) velikosti postranních panelů, proměnné `\beamer@leftmargin` a `\beamer@rightmargin` uchovávají vzdálenost mezi postranním panelem a textem, a makra `\Gm@lmargin` a `\Gm@rmargin` uchovávají vzdálenost mezi okrajem papíru a okrajem textu. Součet `\beamer@leftsidebar` a `\beamer@leftmargin` je tedy přesně `\Gm@lmargin`. Přejete-li si tedy vložit nějaký text přímo vedle levého postranního panelu, zápisem `\hskip-\beamer@leftmargin` se na toto místo dostanete.

Beamer-Template/-Color/-Font sidebar left (levý postranní panel)

Nadřazený prvek pro volbu `color/font`: `sidebar`.

Tato šablona se používá pro vysázení levého postranního panelu. Jak jsme již zmínili výše, velikost levého postranního panelu se nastavuje příkazem `\setbeamesize{sidebar width left=2cm}`

Jsou-li panely příliš rozměrné, třída `textscBEAMER` je automaticky neorízne.

Postranní panel se po vysázení umístí do příkazu `\vbox`. Aktuálně byste měli nastavovat příkazy typu `\hsize` nebo `\parskip` sami. Jsou předdefinovány níže uvedené volby šablony:

- `[default]` instaluje prázdnou šablonu.
- `[sidebar theme]` Tato volba je k dispozici po načtení vnějšího tématu `sidebar` s volbou `left`. V tomto případě se tyto volby vyberou automaticky. V postranním panelu se zobrazí malý přehled kapitol.

Beamer-Template/-Color/-Font sidebar right (pravý postranní panel)

Nadřazený prvek pro volbu `color/font`: `sidebar`. Tato šablona funguje stejně jako šablona pro levý panel.

Jsou předdefinovány níže uvedené volby šablony:

- `[default]` Výchozí pravý postranní panel má nulovou šířku. Zobrazují se v něm však navigační symboly. Ve spodní části postranního panelu se také zobrazuje logo (je-li instalováno), které vystupuje vlevo a zasahuje tak do textu.
- `[sidebar theme]` Tato volba je k dispozici po načtení vnějšího tématu `sidebar` s volbou `left`. V tomto případě se tyto volby vyberou automaticky. V postranním panelu se zobrazí malý přehled kapitol.

Beamer-Template sidebar canvas left (levý postranní panel s plátnem)

Stejně tak jako celkové plátno pozadí se i toto plátno vykresluje za vlastním textem postranního panelu. Šablona by měla za normálních okolností vložit obdélník o velikosti postranního panelu, avšak přílišná výška nevyvolá chybu ani varovnou zprávu. Zavoláte-li tuto šablonu, nainstaluje se `BEAMER-color sidebar left`.

Jsou předdefinovány níže uvedené volby šablony:

- ve výchozím nastavení se jako plátno postranního panelu používá velký obdélník obarvený pomocí `sidebar.bg`. Je-li však pozadí postranního panelu prázdné, nevykreslí se nic a plátno bude "průhledné".
- `[vertical shading] ["color options"]` instaluje vertikálně stínované pozadí. Je možné zadat následující volby *color options*:
 - `top="color"` specifikuje barvu ve vrchní části postranního panelu.

Ve výchozím nastavení se používá 25% z přední části palety `BEAMER-color palette primary`.

- `bottom="color"` specifikuje barvu spodní části postranního panelu (přesněji ve vzdálenosti výšky stránky pod vrchní částí postranního panelu). Ve výchozím nastavení je to pozadí normálního textu (`normal text`) v okamžiku volání tohoto příkazu.
- `middle="color"` specifikuje barvu střední části postranního panelu. Pokud tuto volbu zadáte, barva stínování se mění při přechodu ze spodní části do střední části a ze střední části do vrchní části.
- `midpoint="factor"` specifikuje místo na stránce, na němž se použije barva pro střední část stránky. Faktor o velikosti 0 je spodní částí stránky, faktor 1 je její vrchní částí. Výchozí hodnotou je 0.5, což je střední část stránky. Upozorňujeme, že je zde nutné zadat "reálné" barvy \LaTeX . Před použitím tohoto příkazu je proto často nutné zavolat příkaz `\usebeamercolor`. Dále upozorňujeme, že šířku postranního panelu byste měli nastavit předtím, než použijete tuto volbu.

Příklad: Stylové, ačkoliv nepříliš použitelné stínování:

```
{\usebeamercolor{palette primary}}
  \setbeamertemplate{sidebar canvas}[vertical shading]
[top=palette primary.bg,middle=white,bottom=palette primary.bg]
```

- `[horizontal shading] ["color options"]` instaluje horizontálně stínované pozadí. Je možné zadat následující volby *color options* :
 - `left="color"` specifikuje barvu v levé části postranního panelu.
 - `right="color"` specifikuje barvu v pravé části postranního panelu.
 - `middle="color"` specifikuje barvu ve střední části postranního panelu.
 - `midpoint="factor"` specifikuje místo na stránce, na němž se použije barva pro střední část stránky. Faktor o velikosti 0 je levou částí postranního panelu, faktor 1 je pravou částí. Výchozí hodnotou je 0.5, což je střední část stránky.

Příklad: Přidá dva "sloupce"

```
\setbeamersize{sidebar width left=0.5cm,sidebar width right=0.5cm}
```

```
{\usebeamercolor{sidebar}}
```

```
\setbeamertemplate{sidebar canvas left}[horizontal shading]
[left=white,middle=sidebar.bg,right=white]
\setbeamertemplate{sidebar canvas right}[horizontal shading]
[left=white,middle=sidebar.bg,right=white]
```

Beamer-Template sidebar canvas right (pravý postranní panel s plátnem)

Funguje stejně jako levý panel.

5.3.1.2.3 Navigační lišty

Řada témat instaluje záhlaví nebo postranní panel, jež obsahují navigační lištu. Ačkoliv takové navigační lišty zabírají docela dost místa, jsou často užitečné ze dvou důvodů:

- Poskytují posluchačům vizuální zpětnou vazbu, jež jim signalizuje, jak velkou část prezentace jste již odpřednášeli a o čem ještě budete mluvit. Bez takové zpětné vazby si posluchači často nejsou jistí, zda aktuálně načrtnuté téma ještě později rozvinete, nebo se k němu již nevrátíte.
- Můžete kliknout na kteroukoliv část navigační lišty. Tím ihned přeskočíte k části, na niž jste kliknuli. To je obzvláště užitečné v případě, že chcete přeskočit určité části prezentace nebo když zodpovídáte dotazy a potřebujete se vrátit k určitému rámci.

Některé navigační lišty je možné "stlačit" následující volbou:

```
\documentclass[compress]{beamer}
```

Pokusí se všechny navigační lišty co nejvíce zmenšit. Všechny zmenšené reprezentace rámců jednotlivého oddílu se například zobrazí v navigačních lištách vedle sebe. Obvykle se reprezentace různých pododdílů zobrazují v různých řádkách. Navíc se navigace oddílů a pododdílů stlačí do jediného řádku.

Některá témata používají příkaz `\insertnavigation`, který vloží navigační lištu do záhlaví. V této liště se zobrazí malé ikony (nazývané "mini rámce"), jež reprezentují rámce určité prezentace. Kliknete-li na takovou ikonu, dojde k následujícím akcím:

- Kliknete-li na jiný než aktuální rámec (na jeho ikonu), prezentace odskočí na první snímek rámce, na nějž jste kliknuli.
- Kliknete-li na aktuální rámec a nenacházíte se přitom na posledním snímku tohoto rámce, skočíte na poslední snímek daného rámce.

- Kliknete-li na aktuální rámeček a nacházíte se přitom na posledním snímku tohoto rámečku, skočíte na první snímek daného rámečku.

V souladu s výše uvedenými pravidly můžete:

- Jednoduchým kliknutím skočit na začátek určitého rámečku z jiného místa.
- Dvojitým kliknutím skočit na konec určitého rámečku z jiného místa.
- Jednoduchým kliknutím přeskocit zbytek aktuálního rámečku.

Dále jsem se pokusil vytvořit automatický odskok na poslední snímek již navštíveného rámečku. Ukázalo se však, že takový skok je spíše matoucí než užitečný. V aktuální implementaci se poklepáním vždy dostanete na konec snímku bez ohledu na to, odkud jste na něj "přešli".

Parent Beamer-Template mini frames (mini rámeček)

Tato nadřazená šablona má podřazené mini rámečky `mini frame` a `mini frame in current subsection`.

Příklad: `\setbeamertemplate{mini frames}[box]`

Jsou předdefinovány níže uvedené volby šablony:

- `[default]` mini rámeček zobrazí malými kroužky.
- `[box]` mini rámeček zobrazí malými obdélníčky.
- `[tick]` mini rámeček zobrazí malými vertikálními čarami.

Beamer-Template/-Color/-Font mini frame (mini rámeček)

Šablona se použije k zobrazení mini rámečku aktuálního rámečku v navigační liště.

Šířka šablony se ignoruje. Zobrazí-li se několik mini rámečků, vypočítá se jejich poloha v závislosti na rozměrech `BEAMER-sizes mini frame size` a `mini frame offset`. Další informace s popisem jejich změny najdete u příkazu `\setbeamersize`.

Beamer-Template mini frame in current subsection (mini rámeček v aktuálním pododdílu)

Tato šablona se použije k zobrazení mini rámečků aktuálního pododdílu,

jež nejsou aktuálním rámcem. Před použitím této šablony se instaluje `BEAMER-color/-font mini frame`.

Beamer-Template mini frame in other subsection (mini rámeček v jiném pododdílu)

Tato šablona se použije k zobrazení mini rámečků, které reprezentují rámce z pododdílů, jež nejsou aktuálním pododdílem.

Jsou předdefinovány níže uvedené volby šablony:

- `[default] ["percentage"]` Ve výchozím nastavení tato šablona zobrazuje mini rámeček v aktuálním pododdílu. Barva se však nejdříve změní na `fg! "percentage" !bg`. Výchozí hodnotou `"percentage"` je 50%.

Příklad: Chcete-li získat extrémně stínované znázornění rámečků, které se nacházejí mimo aktuální pododdíl, můžete použít níže uvedený příkaz:

```
\setbeamertemplate{mini frame in other subsection}[default][20]
```

Příklad: Chcete-li stejným způsobem vykreslit všechny mini rámce s výjimkou aktuálního rámce, použijte příkaz

```
\setbeamertemplate{mini frame in other subsection}[default][100]
```

Některá témata zobrazují oddíly a pododdíly v navigačních lištách. Kliknutím v navigační liště na určitý oddíl či pododdíl přejdete ke zvolenému oddílu. Kliknutí na určitý oddíl je obzvláště užitečné v případě, že daný oddíl začíná příkazem `\tableofcontents[currentsection]`, protože to můžete využít k přechodu na různé pododdíly.

Beamer-Template/-Color/-Font section in head/foot (oddíl v záhlaví/zápatí)

Tato šablona se použije k zobrazení položky oddílu, která se vyskytne v záhlaví či zápatí. Pozadí `BEAMER-color` se obvykle používá jako pozadí celé oblasti, v níž se v záhlaví zobrazují položky oddílu. Obvykle nemůžete tuto šablonu použít sami, protože vsuvka `\insertsectionhead` se správně nastaví pouze v případě, že se do záhlaví právě sází seznam oddílů.

Výchozí šablona pouze vloží název oddílu. Pro tuto šablonu jsou užitečné níže uvedené vsuvky:

- `\insertsectionhead` vloží název oddílu, který se má vysázet do navigační lišty.

- `\insertsectionheadnumber` vloží číslo oddílu, který se má vysázet do navigační lišty.
- `\insertpartheadnumber` vloží číslo části aktuálního oddílu či pododílu, které se mají vysázet do navigační lišty.

Beamer-Template section in head/foot shaded (stínovaný oddíl v záhlaví/zápatí)

Tato šablona se používá místo oddílu v záhlaví/zápatí pro vysázení oddílů, jež jsou aktuálně stínované. Takové stínování se obvykle použije na všechny oddíly s výjimkou toho aktuálního.

indent Tato šablona nemá vlastní barvu ani font. Zavoláte-li tuto šablonu, nastaví se font BEAMER-font a barva section in head/foot. Při spuštění šablony obvykle změňte aktuální barvu nebo spustíte prostředí `colormixin`.

Jsou předdefinovány níže uvedené volby šablony:

- `[default] ["percentage"]` Tato výchozí šablona změní aktuální barvu na `fg! "percentage" !bg`. Aktuální barva se zesvětlí nebo ztmaví. Výchozí procentní hodnotou je 50%.

Příklad: Následující příkaz můžete použít k intenzivnímu zesvětlení stínovaných položek:

```
\setbeamertheme{section in head/foot shaded}[default] [20]
```

Beamer-Template/-Color/-Font section in sidebar (oddíl v postranním panelu)

Tato šablona se použije k zobrazení položky oddílu, která se vyskytne v postranním panelu a obvykle je součástí zde zobrazeného mini přehledu kapitol. Pozadí BEAMER-color se obvykle používá jako pozadí dané položky.

Stejně jako v případě section in head/foot, ani tuto šablonu obvykle nemůžete použít sami a měli byste použít i příkaz `\insertsectionhead`, který vloží název oddílu určeného k vysázení. Pro tento případ se neinstaluje žádné výchozí nastavení této šablony.

Jsou předdefinovány níže uvedené volby šablony:

- `[sidebar theme]` Tato šablona, jež je k dispozici pouze v případě, kdy se načte vnější téma sidebar, vloží lištu s popředím v barvě BEAMER-color a pozadím, jenž zobrazuje název oddílu.

Šířka lišty je stejná jako šířka celého postranního panelu.

Je možné použít stejné vsuvky jako pro section in head/foot.

Beamer-Template/-Color section in sidebar shaded (stínovaný oddíl v postranním panelu)

Tato šablona se používá místo `section in sidebar` pro vysázení oddílů, jež jsou aktuálně stínované. Takové stínování se obvykle použije na všechny oddíly s výjimkou toho aktuálního. Na rozdíl od `section in head/foot shaded` má tato šablona vlastní barvu `BEAMER-color`.

Jsou předdefinovány níže uvedené volby šablony:

- `[sidebar theme]` Funguje stejně jako v případě nestínované verze. Jen se použije odlišná barva `BEAMER-color`.

Beamer-Template/-Color/-Font subsection in head/foot (pododdíl v záhlaví/zápatí)

Tato šablona funguje úplně stejně jako `section in head/foot`, avšak pracuje s pododdíly.

- `\insertsubsectionhead` pracuje stejně jako `\insertsectionhead`.
- `\insertsubsectionheadnumber` pracuje stejně jako `\insertsectionheadnumber`.

Beamer-Template subsection in head/foot shaded (stínovaný pododdíl v záhlaví/zápatí)

Tato šablona funguje úplně stejně jako `section in head/foot shaded`, avšak pracuje s pododdíly.

Jsou předdefinovány níže uvedené volby šablony:

- `[default] ["percentage"]` pracuje stejně jako příslušná volba pro oddíly.

Příklad:

```
\setbeamertemplate{section in head/foot shaded}[default] [20]
\setbeamertemplate{subsection in head/foot shaded}[default]
[20]
```

Beamer-Template/-Color/-Font subsection in sidebar (pododdíl v postranním panelu)

Tato šablona funguje úplně stejně jako `section in sidebar`, avšak pracuje

s pododdíly.

Beamer-Template subsection in sidebar shaded (stínovaný pododdíl v postranním panelu)

Tato šablona funguje úplně stejně jako `section in sidebar shaded`, avšak pracuje s pododdíly.

Beamer-Template/-Color/-Font subsubsection in head/foot (podpododdíl v záhlaví/zápatí)

Tato šablona funguje úplně stejně jako `section in head/foot`, avšak pracuje s podpododdíly. Aktuálně ji výchozí témata nepoužívají.

- `\insertsubsubsectionhead` pracuje stejně jako `\insertsectionhead`.
- `\insertsubsubsectionheadnumber` pracuje stejně jako `\insertsectionheadnumber`.

Beamer-Template subsubsection in head/foot shaded (stínovaný podpododdíl v záhlaví/zápatí)

Tato šablona funguje úplně stejně jako `section in head/foot shaded`, ale pracuje s podpododdíly.

Jsou předdefinovány níže uvedené volby šablony:

- `[default] ["percentage"]` pracuje stejně jako příslušná volba pro oddíly.

Beamer-Template/-Color/-Font subsubsection in sidebar (podpododdíl v postranním panelu)

Tato šablona funguje úplně stejně jako `section in sidebar`, avšak pracuje s podpododdíly.

Beamer-Template subsubsection in sidebar shaded (stínovaný podpododdíl v postranním panelu)

Tato šablona funguje úplně stejně jako `section in sidebar shaded`, avšak pracuje s podpododdíly.

Kliknutím na titul dokumentu v navigační liště (nezobrazují ji všechna témata) přejdete k prvnímu snímku prezentace (obvykle titulní stránka), tedy pokud se již na první stránce nenacházíte. Kliknete-li na prvním snímku na

titul dokumentu, přejdete na konec existující prezentace. Poklepáním na titul dokumentu v navigační liště můžete tedy přeskočit na konec.

5.3.1.2.4 Navigační symboly

Navigační symboly jsou malé ikony, které se standardně zobrazují na každém snímku. K dispozici jsou níže uvedené symboly:

1. Ikona snímku, která se zobrazuje jako jednoduchý obdélník. Vlevo a vpravo od této ikony se zobrazuje šipka vlevo a vpravo.
2. Ikona rámce, která se zobrazuje jako tři ikony snímku položené na sobě. Tento symbol je zarámován šipkami.
3. Ikona pododdílu, která se zobrazuje jako zvýrazněná položka pododdílu v přehledu kapitol. Tento symbol je zarámován šipkami.
4. Ikona oddílu, která se zobrazuje jako zvýrazněná položka oddílu v přehledu kapitol (spolu se všemi pododdíly). Tento symbol je zarámován šipkami.
5. Ikona prezentace, která se zobrazuje jako zcela zvýrazněný přehled kapitol.
6. Ikona dodatku, která se zobrazuje jako zcela zvýrazněný přehled kapitol sestávající pouze z jednoho oddílu. (Tato ikona se zobrazuje pouze v případě, že je připojen dodatek.)
7. Ikony pro pohyb vzad a vpřed se zobrazují jako oválné šipky.
8. Ikona hledání nebo vyhledání, která se zobrazuje jako detektivní lupa.

Kliknutím na levou šipku vedle určité ikony vždy přeskočíte k předcházejícímu snímku (poslednímu snímku), rámci, pododdílu nebo oddílu. Kliknutím na pravou šipku vedle určité ikony vždy přeskočíte k následujícímu snímku (prvnímu snímku), rámci, pododdílu nebo oddílu.

Kliknutím na kteroukoliv z těchto ikon spustíte různé akce:

1. Pokud to prohlížeč dovoluje, můžete kliknutím na ikonu snímku otevřít okno, v němž zadáte číslo snímku, na nějž se má přeskočit.
2. Kliknutím na levou stranu ikony snímku přeskočíte na první snímek daného rámce. Kliknutím na pravou stranu přeskočíte na poslední snímek rámce (to může být užitečné, chcete-li přeskočit překrytí).

3. Kliknutím na levou stranu ikony pododdílu přeskočíte na první snímek daného pododdílu. Kliknutím na pravou stranu přeskočíte na poslední snímek pododdílu.
4. Kliknutím na levou stranu ikony oddílu přeskočíte na první snímek daného oddílu. Kliknutím na pravou stranu přeskočíte na poslední snímek oddílu.
5. Kliknutím na levou stranu ikony prezentace přeskočíte na první snímek. Kliknutím na pravou stranu přeskočíte na poslední snímek prezentace. To se však nevztahuje na dodatek.
6. Kliknutím na levou stranu ikony dodatku přeskočíte na první snímek daného dodatku. Kliknutím na pravou stranu přeskočíte na poslední snímek dodatku.
7. Pokud to prohlížeč umožňuje, kliknutím na symboly pro pohyb vzad a vpřed přeskočíte na dříve navštívené snímky.
8. Pokud to prohlížeč umožňuje, kliknutím na ikonu vyhledávání otevřete okno, do nějž můžete zadat hledaný řetězec. Bude-li řetězec nalezen, prohlížeč k tomuto řetězci přeskočí.

Počet zobrazených ikon můžete snížit nebo změnit jejich rozložení úpravou šablony `navigation symbols`.

Beamer-Template/-Color/-Font navigation symbols (navigační symboly)

Tuto šablonu vyvolají témata v "režimu tří hvězdiček" v místě, kde by se navigační symboly měly zobrazit. "Režim tří hvězdiček" znamená, že se použije příkaz `\usebeamertemplate***`.

Ačkoliv by se mohlo zdát, že tyto symboly jsou součástí zápatí, jsou mnohem častěji součástí neviditelného pravého postranního panelu.

Jsou předdefinovány níže uvedené volby šablony:

- `[default]` Srovná navigační symboly horizontálně.
- `[horizontal]` Toto je alias pro výchozí nastavení.
- `[vertical]` Srovná navigační symboly vertikálně.
- `[only frame symbol]` Navigační symbol zobrazí pouze v případě navigačních rámců.

Příklad: Níže uvedený příkaz potlačí všechny navigační symboly:

```
\setbeamertemplate{navigation symbols}{}
```

Uvnitř této šablony jsou užitečné níže uvedené vsuvky:

- `\insertslidenavigationsymbol` Vloží navigační symboly snímku, tj. symboly snímku (obdélník) spolu se šipkami vlevo a vpravo, které jsou opatřeny odkazem typu hyperlink.
- `\insertframenavigationsymbol` Vloží navigační symbol rámce.
- `\insertsubsectionnavigationsymbol` Vloží navigační symbol podod-
dílu.
- `\insertsectionnavigationsymbol` Vloží navigační symbol oddílu.
- `\insertdocnavigationsymbol` Vloží navigační symbol prezentace a (v
případě potřeby) navigační symbol dodatku.
- `\insertbackfindforwardnavigationsymbol` Vloží navigační symbol
pro pohyb vzad, hledání a pohyb vpřed.

5.3.1.2.5 Logo

Logo instalujete níže uvedeným příkazem:

```
\logo{"logo text"}
```

"Logo text" je obvykle příkazem, který zahrne grafiku, avšak může to být jakýkoliv text. Pozice vložení loga se určí podle aktuálního tématu. Tuto pozici nelze (aktuálně) zadat přímo.

Příklad:

```
\pgfdeclareimage[height=0.5cm]{logo}{tu-logo}  
\logo{\pgfuseimage{logo}}
```

Příklad:

```
\logo{\includegraphics[height=0.5cm]{logo.pdf}}
```

Aktuálně tento příkaz nastavuje pouze šablonu loga. V budoucnu však může být implementována sofistikovanější funkcionality.

Beamer-Template/-Color/-Font logo

Tato šablona se používá k vykreslení loga.

Je možné použít níže uvedenou vsuvku k vložení loga na určitou pozici:

- `\insertlogo` vloží logo na aktuální pozici. Tento příkaz má stejný efekt jako `\usebeamertemplate*{logo}`.

5.3.1.2.6 Titul rámce

Titul rámce se zobrazuje viditelně ve vrchní části rámce a je možné jej zadat níže uvedeným příkazem:

```
\frametitle<"overlay specification">["short frame title"]
{"frame title text"}
```

"*Frame title text*" (text titulu rámce) byste měli ukončit tečkou, tvoří-li titul ucelenou větu. V opačném případě by za titulem neměla být tečka. "*Short frame title*" (krátký titul rámce) se obvykle nezobrazuje, avšak je k dispozici pomocí příkazu `\insertshortframetitle`. Volba "*overlay specification*". (specifikace překrytí) je většinou užitečná při potlačení titulu rámce v režimu `article`.

Příklad:

```
\begin{frame}
  \frametitle{Titul rámce je důležitý.}
  \framesubtitle{Podtituly nejsou tak důležité.}

  Obsah rámce.
\end{frame}
```

Používáte-li volbu `allowframebreaks` s aktuálním rámcem, text pokračování (napří "cont.") nebo něco podobného v závislosti na šabloně `frametitle continuation` se automaticky přidá k textu "*frame title text*" a bude oddělen mezerou.

Beamer-Template/-Color/-Font frametitle (titul rámce)

Nadřazený prvek pro volbu `Color/font: titlelike`

Jakmile se má vysázet titul a podtitul, zavolá se tato šablona `frametitle` s nastavenou barvou a fontem `BEAMER`. Tato šablona se nevyvolá při volání příkazů `\frametitle` či `\framesubtitle`. Vyvolá se až po úplném načtení celého rámce. Do té doby se text titulu rámce a podrámce uloží na zvláštním místě. Tímto způsobem se při volání šablony nastaví obě vsuvky správně. Výsledný rámeček `LATEX` se poté šikovně vloží zpět do vrchní části rámce.

Jsou předdefinovány níže uvedené volby šablony:

- [default] ["alignment"] Rámec se vysází pomocí `BEAMER-color frametitle` a `BEAMER-font frametitle`. Podtitul se vloží níže pomocí barvy a fontu `framesubtitle`. Má-li barevný titul `frametitle` pozadí, vloží se za titul obdélník s pozadím, který se roztáhne přes celou šířku rámce. Barva pozadí podtitulu se ignoruje. Volba zarovnání "*alignment*" se předá do prostředí `beamercolorbox`. Užitečné jsou obzvláště volby `left`, `center` a `right`. Ve speciálním případě způsobí volba `right`, že se levý okraj titulu rámce poněkud zvětší, takže se titul rámce přesune více do středu rámce.
- [shadow theme] Tato volba je k dispozici po načtení vnějšího tématu `shadow`. Vykreslí titul rámce nad horizontální stínování mezi barvy pozadí `frametitle` a `frametitle right`. Na tento panel se vloží i podtitul, je-li zadán. Pod panelem se vykreslí "stín".
- [sidebar theme] Tato volba je k dispozici, pokud se načte vnější téma `sidebar` a není-li výška záhlaví nastavena na 0 pt (což je možné provést pomocí volby tématu `sidebar`). Touto volbou se titul rámce vloží dovnitř obdélníkové oblasti, která je součástí záhlaví (použije se "negativní oblast" pro vyzdvížení titulu rámce do této oblasti). Pozadí barevného `frametitle` se nepoužije, v tomto případě se jedná o úlohu šablony záhlaví.
- [smoothbars theme] Tato volba bude k dispozici v případě, že se načte vnější téma `smoothbars`. Vysází titul rámce na barevný panel s barvou pozadí `frametitle`. Vrchní a spodní část panelu se hladce prolne s pozadím nad a pod panelem.
- [smoothtree theme] Stejně jako téma `smoothbars`, pouze pro téma `smoothtree`.

Pro tuto šablonu jsou užitečné níže uvedené příkazy:

- `\insertframetitle` vrátí titul rámce.
- `\insertframesubtitle` vrátí podtitul rámce.
- `\framesubtitle<"overlay specification">{"frame subtitle text" }`

Případný podtitul se zobrazí menším fontem pod hlavním titulem. Stejně jako v případě příkazu `\frametitle` je možné tento příkaz zadat kdekoliv v rámci, jelikož titul rámce se řádně vysází až poté, co se vysází vše ostatní.

Příklad:

```

\begin{frame}
  \frametitle<presentation>{Titul rámce by měl být napsán
velkými písmeny.}
  \framesubtitle{Podtituly mohou být napsány malými písmeny,
tvoří-li ucelené věty.}

  Obsah rámce.
\end{frame}

```

Beamer-Color/-Font framesubtitle (podtitul rámce)

Nadřazený prvek pro volbu Color/font: `frametitle`

Tento element poskytuje podtitulu barvu a font, nikoliv však šablonu. Úlohou šablony `frametitle` je také vysadit podtitul.

Standardně se veškerý materiál snímku vertikálně vystředí. To můžete změnit pomocí níže uvedených voleb třídy:

```
\documentclass[t]{beamer}
```

Umístí text snímků (vertikálně) do vrchní části snímků. To odpovídá vertikálnímu "zarovnání". To můžete potlačit u jednotlivých rámců pomocí voleb `c` nebo `b`.

```
\documentclass[c]{beamer}
```

Umístí text snímků (vertikálně) do středu snímků. Toto je výchozí nastavení. To můžete potlačit u jednotlivých rámců pomocí voleb `t` nebo `b`.

5.3.1.2.7 Pozadí

Každý rámeček má pozadí, které se nachází (jak napovídá jeho název) "za vším ostatním". Pozadí je překvapivě komplexním objektem: Ve třídě BEAMER sestává z plátna pozadí a hlavního pozadí. Plátno pozadí je možné si představit jako rozlehlou plochu, na niž se vše vykresluje (hlavní pozadí a vše ostatní). Standardně je toto plátno velký obdélník, který vyplňuje celý rámeček, jehož barva je pozadím `BEAMER-color background canvas`. Jelikož tato barva dědí z `normal text` (normální text), můžete změnou barvy pozadí normálního textu změnit tuto barvu plátna.

Příklad: Následující příkaz změní barvu pozadí na světle červenou.

```
\setbeamercolor{normal text}{bg=red!20}
```

Plátno nemusí být monochromatické. Místo toho můžete instalovat stínování nebo plátno nastavit jako průhledné. Průhledné plátno se hodí v případě, že chcete snímky začlenit do nějakého jiného dokumentu.

Příklad: Následujícím příkazem nastavíte průhledné plátno:

```
\setbeamercolor{background canvas}{bg=}
```

Beamer-Template/-Color/-Font background canvas (plátno pozadí)

Dědí barvu z: `normal text`

Tato šablona se vloží "za všechny objekty". Šablonou by obvykle měly být nějaké příkazy \LaTeX , které vytvoří obdélník o výšce `\paperheight` a šířce `\paperwidth`.

Jsou předdefinovány níže uvedené volby šablony:

- `[default]` instaluje rozměrný obdélník s barvou pozadí. Je-li pozadí prázdné, plátno bude "průhledné". Jelikož plátno pozadí dědí z `normal text` (normální text), můžete změnou barvy pozadí `BEAMER-color normal text` změnit tuto barvu výchozího plátna. Chcete-li však vytvořit průhledné plátno, vytvořte pouze prázdné pozadí plátna. Pochechte pozadí normálního textu bílé.
- `[vertical shading] ["color options"]` instaluje vertikálně stínované pozadí. Používejte opatrně: Stínované pozadí často odvádí pozornost! Je možné zadat následující volby "*color options*":
 - `top="color"` specifikuje barvu ve vrchní části stránky. Ve výchozím nastavení se používá 25% z přední části palety `BEAMER-color palette`.
 - `bottom="color"` specifikuje barvu spodní části stránky. Ve výchozím nastavení je to pozadí normálního textu v okamžiku volání tohoto příkazu.
 - `middle="color"` specifikuje barvu střední části postranního stránky. Pokud tuto volbu zadáte, barva stínování se mění při přechodu ze spodní části do střední části a pak ze střední části do vrchní části.
 - `midpoint="factor"` specifikuje místo na stránce, na němž se použije barva pro střední část stránky. Faktor o velikosti 0 je spodní částí stránky, faktor 1 je její vrchní částí. Výchozí hodnotou je 0.5, což je střed.

Hlavní pozadí se vykreslí na vrchní vrstvu plátna pozadí. Můžete to použít například k přidání mřížky či velkého obrázku na pozadí každého rámce.

Beamer-Template/-Color/-Font background (pozadí)

Dědí barvu z: `background canvas`

Tato šablona se vloží za všechny objekty, avšak do nejvrchnější vrstvy plátna pozadí. Používejte ji pro obrázky či mřížky nebo jiné objekty, které nutně nezaplňují celé pozadí. Při vysázení této šablony se nastaví `BEAMER-color` a `{font background}`.

Jsou předdefinovány níže uvedené volby šablony:

- `[default]` je prázdná.
- `[grid] ["grid options"]` umístí na pozadí mřížku. Je možné zadat následující volby "*grid options*":
 - `step="dimension"` specifikuje vzdálenost mezi čarami mřížky. Výchozí hodnotou je 0.5 cm.
 - `color="color"` specifikuje barvu čar mřížky. Výchozí hodnotou je 10% pozadí.

5.3.1.3 Velikost okrajů

"Velikost papíru" prezentace BEAMER je zafixována na hodnotě 128 mm krát 96 mm. Poměr stran této velikosti je 4:3, což je přesně velikost, s níž pracuje většina projektorů. Je úlohou prezentačního programu (např. `acoread` nebo `xpdf`) zobrazit snímky v plné velikosti. Hlavní výhodou používání malé "velikosti papíru" je to, že můžete používat všechny fonty v jejich přirozených velikostech. Obzvláště obrázky s popisky o velikosti 11 p budou mít při prezentaci rozumnou velikost.

"Velikost papíru" byste neměli měnit. Můžete však změnit velikost levého a pravého okraje, která je standardně nastavena na 1 cm. Změnit byste je měli následujícím příkazem:

```
\setbeamersize{"options" }
```

Je možné zadat následující volby "*options*" :

- `text margin left="TEX dimension"` nastaví nový levý okraj. Hodnota se nevztahuje na levý postranní panel. Je to tudíž vzdálenost mezi pravým okrajem levého postranního panelu a levým okrajem textu.
- `text margin right="TEX dimension"` nastaví nový pravý okraj.
- `sidebar width left="TEX dimension"` nastaví velikost levého postranního panelu. Aktuálně byste tento příkaz měli vložit předtím, než se instaluje stínování plátna postranního panelu.

- `sidebar width right="TEX dimension"` nastaví velikost pravého postranního panelu.
- `description width="TEX dimension"` nastaví výchozí šířku popisek, viz oddíl 12.1.
- `description width of="text"` nastaví výchozí šířku popisek na šířku textu "text", viz oddíl 12.1.
- `mini frame size="TEX dimension"` nastaví velikost mini rámců v navigační liště. Zobrazí-li se vedle sebe dvě ikony mini rámců, jejich levé koncové body od sebe budou mít vzdálenost "*TEX dimension*".
- `mini frame offset="TEX dimension"` nastaví dodatečný vertikální posun, který se přidá k velikosti mini frame při jejich vertikálním uspořádání.

5.3.1.4 Omezení počtu snímků v rámci

Počet snímků daného rámce se vypočítá automaticky. Je-li nejvyšší počet snímků ve specifikaci překrytí uvnitř rámce 4, vytvoří se čtyři snímky (navzdory faktu, že specifikace `{4-}` by mohla naznačovat, že je možné přidat více než čtyři snímky).

Počet snímků daného rámce můžete specifikovat i "ručně". To provedete předáním příkazu `\frame` do specifikace překrytí. Rámec bude obsahovat pouze snímky zadané tímto argumentem.

Podívejte se na níže uvedený příklad.

```
\begin{frame}<1-2,4->
  Toto snímek číslo \only<1>{1}\only<2>{2}\only<3>{3}%
  \only<4>{4}\only<5>{5}.
\end{frame}
```

Tento příkaz vytvoří rámec se čtyřmi snímky. První snímek bude obsahovat text "Toto je snímek číslo 1", druhý "Toto je snímek číslo 2", třetí "Toto je snímek číslo 4" a čtvrtý "Toto je snímek číslo 5".

Užitečnou specifikací je prostě `{0}`. Ta způsobí, že rámec nebude mít vůbec žádné snímky. Například `\begin{frame}<handout:0>` způsobí, že se rámec potlačí ve verzi prospekt, avšak ve všech ostatních verzích se zobrazí. Další užitečnou specifikací je `<beamer>`. Ta způsobí, že se rámec zobrazí normálně v režimu projekce, avšak ve všech dalších verzích se potlačí.

5.3.2 Vytvoření překrytí

5.3.2.1 Pauzovací příkazy

Pauzovací příkaz nabízí jednoduchý, avšak nepříliš flexibilní způsob vytvoření rámců, které se odkrývají postupně. Vložíte-li někde do rámce příkaz `\pause`, na prvním snímku se zobrazí pouze text před příkazem `\pause`. Na druhém snímku se zobrazí všechno před druhým příkazem `\pause` atd. Příkaz `\pause` můžete použít i uvnitř prostředí. Jeho efekt bude trvat i po proběhnutí prostředí. Pokud to však s užíváním příkazu `\pause` uvnitř vloženého prostředí přeženete, nemusíte se dočkat požadovaných výsledků.

Daleko jemnějšího stupně ovládní zobrazených objektů lze na jednotlivých snímcích dosáhnout pomocí specifikací překrytí. Další informace najdete v níže uvedených částech. Ve spoustě jednoduchých případů je však příkaz `\pause` dostatečný. Efekt příkazu `\pause` trvá až do dalšího příkazu `\pause`, `\onslide` nebo dosažení konce rámce.

```

\begin{frame}
  \begin{itemize}
    \item
      Zobrazuje se od prvního snímku.
    \pause
    \item
      Zobrazuje se od druhého snímku.
    \begin{itemize}
      \item
        Zobrazuje se od druhého snímku.
      \pause
      \item
        Zobrazuje se od třetího snímku.
    \end{itemize}
    \item
      Zobrazuje se od třetího snímku.
    \pause
    \item
      Zobrazuje se od čtvrtého snímku.
    \end{itemize}

    Zobrazuje se od čtvrtého snímku.

    \begin{itemize}
      \onslide

```

```

\item
  Zobrazuje se od prvního snímku.
\pause
\item
  Zobrazuje se od pátého snímku.
\end{itemize}
\end{frame}

\pause["number" ]

```

Text následující po tomto příkazu se zobrazuje pouze od následujícího snímku. Zadáte-li volitelný parametr *"number"*, zobrazuje se text od snímku specifikovaného číslem *"number"*. Zadáte-li volitelný parametr *"number"*, nastaví se počítadlo `beamerpauses` na toto číslo. Tento příkaz používá interně příkaz `\onslide`. Tento příkaz nepracuje uvnitř prostředí `amsmath`, jako je například `align`, protože tato prostředí provádějí velmi záludné akce.

Příklad:

```

\begin{frame}
  \begin{itemize}
    \item
      A
    \pause
    \item
      B
    \pause
    \item
      C
  \end{itemize}
\end{frame}

```

5.3.2.2 Obecný koncept specifikací překrytí

Přístup, který většina prezentačních tříd zaujímá k překrytím, je do jisté míry podobný výše uvedenému příkazu `\pause`. Tyto příkazy přebírají na vstupu číslo určitého snímku a na snímku určitým způsobem ovlivní text, jenž za tímto příkazem následuje. Například příkaz PROSPER'S `\FromSlide{2}` způsobí, že se veškerý text následující za tímto příkazem zobrazí od druhého snímku.

Třída BEAMER používá jiný přístup (ačkoliv výše zmíněný příklad je také k dispozici: `\onslide<2->` má stejný účinek jako `\FromSlide{2}`

s výjimkou toho, že příkaz `\onslide` překračuje prostředí. Stejně tak je příkaz `\pause` interně namapován na příkaz s příslušnou specifikací překrytí). Princip spočívá v tom, přidat specifikace překrytí k určitým příkazům. Tyto specifikace jsou vždy zadány ve špičatých závorkách a následují po příkazu co nejdříve. V některých případech však třída BEAMER povoluje i trochu pozdější zadání specifikace překrytí. V nejjednodušším případě obsahuje daná specifikace pouze číslo. Příkaz, po němž následuje specifikací překrytí, bude mít efekt pouze na snímek (-ky), které jsou uvedeny ve specifikaci. Co přesně znamená mít efekt záleží na daném příkazu. Podívejte se na níže uvedený příklad.

```
\begin{frame}
  \textbf{Tento řádek je tučně na všech třech snímcích.}
  \textbf<2>{Tento řádek je tučně pouze na druhém snímku.}
  \textbf<3>{Tento řádek je tučně pouze na třetím snímku.}
\end{frame}
```

U příkazu `\textbf` způsobí specifikace překrytí, že se text vysází tučně pouze na zadaných snímcích. Na všech ostatních snímcích se text vysází normálním fontem. Níže uvádíme druhý příklad:

```
\begin{frame}
  \only<1>{Tento řádek se vloží pouze na snímku 1.}
  \only<2>{Tento řádek se vloží pouze na snímku 2.}
\end{frame}
```

Příkaz `\only`, který zavádí třída BEAMER, obvykle prostě vloží svůj parametr do aktuálního snímku. Je-li však přítomna specifikace překrytí, příkaz svůj parametr na neuvedených snímcích zahodí.

Specifikace překrytí je možné zapsat pouze za některé příkazy, nikoliv za každý příkaz. V další části si vysvětlíme, jaké příkazy je možné používat a jaké jsou jejich účinky. Je však docela jednoduché předefinovat stávající příkaz tak, že reaguje na specifikaci překrytí. Další informace najdete také v části 9.3.

Syntaxe (základních) specifikací překrytí je následovná: Jsou to seznamy snímků a rozsahů oddělené čárkami. Rozsahy jsou zadány takto: 2-5, což znamená od snímku dvě po snímek pět. Počátek nebo konec určitého rozsahu je možné vynechat. Například 3- znamená snímky tři, čtyři, pět atd. a -5 znamená to samé jako 1-5. Komplikovaným příkladem je -3,6-8,10,12-15. V tomto případě se vyberou snímky 1, 2, 3, 6, 7, 8, 10, 12, 13, 14 a 15.

5.3.2.3 Příkazy se specifikacemi překrytí

Přidáte-li k níže uvedeným příkazům specifikaci překrytí, budou tyto příkazy jednoduše ignorovány na snímcích, jež nejsou ve specifikaci zahrnuty: `\textbf`, `\textit`, `\textsl`, `\textrm`, `\textsf`, `\color`, `\alert`, `\structure`. Přebírá-li určitý příkaz několik argumentů, například `\color`, specifikace by měla ihned následovat po příkazu tak, jako v následujícím příkazu (k tomuto pravidlu však existují výjimky):

```
\begin{frame}
  \color<2-3>[rgb]{1,0,0} Tento text je červený na snímcích
  2 a 3. Jinak je černý.
\end{frame}
```

U níže uvedených příkazů má specifikace překrytí zvláštní efekt:

```
\onslide"modifier"<"overlay specification">{"text"}
```

Chování tohoto příkazu závisí na tom, zda zadáte volitelný argument *text* (všimněte si, že volitelný argument se zadává v normálních závorkách, nikoliv v hranatých závorkách). Je-li volitelný argument přítomen, může být zadán buď jako +, nebo jako *.

Není-li zadán žádný *text*, dojde k následujícím akcím: Veškerý text následující po tomto příkazu bude zobrazen (odkrytý) pouze na zadaných snímcích. I na nespecifikovaných snímcích text nadále zabírá místo. Nejsou-li zadány žádné snímky, následující text se zobrazí vždy. Tento příkaz nemusíte volat ve stejné skupině T_EX, jeho účinky překračují skupiny bloků. Tento příkaz má však různé účinky uvnitř prostředí overprint. Další informace najdete v popisu prostředí overprint. Je-li *modifier* (modifikátor) +, skrytý text se nezpracuje jako zakrytý, nýbrž jako neviditelný. Rozdíl je stejný jako mezi příkazy `\uncover` a `\visible`. Modifikátor * nelze zadat, není-li zadán argument *text*.

Příklad:

```
\begin{frame}
  Zobrazuje se na prvním snímku.
  \onslide<2-3>
  Zobrazuje se na druhém a třetím snímku.
  \begin{itemize}
  \item
    Stále se zobrazuje na druhém a třetím snímku.
  \onslide+<4->
```

```

\item
  Zobrazuje se od 4. snímku.
\end{itemize}
Zobrazuje se od 4. snímku.
\onslide
  Zobrazuje se na všech snímcích.
\end{frame}

```

Je-li přítomen argument *text*, příkaz `\onslide` (bez modifikátoru *width*) se namapuje na `\uncover`, `\onslide+` se namapuje na `\visible` a `\onslide*` se mapuje na `\only`.

Příklad:

```

\begin{frame}
  \onslide<1>{Stejný efekt jako následující příkaz.}
  \uncover<1>{Stejný efekt jako předchozí příkaz.}

  \onslide+<2>{Stejný efekt jako následující příkaz.}
  \visible<2>{Stejný efekt jako předchozí příkaz.}

  \onslide*<3>{Stejný efekt jako následující příkaz.}
  \only<3>{Stejný efekt jako předchozí příkaz.}
\end{frame}

```

```

\only<"overlay specification" >{"text" }<"overlay specifica-
tion" >

```

Je-li přítomna některá specifikace překrytí *overlay specification* (ačkoliv může být přítomna pouze jedna), vloží se *text* pouze do zadaných snímků. U ostatních snímků se text jednoduše zahodí. Především pak nezabírá žádné místo.

Příklad:

```

\only<3->{Text se vkládá od snímku 3.}

```

Jelikož je možné zadat specifikaci překrytí i po zadání textu, můžete často prostě použít příkaz `\only`, po němž začnou ostatní příkazy reagovat na specifikaci překrytí:

Příklad:

```

\newcommand{\myblue}{\only{\color{blue}}}
\begin{frame}
  \myblue<2> Tento text je modrý pouze na snímku 2.
\end{frame}

\uncover<"options" >{"text" }

```

Je-li zadána specifikace překrytí "*overlay specification*", zobrazí se "*text*" ("odkrytý") pouze na zadaných snímcích. Na dalších snímcích zabírá text i nadále místo. Text vysází, avšak nezobrazuje se nebo se zobrazuje průhledně.

Příklad:

```

\uncover<3->{Text se zobrazuje od snímku 3.}

\visible<"overlay specification" >{"text" }

```

Tento příkaz provádí takřka stejnou činnost jako `\uncover`. Jediným rozdílem je, že pokud se text nemá zobrazit, nezobrazí se nikdy průhledně – nezobrazuje se totiž vůbec. Na tento příkaz tedy nemají vliv žádná nastavení průhlednosti.

Příklad:

```

\visible<2->{Text zobrazený od snímku 2.}

\invisible<"overlay specification" >{"text" }

```

Tento příkaz má opačný účinek než `\visible`.

Příklad:

```

\invisible<2->{Text zobrazený od snímku 3.}

\alt<"overlay specification" >{"default text" }{"alternative
text" }<"overlay specification">

```

Je možné zadat pouze jednu specifikaci překrytí "*overlay specification*". Výchozí text se zobrazí na zadaných snímcích, v opačném případě se zobrazí alternativní text. Specifikace musí být zadána vždy.

Příklad:

```

\alt<2>{Na snímku 2}{Není na snímku 2.}

```

Je užitečné zadat specifikaci překrytí na konec v případě, že se příkaz používá uvnitř jiných příkazů.

Příklad: Definice příkazu `\uncover`:

```
\newcommand{\uncover}{\alt{\@firstofone}{\makeinvisible}}

\temporal<"overlay specification" >{"before slide text" }
{"default text" }{"after slide text" }
```

Tento příkaz přepíná mezi třemi různými texty v závislosti na tom, zda je aktuální snímek dočasně před zadanými snímky nebo je jedním ze zadaných snímků nebo na něj dojde až po nich. Není-li specifikace překrytí intervalem (tj. má-li v sobě díru), považuje se díra za součást předchozích snímků.

Příklad:

```
\temporal<3-4>{Zobrazeno na 1, 2}{Zobrazeno na 3, 4}{Zobra-
zeno 5, 6, 7, ...}
\temporal<3,5>{Zobrazeno na 1, 2, 4}{Zobrazeno na 3, 5}{Zob-
razeno 6, 7, 8, ...}
```

Příkaz `\temporal` je možné použít například takto:

Příklad:

```
\def\colorize<#1>{%
\temporal<#1>{\color{red!50}}{\color{black}}{\color
{black!50}}}
```

```
\begin{frame}
\begin{itemize}
\colorize<1> \item První položka.
\colorize<2> \item Druhá položka.
\colorize<3> \item Třetí položka.
\colorize<4> \item Čtvrtá položka.
\end{itemize}
\end{frame}
```

```
\item<"alert specification" >["item label" ]<"alert specifica-
tion" >
```

Příklad:

```

\begin{frame}
  \begin{itemize}
    \item<1-> První bod, zobrazuje se na všech snímcích.
    \item<2-> Druhý bod, zobrazuje se od snímku 2.
    \item<2-> Třetí bod, zobrazuje se od snímku 2.
    \item<3-> Čtvrtý bod, zobrazuje se na snímku 3.
  \end{itemize}
\end{frame}

```

```

\begin{frame}
  \begin{enumerate}
    \item<3-| alert@3>[0.] Nultý bod, zobrazuje se úplně
na konci.
    \item<1-| alert@1> První a hlavní bod.
    \item<2-| alert@2> Druhý bod.
  \end{enumerate}
\end{frame}

```

```
\label<"overlay specification">{"label name"}
```

Je-li zadána specifikace překrytí "*overlay specification*", vloží se popisec pouze do zadaného snímku. Vložíte-li popisec do více než jednoho snímku, zobrazí se upozornění "multiple labels" (více popisec). Není-li však zadána žádná specifikace překrytí, nastaví se specifikace pouze na hodnotu "1", a popisec se tedy vloží pouze na první snímek. Takové chování je obvykle vyžadováno, protože v podstatě nezáleží na tom, do kterého snímku se popisec vkládá s výjimkou případu, kdy použijete příkaz `\only` nebo si přejete použít takový popisec jako cíl hyperlinkového odkazu. Následně je nutné specifikovat snímek. Popisky je možné využívat jako cíl hyperlinkových odkazů. Chcete-li určitý rámeček opatřit popisky, je pohodlné použít volbu `label="name"` prostředí `frame`. Tím však dojde k tomu, že se celý rámeček uchová v paměti až do konce kompilace, což může představovat problém.

Příklad:

```

\begin{frame}
  \begin{align}
    a~&= b + c \label{first} \\ % není nutná žádná specif.
    c &= d + e \label{second} \\ % není nutná žádná specif.
  \end{align}

```

```
Nějaký text, \uncover<2>{další text.}
```

```
\only<3>{Nyní je nutná specifikace.\label<3>{mylabel}}
\end{frame}
```

5.3.2.4 Prostředí se specifikacemi překrytí

Prostředí můžete také vybavit specifikacemi překrytí. Zadáte-li specifikaci překrytí, celé prostředí se odkryje pouze na zadaných snímcích. To je užitečné v případě, že chcete obsah zobrazovat postupně, třeba jako v níže uvedeném příkladu.

```
\begin{frame}
  \frametitle{Teorém o nekonečných množinách}

  \begin{theorem}<1->
    Existují nekonečné množiny.
  \end{theorem}

  \begin{proof}<3->
    To vyplývá z postulátu o nekonečnosti.
  \end{proof}

  \begin{example}<2->
    Množina přirozených čísel je nekonečná.
  \end{example}
\end{frame}
```

V daném příkladu obsahuje první snímek pouze teorém, na druhém snímku se přidá příklad a na třetím snímku se zobrazí důkaz. Pro každý základní příkaz `\only`, `\alt`, `\visible`, `\uncover` a `\invisible` existují "verze pro prostředí" `onlyenv`, `altenv`, `visibleenv`, `uncoverenv` a `invisibleenv`. S výjimkou `altenv` a `onlyenv` provádějí tato prostředí stejné akce jako příkazy.

```
\begin{onlyenv}<"overlay specification">
  "environment contents"
\end{onlyenv}
```

Je-li zadána specifikace překrytí "*overlay specification*", vloží se obsah prostředí do textu pouze na specifikovaných snímcích. Rozdíl mezi tímto příkazem a příkazem `\only` je v tom, že se text v podstatě vysází do rámečku, který se následně zahodí, zatímco `\only` zahodí obsah okamžitě. Jestliže text nelze vysázet, příkaz `onlyenv` vyvolá chybu, zatímco příkaz `\only` nikoliv.

Příklad:

```

\begin{frame}
  Tento řádek se vždy zobrazí.
  \begin{onlyenv}<2>
    Tento řádek se vloží do snímku 2.
  \end{onlyenv}
\end{frame}

\begin{altenv}<"overlay specification">{"begin text"}{"end
text"}{"alternate begin text"}{"alternate end text"}<"overlay
specification">
  "environment contents"
\end{altenv}

```

Je možné zadat pouze jednu specifikaci překrytí *"overlay specification"*. Na specifikovaných snímcích se *"begin text"* vloží na začátek prostředí a *"end text"* na jeho konec. Na všech ostatních snímcích se použijí *"alternate begin"* a *"alternate end"*.

Příklad:

```

\begin{frame}
  Toto
  \begin{altenv}<2>{( )}{[ ]}
    slovo
  \end{uncoverenv}
  je v kulatých závorkách na snímku 2 a v hranatých závorkách
  na snímku 1.
\end{frame}

```

5.3.2.5 Dynamická změna textu nebo obrázků

Někdy je nutné, aby se určité části rámce na jednotlivých snímcích dynamicky měnily. Na jednotlivých snímcích daného rámce by se v této oblasti měly zobrazit různé informace. Takového efektu můžete dosáhnout dynamickou změnou textu tak, že zadáte seznam příkazů `\only`:

```

\only<1>{Úvodní text.}
\only<2>{Původní text nahradíme tímto textem na druhém snímku.}
\only<3>{Opětovná náhrada na třetím snímku.}

```

Problém tohoto přístupu spočívá v tom, že se mohou objevit drobné, avšak iritující rozdíly ve výšce řádků, což vede k tomu, že celý rámeček mezi jednotlivými snímky "poskakuje". Tento problém se ještě více prohloubí, roztáhne-li se nahrazovaný text na několika řádcích. Tento problém můžete vyřešit použitím dvou prostředí: `overlayarea` a `overprint`. První je flexibilnější, avšak poněkud méně uživatelsky přívětivý.

```
\begin{"overlayarea"}{"area width"}{"area height"}
  "environment contents"
\end{overlayarea}
```

Vše, co se v prostředí nachází, bude umístěno do obdélníkové oblasti o zadané velikosti. Oblast bude mít stejnou velikost na všech snímcích rámeček, bez ohledu na vlastní obsah.

Příklad:

```
\begin{overlayarea}{\textwidth}{3cm}
  \only<1>{Nějaký text pro první snímek.\\Možná několik
řádů dlouhý.}
  \only<2>{Náhradní text na druhém snímku.}
\end{overlayarea}
```

```
\begin{overprint}["area width"]
  "environment contents"
\end{overprint}
```

Šířka oblasti "*area width*" se standardně nastaví na šířku textu. Uvnitř prostředí pomocí příkazů `\onslide` zadejte, co se má zobrazit v tomto prostředí na jednotlivých snímcích. Příkazy `\onslide` se používají podobně jako příkazy `\item`. Vše, co se v prostředí nachází, bude umístěno do obdélníkové oblasti o zadané šířce. Výška a hloubka oblasti se zvolí tak, aby se do ní vešel největší obsah dané oblasti. Specifikace překrytí příkazů `\onslide` musí být nespojitá. To může vést k problémům v případě prospektů, jelikož jsou v nich všechny specifikace překrytí standardně nastaveny na 1. Použijete-li volbu `handout`, můžete s výjimkou jednoho zakázat všechny ostatní příkazy `\onslide` nastavením na hodnotu 0.

Příklad:

```
\begin{overprint}
  \onslide<1|handout:1>
```



```

    Nějaký text prvního snímku.\\
    Text může být několik řádků dlouhý.
\onslide<2| handout:0>
    Náhradní text na druhém snímku. Potlačeno pro účely
prospektu.
\end{overprint}

```

Dynamické změny budete potřebovat i v případě, pracujete-li například s řadou obrázků pojmenovaných `first.pdf`, `second.pdf` a `third.pdf`, na nichž jsou zobrazeny fáze určitého procesu. Chcete-li vytvořit rámeček, který zobrazuje tyto obrázky na různých snímcích, můžete použít následující kód:

```

\begin{frame}
  \frametitle{Proces o třech fázích}

  \includegraphics<1>{first.pdf}
  \includegraphics<2>{second.pdf}
  \includegraphics<3>{third.pdf}
\end{frame}

```

Výše uvedený kód využívá faktu, že třída BEAMER nastaví příkaz `\includegraphics` tak, aby reagoval na specifikaci překrytí. Funguje to dobře v případě, že jednotlivé soubory `.PDF` obsahují kompletní grafiku, která se má zobrazit. Některé programy (například `xfig`) však občas vyprodukují sadu s grafikou, jejíž jednotlivé soubory obsahují pouze dodatečné grafické prvky, které se mají zobrazit na dalším snímku. V tomto případě je nutné první grafiku zobrazit nikoliv na překrytí 1, nýbrž od překrytí 1 dále, atd. Toho je snadné dosáhnout změnou specifikace překrytí z `<1>` na `<1->`, avšak jednotlivé grafické prvky je nutné zobrazit i ve vrstvách poskládaných na sebe. Toho lze snadno dosáhnout použitím příkazu `TeX`

```

\begin{frame}
  \frametitle{Proces o třech fázích}

  \includegraphics<1->{first.pdf}%
  \llap{\includegraphics<2->{second.pdf}}%
  \llap{\includegraphics<3->{third.pdf}}
\end{frame}

```

nebo takto:

```

\begin{frame}
  \frametitle{Proces o třech fázích}

```

```

\includegraphics{first.pdf}%
\pause%
\llap{\includegraphics{second.pdf}}%
\pause%
\llap{\includegraphics{third.pdf}}
\end{frame}

```

Pohodlnějším způsobem je použít příkaz `\multiinclude`.

5.3.2.6 Pokročilé specifikace překrytí

5.3.2.6.1 Vytváření příkazů a prostředí, které reagují na specifikace překrytí

Tato část vysvětluje způsob, jak definovat nové příkazy, jež reagují na specifikace překrytí. Dále vysvětluje, jak správně nastavit počítadla, která zvyšují svou hodnotu při přechodu z rámce na rámeček (například číslování rovnic), avšak nikoliv při přechodu ze snímku na snímek. Nemáte-li zájem si napsat vlastní rozšíření třídy BEAMER, můžete tuto část přeskočit.

Třída BEAMER rozšiřuje syntaxi standardního příkazu `LATEX\newcommand`:

```

\newcommand<>{"command name"}["argument number"]["default optional value"]{"text"}

```

Deklaruje nový příkaz s názvem *command name*. Parametr *text* by měl obsahovat tělo tohoto příkazu a může obsahovat výskyt parametrů, například *#number*. V tomto případě může být hodnota *number* mezi 1 a *argument number* + 1. Dodatečně povolený argument je specifikace překrytí.

Použijete-li *command name*, zkontroluje se tolik argumentů, kolik je zadáno volbou *argument number*. Při jejich kontrole se ověří specifikace překrytí, kterou je možné zadat mezi kterýmikoliv dvěma argumenty, před prvním argumentem nebo za posledním argumentem. Nalezne-li se specifikace překrytí, například `<3>`, zavolá se *text* s počtem argumentů 1 až *argument number*, které se nastaví na normální argumenty, a počtem argumentů *argument number* + 1 nastavených na `<3>` (včetně špičatých závorek). Není-li nalezena žádná specifikace překrytí, extra argument bude prázdný.

Zadáte-li výchozí volitelnou hodnotu *default optional value*, bude první argument příkazu *command name* volitelný. Nezádáte-li žádný volitelný argument v hranatých závorkách, použije se výchozí volitelná hodnota *default optional value*.

Příklad: Následující příkaz vysází svůj argument červeně na zadaných snímcích:

```
\newcommand<>\makered}[1]{\color#2{red}#1}}
```

Příklad: Definice příkazu `\emph` ve třídě BEAMER:

```
\newcommand<>\emph}[1]{\only#2{\itshape}#1}}
```

Příklad: Definice příkazu `\transdissolve` ve třídě BEAMER (příkaz `\beamer@dotrans` především předává svůj argument do `hyperref`):

```
\newcommand<>\transdissolve}[1] []{\only#2{\beamer@dotrans
[#1]{Dissolve}}}
```

```
\renewcommand<>{"existing command name"}["argument number"]
["default optional value"] {"text"}
```

Stejně jako v případě příkazu `\newcommand<>` opětovně deklaruje určitý příkaz, který již existuje. Uvnitř volby *text* můžete nadále přistupovat k původním definicím pomocí příkazu `\beameroriginal`, viz příklad.

Příklad: Tento příkaz se používá ve třídě BEAMER k nastavení volby `\hyperlink` tak, aby pracovala se specifikací překrytí.

```
\renewcommand<>\hyperlink}[2]{\only#3{\beameroriginal{\hy-
perlink}{#1}{#2}}}
```

```
\newenvironment<>{"environment name"}["argument number"]
["default optional value"] {"begin text"} {"end text"}
```

Deklaruje nové prostředí, které reaguje na specifikace překrytí. Narazí-li program na toto prostředí, použije se pro zpracování argumentu a specifikaci překrytí stejný algoritmus jako pro `\newcommand<>`. *End text* nemusí obsahovat žádné argumenty, například `#1`. Především pak nemáte přístup ke specifikaci překrytí. V tomto případě je obvykle vhodné v *begin text* použít prostředí `altenv`.

Příklad: Deklarujte si svým vlastní blok akcí:

```
\newenvironment<>\myboldblock}[1]{%
  \begin{actionenv}#2%
    \textbf{#1}
  \par}
```

```

{\par%
\end{actionenv}}

\begin{frame}
  \begin{myboldblock}<2>
    Tento teorém se zobrazuje pouze na druhém snímku.
  \end{myboldblock}
\end{frame}

```

Příklad: Text v následujícím prostředí je na nespecifikovaných snímcích běžně napsán tučně a kurzívou.

```

\newenvironment<>{boldornormal}
  {\begin{altenv}#1
   {\begin{bfseries}}{\end{bfseries}}
   {}{}}
{\end{altenv}}

```

Jelikož příkaz `altenv` přijímá svůj argument i na konci, stejného efektu jste mohli dosáhnout pouhým zapsáním

```

\newenvironment{boldornormal}
  {\begin{altenv}
   {\begin{bfseries}}{\end{bfseries}}
   {}{}}
{\end{altenv}}

```

```

\renewenvironment<>{"existing environment name" }["argument
number"] ["default optional value"] {"begin text"} {"end text"}

```

Redefinuje stávající prostředí. Původní prostředí je nadále k dispozici pod názvem original "*existing environment name*".

Příklad:

```

\renewenvironment<>{verse}
  {\begin{actionenv}#1\begin{originalverse}}
  {\end{originalverse}\end{actionenv}}

```

Níže uvedené dva příkazy je možné použít tak, aby se určité počítadlo automaticky resetovalo na následujících snímcích daného rámce. To je nezbytné například u označení počtu rovnic. Pravděpodobně si budete přát, aby se toto počítadlo zvyšovalo při přechodu z rámce na rámeček, avšak určitě ne

z překrytého snímku na překrytý snímek. Co se týče počítadel rovnic a počítadel poznámek pod čarou (poznámky pod čarou by se neměly používat), byly tyto příkazy již vyvolány.

```
\resetcounteronoverlays{"counter name"}
```

Po zavolání tohoto příkazu bude hodnota zadaného počítadla na všech snímcích každého rámce stejná.

Příklad:

```
\resetcounteronoverlays{equation}

\resetcountonoverlays{"count register name"}
```

Stejně jako `\resetcounteronoverlays`. Tento příkaz byste však měli používat s počítadly vytvořenými pomocí primitivního příkazu `TEX\newcount` namísto příkazu `LATEX\definecounter`.

Příklad:

```
\newcount\mycount
\resetcountonoverlays{mycount}
```

5.3.2.6.2 Specifikace režimů

V určitých případech si budete možná přát odlišné specifikace překrytí, které na určitý příkaz použijete v různých režimech. Například budete chtít, aby se určitý text během prezentace zobrazil pouze od třetího snímku, avšak v prospektech, které rozdáte publiku, žádný druhý snímek nebude, a text by se proto měl objevit již na druhém snímku. V takovém případě byste mohli zapsat

```
\only<3| handout:2>{Nějaký text}
```

Vertikální čára, po níž musí následovat (bílá) mezera, odděluje dvě různé specifikace 3 a prospekt `handout: 2`. Zápisem názvu režimu před středník zadáte, že následující specifikace se použije pouze na tento zadaný režim. Není-li zadán žádný režim, tak jako v případě 3, automaticky se přidá režim `beamer`. Z tohoto důvodu platí, že zapíšete-li `\only<3>{Text}` a jste v režimu prospektu, text se zobrazí na všech snímcích, jelikož neexistuje žádné omezení pro prospekty a 3 je vždy to samé jako `beamer:3`.

Je také možné zadat specifikaci překrytí, která obsahuje pouze název režimu (nebo několik názvů oddělených vertikální čarou):

```
\only<article>{Tento text se zobrazí pouze v režimu article}
```

Specifikace překrytí, která neobsahuje žádná čísla snímků, se nazývá (čistá) specifikace režimu. Zadáte-li specifikaci režimu, všechny režimy, které nejsou zadány, se automaticky potlačí. Výraz `<beamer:1->` tedy znamená "na všech snímcích v režimu BEAMER a také na všech snímcích ve všech ostatních režimech, jelikož pro ně není zadáno nic zvláštního", zatímco `<beamer>` znamená "na všech snímcích v režimu BEAMER, ale na žádných jiných snímcích. Specifikace režimů je možné použít i mimo rámce. Viz níže uvedené příklady:

```
\section<presentation>{Tento oddíl existuje pouze v režimech prezentace}
```

```
\section<article>{Tento oddíl existuje jen v režimu article}
```

Můžete také smíchat čisté specifikace režimů a specifikace překrytí, ačkoliv to může být poněkud matoucí:

```
\only<article| beamer:1>{Hádanka}
```

Tím vložíte text Hádanka v režimu article na první snímek daného rámce v režimu beamer, avšak nikoliv v režimu handout nebo trans.

Jako by toto vše nebylo ještě dostatečně komplikované, existuje další režim, který se chová zvláštním způsobem: režim `second`. Pro tento režim platí zvláštní pravidlo: Specifikace překrytí režimu beamer platí i pro režim `second` (avšak nikoliv obráceně). Pracujeme-li tedy v režimu `second`, specifikace `<second:2>` znamená "na snímku 2" a `<beamer:2>` také znamená "na snímku 2". Chcete-li získat snímek, který je vysázen v režimu beamer, avšak nikoliv v režimu second, můžete použít `<second:0>`.

5.3.2.6.3 Specifikace akcí

Některé příkazy, které pracují se specifikacemi překrytí, nejenže nejsou schopny zpracovat běžné specifikace překrytí, ale ani tzv. specifikace akcí. Ve specifikaci akcí mají čísla snímků a rozsahy předponu "action"@, kde "action" je název určité akce, která se provede na zadaných snímcích:

```
\item<3-| alert@3> Zobrazuje se od snímku 3, aktivováno na snímku 3.
```

Ve výše uvedeném příkladu odkryje příkaz `\item`, díky němuž můžete specifikovat akce, text položky od snímku tři a dále tuto položku aktivuje přesně na snímku 3. Ne všechny příkazy mohou přebírat specifikaci akcí. Aktuálně je zpracovávají pouze příkazy `\item` (i když aktuálně nikoliv v režimu `article`), `\action`, příkaz pro práci s prostředím `actionenv`, a bloková prostředí (například `block` nebo `theorem`). Standardně jsou k dispozici následující akce:

- `alert` aktivuje položku nebo blok.
- `uncover` odkryje položku nebo blok (toto je výchozí nastavení, není-li zadána žádná akce).
- `only` vloží celou položku nebo blok pouze do zadaných snímků.
- `visible` zviditelní text pouze na zadaných snímcích (rozdíl mezi `uncover` a `visible` je stejný jako mezi `\uncover` a `\visible`).
- `invisible` zneviditelní text na zadaných snímcích.

Zbytek tohoto oddílu vysvětluje způsob, jak přidat vlastní akce a upravit příkazy tak, aby pracovaly se specifikacemi akcí.

Snadno můžete přidat své vlastní akce: Specifikace akcí, například `"action"@"slide numbers"` jednoduše vloží prostředí s názvem `"actionenv"` okolo příkazu `\item` nebo parametru `\action` se specifikací překrytí `"slide numbers"`. Definicí nového prostředí s názvem `"my action name"env`, které pracuje se specifikací překrytí, můžete přidat vlastní akci:

```
\newenvironment{checkenv}{\only{\setbeamertemplate{itemize
item}{X}}}{}
```

Následně můžete zapsat

```
\item<beamer:check@2> Text.
```

Tím změníte výčtový symbol před položkou `Text.` na `X` na snímku 2 v režimu `beamer`. Definice příkazu `checkenv` využila faktu, že `\only` přijímá i specifikaci překrytí zadávanou za svým argumentem. Celý mechanismus akcí je založen na níže uvedeném prostředí:

```
\begin{actionenv}<"action specification">
  "environment contents"
\end{actionenv}
```

Toto prostředí extrahuje všechny akce ze specifikace akcí `"action specification"` pro aktuální režim. Pro každou akci ve formě `"action"@"slide numbers"` vloží následující text: `\begin{"action"env}<"slide number">` na začátku prostředí a text `\end{"action"env}` na jeho konci. Existuje-li několik specifikací akcí, otevře se několik prostředí (které se opět zavřou v příslušném pořadí). Specifikace překrytí `"overlay specification"` bez akce se prosadí do `uncover@"overlay specification"`.

Není-li tzv. výchozí specifikace překrytí prázdná, použije se za předpokladu,

že není zadána žádná specifikace akcí "*action specification*". Výchozí specifikace překrytí je obvykle jednoduše prázdná, avšak je možné ji nastavit buď zadáním dalšího volitelného argumentu do příkazu `\frame` nebo do prostředí `itemize`, `enumerate` nebo `description` (další informace najdete u popisu těchto prostředí). Výchozí specifikaci akcí je možné nastavit také použitím příkazu `\beamerdefaultoverlayspecification`, viz níže.

Příklad:

```
\begin{frame}
  \begin{actionenv}<2-| alert@3-4,6>
    Tento text se zobrazí stejným způsobem jako text níže.
  \end{actionenv}

  \begin{uncoverenv}<2->
    \begin{alertenv}<3-4,6>
      Tento text se zobrazí stejným způsobem jako text
výše.
    \end{alertenv}
  \end{uncoverenv}
\end{frame}
```

```
\action<"action specification">{"text"}
```

Příkaz má stejný efekt jako vložení textu "*text*" do příkazu `actionenv`.

Příklad:

```
\action<alert@2>{Také lze použít \alert<2>{}}.}

\beamerdefaultoverlayspecification{"default overlay specification"}
```

Lokálně nastaví výchozí specifikaci překrytí na zadanou hodnotu. Tato specifikace překrytí se použije v každém prostředí `actionenv` a v každé položce `\item`, která nemá vlastní specifikaci překrytí. Hlavním použitím tohoto příkazu je instalace přírůstkové specifikace překrytí, například `<+>` nebo `<+>| alert@+>`.

Obvykle se výchozí specifikace překrytí instaluje automaticky pomocí volitelných argumentů do `\frame`, `frame`, `itemize`, `enumerate`, a `description`. Tento příkaz je nutné použít v případě, že chcete provádět nějaké zábavné akce.

Zadáte-li jej mimo rámeček, nastaví tento příkaz výchozí specifikaci překrytí u všech následujících rámečků, u nichž si nepřejete potlačit výchozí specifikaci překrytí.

Příklad:

```
\beamerdefaultoverlayspecification{<+>}
```

Příklad:

```
\beamerdefaultoverlayspecification{}
```

Vymaže výchozí specifikaci překrytí. (V podstatě nainstaluje výchozí specifikaci překrytí <*>, což jednoduše znamená "vždy", avšak přenositelným způsobem vymazání výchozí specifikace překrytí je toto volání.)

5.3.2.6.4 Přírůstkové specifikace

Tato část má význam spíše pro uživatele, kteří již ve velké míře specifikace překrytí používali a nadále je nebudou psát kód, jako je např. <1->, <2->, <3-> apod. stále dokola.

Často budete vytvářet obdobné specifikace překrytí s níže uvedenou strukturou:

```
\begin{itemize}
\item<1-> Jablko
\item<2-> Broskev
\item<3-> Švestka
\item<4-> Pomeranč
\end{itemize}
```

Problémy nastanou v okamžiku, kdy se rozhodnete vložit nové ovoce, například na začátek. V takovém případě byste museli upravit všechny specifikace překrytí. Pokud navíc přidáte příkaz `\pause` před `itemize`, museli byste aktualizovat i specifikace překrytí.

Třída BEAMER poskytuje speciální syntaxi, díky níž vytvoříte seznamy podobné tomu výše uvedenému poněkud "robustněji". Můžete jej nahradit níže uvedeným seznamem přírůstkových specifikací překrytí:

```
\begin{itemize}
\item<+> Jablko
\item<+> Broskev
\item<+> Švestka
\item<+> Pomeranč
\end{itemize}
```

Příkaz `--sign` má následující efekt: Můžete jej použít v kterékoliv specifikaci překrytí v místech, kde byste obvykle použili číslo. Načte-li se příkaz `--sign`, nahradí se aktuální hodnotou počítadla `LATEXbeamerpauses`, které má na začátku rámce hodnotu 1. Poté se počítadlo zvýší o 1, ačkoliv se zvyšuje pouze jednou za každou specifikaci překrytí, a to i v případě, že specifikace obsahuje několik příkazů `--sign` (nahradí se stejným číslem).

Ve výše uvedeném příkladu je první specifikace nahrazena hodnotou `<1->`. Druhá je nahrazena hodnotou `<2->` atd. Nyní můžeme snadno vložit nové položky, aniž bychom museli cokoli změnit. Můžeme zapsat následující kód:

```
\begin{itemize}
\item<+ -| alert@+> Jablko
\item<+ -| alert@+> Broskev
\item<+ -| alert@+> Švestka
\item<+ -| alert@+> Pomeranč
\end{itemize}
```

Tím při odkrytí aktivujeme aktuální položku. První specifik. `<+ -| alert@+>` se například nahradí výrazem `<1 -| alert@1>`, druhá se nahradí výrazem `<2 -| alert@2>` atd. Jelikož vám prostředí `itemize` umožňuje specifikovat výchozí specifikaci překrytí (viz dokumentace daného prostředí), výše uvedený příklad je možné zapsat dokonce ještě úsporněji takto:

```
\begin{itemize}[<+ -| alert@+>]
\item Jablko
\item Broskev
\item Švestka
\item Pomeranč
\end{itemize}
```

Příkaz `\pause` také aktualizuje počítadlo `beamerpauses`. Toto počítadlo můžete změnit i sami pomocí normálního příkazu `LATEX\setcounter` či `\addtocounter`.

Po jakémkoliv výskytu příkazu `--sign` může následovat posunutí v kulatých závorkách. Toto posunutí se přidá k hodnotě `beamerpauses`. Má-li tedy `beamerpauses` hodnotu 2, pak se `<+(1)->` zvětší na `<3->` a `<+(-1)->` se zvětší na `<1-2>`.

Existuje ještě jedno znaménko, které můžete použít ve specifikaci překrytí a chová se podobně jako `--sign`: tečka. Zapišete-li `<.->`, dojde k podobné akci jako v případě `<+ ->` s výjimkou toho, že počítadlo `beamerpauses` se nezvýší a obdržíte hodnotu počítadla `beamerpauses` sniženou o jedna. Z toho vyplývá, že tečka následovaná posunutím jednoduše zvýší aktuální hodnotu počítadla

`beamerpauses` minus jedna a možné posunutí. Tato notace s tečkou může být užitečná například v níže uvedeném případě:

```
\begin{itemize}[<+>]
\item Jablko
\item<.-> Broskev
\item Švestka
\item Pomeranč
\end{itemize}
```

V tomto příkladu se ukáže druhá položka ve stejný okamžik jako první položka, jelikož neaktualizuje počítadlo. V níže uvedeném příkladu se aktivuje zadaný text pokaždé, když je odkryta určitá položka. Při odkrytí další položky dojde k ukončení tohoto pozměňování.

```
\begin{itemize}[<+>]
\item Toto je \alert<.>{důležité}.
\item Chceme \alert<.>{zvýraznit} toto a~\alert<.>{toto}.
\item Co je \alert<.>{matice}?
\end{itemize}
```

5.3.3 Struktura prezentace: statická globální struktura

Tato část uvádí seznam příkazů, které se používají pro "globální" strukturování prezentace pomocí příkazů, jako jsou `\section` nebo `\part`. Tyto příkazy se používají k vytvoření statické struktury, tj. výsledná prezentace je normálně zobrazována snímek za snímkem v pořadí jednotlivých snímků. V případě interaktivní struktury je nutné interagovat s prezentačním programem. Typicky předvádíte prezentaci klikáním na hyperlinkové odkazy.

5.3.3.1 Přidání titulu stránky

Příkaz `\titlepage` můžete použít k vložení titulu stránky do rámce. Standardně tento příkaz rozvrhne na titulní stránce níže uvedené prvky: titul dokumentu, jméno autora (autorů), jejich příslušnost, titulní grafiku a datum.

```
\titlepage
```

Vloží text titulní stránky do aktuálního rámce.

Příklad:

```
\frame{\titlepage}
```

Příklad:

`\frame[plain]{\titlepage}` v případě titulní stránky, která vyplní celý rámeček.

Beamer-Template/-Color/-Font title page

Tato šablona se zavolá při použití příkazu `\titlepage`.

Jsou předdefinovány níže uvedené volby šablony:

- `[default]["alignment"]` Titulní stránka se vysází se zobrazením titulu, autora, jeho či její příslušnosti, data a titulní grafiky. Chybí-li některý z těchto prvků, nezobrazí se. Jsou-li definovány `BEAMER-color title`, `author`, `institute` nebo `date`, použijí se na tyto položky s výjimkou titulní grafiky jako barva textu `textcolor`. Mají-li definovanou barvu pozadí, vykreslí se za nimi panel odpovídající barvy, který se roztáhne přes celou šířku textu. Pro tyto položky se použijí odpovídající fonty `BEAMER-fonts`.

Volba zarovnání *"alignment option"* se předá do `beamercolorbox` a je možné ji použít zadáním volby `left` například k zarovnání titulu stránky vlevo. Pro tuto šablonu jsou užitečné níže uvedené příkazy:

- `\insertauthor` vloží verzi jména autora, která se hodí na titulní stránku.
- `\insertdate` vloží datum.
- `\insertinstitute` vloží institut.
- `\inserttitle` vloží verzi titulu dokumentu, která se hodí pro titulní stránku.
- `\insertsubtitle` vloží verzi podtitulu dokumentu, která se hodí pro titulní stránku.
- `\inserttitlegraphic` vloží do šablony titulní grafiku.

Z důvodu kompatibility s ostatními třídami se v režimu `article` poskytuje i níže uvedený příkaz:

```
\maketitle
```

```
\title["short title"]{"title"}
```

Volba krátkého titulu *"short title"* se používá v záhlaví a zápatí. Uvnitř *"title"* je možné vložit zalomení řádků pomocí příkazu se dvěma zpětnými lomítky.

Příklad:

```
\title{Třída Beamer}
\title[Short Version]{Velmi dlouhý titul\\na několika řá-
cích}

\subtitle["short subtitle"] {"subtitle"}
```

Volba *"subshort title"* se standardně nepoužívá, avšak je k dispozici pomocí vsuvky `\insertshortsubtitle`. Podtitul se zobrazí pod titulem a je napsán menším fontem.

Příklad:

```
\title{Třída Beamer}
\subtitle{Úvod napsaný volným tempem s mnoha příklady.}

\author["short author name"] {"author name"}
```

Jména by měla být oddělena příkazem `\and`. Mají-li autoři odlišnou příslušnost, měli by mít koncovku přiřazenou příkazem `\inst` s různými parametry.

Příklad:

```
\author[Hemaspaandra et al.]{L. Hemaspaandra\inst{1} \and T.
Tantau\inst{2}}

\institute["short institute"] {"institute"}
```

Je-li zadán více než jeden institut, měly by se jednotlivé instituty oddělit příkazem `\and` a měli byste jim přidat předponu příkazem `\inst` s různými parametry.

Příklad:

```
\institute[Univerzity Rochester a Berlín]{
\inst{1}Oddělení počítačových věd\\
Univerzita Rorchester
\and
\inst{2}Fakulta elektrotechniky a informatiky\\
Technická univerzita Belínn}
```

```
\date["short date"]{"date"}
```

Příklad:

```
\date{\today} nebo \date[STACS 2003]{STACS Conference, 2003}.
```

```
\titlegraphic{"text"}
```

Text *"text"* se zobrazuje jako grafika titulu. Obvykle se jako *"text"* použije prostředí s obrázky.

Příklad:

```
\titlegraphic{\pgfuseimage{titlegraphic}}
```

```
\subject{"text"}
```

Zadá *"text"* jako text předmětu v údajích o dokumentu PDF. Aktuálně nemá žádný efekt.

```
\keywords{"text"}
```

Zadá *"text"* jako klíčová slova do údajů o dokumentu PDF. Aktuálně nemá žádný efekt. Standardně vloží příkazy `\title` a `\author` do výsledného souboru PDF a souvisejících polí s informacemi o dokumentu také své argumenty. To může způsobit problémy, používáte-li jako argumenty těchto příkazů komplikované struktury, jako jsou například rámečky. V tomto případě můžete vypnout automatické generování těchto položek použitím níže uvedené volby třídy:

```
\documentclass[usepdftitle=false]{beamer}
```

Potlačí automatické generování položek titulu a autora v údajích o dokumentu PDF.

5.3.3.2 Přidání oddílů a pododdílů

Text můžete strukturovat pomocí příkazů `\section` a `\subsection`. Na rozdíl od standardního systému \LaTeX nevytvoří tyto příkazy titul na pozici, kde je použijete. Místo toho přidají položku do přehledu kapitol a také do navigačních lišt. Chcete-li vytvořit zalomení řádků v přehledu kapitol (to většinou není dobrý nápad), můžete použít příkaz `\breakhere`. Standardní příkaz `\` nefunguje.

```
\section<"mode specification">["short section name"]{"section name"}
```

Zahájí oddíl. Nevytvoří se žádný titul. Název oddílu *"section name"* se zobrazí v přehledu kapitol a v navigačních lištách s výjimkou případu, kdy je zadán zkrácený název oddílu *"short section name"*. V tomto případě se namísto toho použije v navigačních lištách krátký název oddílu *"short section name"*. Je-li zadána specifikace režimů *"mode specification"*, bude mít příkaz efekt pouze v zadaných režimech.

Příklad:

```
\section[Summary]{Přehled hlavních výsledků}
```

Příklad:

```
\section<presentation>[Results]{Výsledky hlavního problému}
\section<article>{Výsledky hlavního problému}
```

Beamer-Template/-Color/-Font section in toc (oddíl v toc)

Tato šablona se používá k vysázení položky oddílu. Povolené volby *"options"* najdete u nadřazené šablony *table of contents*.

Pro tuto šablonu jsou užitečné níže uvedené příkazy:

- `\inserttocsection` vloží verzi názvu aktuálního oddílu s přehledem kapitol.
- `\inserttocsectionnumber` vloží číslo aktuálního oddílu (v přehledu kapitol).

Beamer-Template/-Color/-Font section in toc shaded (stínovaný oddíl v toc)

Tato šablona se používá místo té předchozí v případě, že se daný oddíl má zobrazit stínovaně, jelikož není aktuálním oddílem. Povolené volby *"options"* najdete u nadřazené šablony *table of contents*.

```
\section<"mode specification">*"section name"}
```

Zahájí sekci bez položky v přehledu kapitol. Nevytvoří se žádný titul, avšak v navigační liště se zobrazí název oddílu *"section name"*.

Příklad:

```
\section*{Přehled}
```

Příklad:

```
\section<beamer>*{Přehled}
```

```
\subsection<"mode specification">["short subsection name"]
{"subsection name"}
```

Tento příkaz pracuje stejně jako příkaz `\section`

Příklad:

```
\subsection[Applications]{Prostředky pro redukci znečištění}
```

Beamer-Template/-Color/-Font subsection in toc (pododdíl v toc)

Stejně jako oddíl v toc, avšak platí pro pododdíl.

Kromě vsuvek pro šablonu `section in toc` jsou pro tuto šablonu k dispozici níže uvedené příkazy:

- `\inserttocsubsection` vloží verzi aktuálního názvu pododdílu s přehledem kapitol.
- `\inserttocsubsectionnumber` vloží číslo aktuálního pododdílu (v přehledu kapitol).

Beamer-Template/-Color/-Font subsection in toc shaded (stínovaný pododdíl v toc)

Stejně jako stínovaný oddíl v toc, avšak platí pro pododdíly.

```
\subsection<"mode specification">*"subsection name"
```

Zahájí pododdíl bez položky v přehledu kapitol. Nevytvoří se žádný titul, avšak v navigační liště se zobrazí název pododdílu *"subsection name"* s výjimkou případu, kdy je *"subsection name"* prázdné. V tomto případě se nevytvoří ani přehled kapitol ani položka navigační lišty. Jakékoliv rámce v tomto *"prázdném"* pododdílu se zobrazí v navigační liště.

Příklad:


```

\section{Shrnutí}
  \frame{Tento rámeček se nezobrazí v navigační liště}
  \subsection*{}
  \frame{Tento rámeček se zobrazí v navigační liště, avšak
nezobrazí se žádná
      položka pododdílu.}
  \subsection*{Pododdíl}
  \frame{Normální rámeček, zobrazí se v navigační liště. Ná-
zev pododdílu se
      také zobrazí v navigační liště, avšak nikoliv v přehle-
du kapitol.}

```

```

\subsubsection<"mode specification">["short subsubsection na-
me"]{"subsubsection name"}

```

Tento příkaz pracuje jako `\subsection`. Podpododdíly jsou však podpo-
rovány méně než pododdíly. Podpododdíly se například v přehledu kapitol
zobrazují vždy se stejnými parametry stínování/skrytí jako pododdíly. Silně
nedoporučuji používat podpododdíly v prezentacích. Pokud je nepoužijete,
prezentace bude lepší.

Příklad:

```

\subsubsection[Applications]{Prostředky pro redukci znečiš-
tění}

```

Beamer-Template/-Color/-Font subsubsection in toc (podpodod-
díly v toc)

Stejně jako pododdíl v toc, avšak platí pouze pro pododdíl.

Kromě vsuvek pro šablonu `subsection in toc` jsou pro tuto šablonu k dis-
pozici níže uvedené příkazy:

- `\inserttocsubsubsection` vloží verzi aktuálního názvu podpododdílu
s přehledem kapitol.
- `\inserttocsubsubsectionnumber` vloží číslo aktuálního podpododdílu
(v přehledu kapitol).

Beamer-Template/-Color/-Font subsubsection in toc shaded (stí-
nované podpododdíly v toc)

Stejně jako stínovaný pododdíl v toc, avšak platí pouze pro podpododdíly.

```
\subsection<"mode specification">*"subsubsection name"}
```

Zahájí pododdíl bez položky v přehledu kapitol. Nevytvoří se žádný titul, avšak v případném postranním panelu se zobrazí název podpododdílu *"subsubsection name"*. Často budete chtít zobrazit určitý typ rámce hned po zahájení oddílu či pododdílu. Například si můžete přát, aby každý pododdíl začínal rámcem, který zobrazí přehled kapitol, v němž bude aktuální pododdíl zvýrazněn. To můžete provést níže uvedenými příkazy.

```
\AtBeginSection["special star text"]{"text"}
```

Text se vloží na začátek každého oddílu. Zadáte-li parametr *"special star text"*, tento text se namísto toho použije pro hvězdičkové oddíly. Různá volání tohoto příkazu *"nepřidají"* zadané texty (jako to dělá příkaz `\AtBeginDocument`), ale přepíšu jakýkoliv předchozí text.

Příklad:

```
\AtBeginSection[] % Nedělat nic v oddílu \section*
{
  \begin{frame}<beamer>
    \frametitle{Přehled}
    \tableofcontents[currentsection]
  \end{frame}
}
```

```
\AtBeginSubsection["special star text"]{"text"}
```

Zadaný text se vloží na začátek každého pododdílu. Zadáte-li parametr *"special star text"*, tento text se namísto toho použije pro hvězdičkové pododdíly. Různá volání tohoto příkazu zadané texty *"nepřidávají"*.

Příklad:

```
\AtBeginSubsection[] % Nedělat nic pro pododdíl \subsection*
{
  \begin{frame}<beamer>
    \frametitle{Přehled}
    \tableofcontents[currentsection,currentsubsection]
  \end{frame}
}
```

```
\AtBeginSubsubsection["special star text"]{"text"}
```

Stejně jako `\AtBeginSubsection`, pouze pro podpododdíly.

5.3.3.3 Přidání částí

Přednášíte-li dlouhou prezentaci (například přednášku), můžete svoji prezentaci rozvrhnout do několika částí. Každá část pak funguje jako "prezentace sama o sobě", která má svůj vlastní přehled kapitol, navigační lišty atd. V dané části se pak oddíly a pododdíly ostatních částí vůbec nezobrazují.

Novou část vytvoříte příkazem `\part`. Všechny oddíly a pododdíly následující po tomto příkazu budou pro tuto část "lokální". Stejně jako příkazy `\section` a `\subsection`, ani příkaz `\part` nevytváří žádné rámce ani zvláštní text. Často se však doporučuje na začátku jednotlivých částí použít příkaz `\partpage`, jímž vložíte text do rámce, který "ohlašuje" začátek nové části.

```
\part<"mode specification">["short part name"]{"part name"}
```

Zahájí část. Název "*part name*" se zobrazí při použití příkazu `\partpage`. Název "*shown part name*" se standardně nikde nenobrazuje, avšak získáte k němu přístup příkazem `\insertshortpart`.

Příklad:

```
\begin{document}
  \frame{\titlepage}

  \section*{Přehled}
  \subsection{Část I: Souhrn předchozí přednášky}
  \frame{
    \frametitle{Přehled části I}
    \tableofcontents[part=1]}
  \subsection{Část II: Dnešní přednáška}
  \frame{
    \frametitle{Přehled části II}
    \tableofcontents[part=2]}

  \part{Přehled předchozí přednášky}
  \frame{\partpage}
  \section[Previous Lecture]{Souhrn předchozí přednášky}
  \subsection{Témata}
  \frame{...}
  \subsection{Cíle přednášky}

  \frame{...}
```

```

\part{Dnešní přednáška}
\frame{\partpage}
\section{Téma A}
\frame{\tableofcontents[currentsection]}
\subsection{Fuj}
\frame{...}
\section{Téma B}
\frame{\tableofcontents[currentsection]}
\subsection{bar}
\frame{...}
\end{document}

```

`\partpage`

Funguje stejně jako `\titlepage`, avšak "ohlašuje se" aktuální část, nikoliv aktuální prezentace.

Příklad:

```
\frame{\partpage}
```

Beamer-Template/-Color/-Font part page

Tato šablona se zavolá při použití příkazu `\partpage`.

Jsou předdefinovány níže uvedené volby šablony:

- `[default] ["alignment"]` Stránka z dané části se vysází s vyobrazením aktuálního čísla dané části. Níže se uvede titul aktuální části. Použijí se nastavení barvy a fontu `BEAMER-color {font part page}` včetně barvy pozadí na stránce v příslušné části. Co se týče šablony titulní stránky, volba zarovnání "*alignment option*" se předá do `beamercolor-box`.

Pro tuto šablonu jsou užitečné níže uvedené příkazy:

- `\insertpart` vloží název aktuální části.
- `\insertpartnumber` vloží do šablony číslo aktuální části v podobě arabské číslice.
- `\insertpartromannumber` vloží do šablony číslo aktuální části v podobě římské číslice.

```
\AtBeginPart{"text"}
```

Zadaný text se vloží na začátek každé části.

Příklad:

```
\AtBeginPart{\frame{\partpage}}
```

5.3.3.4 Rozdělení kurzu do přednášek

Používáte-li třídu BEAMER v režimu article, budete si možná chtít uschovat poznámky z přednášek celého kurzu do jednoho souboru. V tomto případě je součástí dané přednášky vlastně jen několik rámců.

Příkazem `\lecture` snadno vyberete pouze určitou část rámců z prezentovaného souboru. Tento příkaz přebírá (kromě jiného) název popisky. Zapišete-li příkaz `\includeonlylecture` pomocí tohoto názvu popisky, zobrazí se pouze rámce, v nichž následuje odpovídající příkaz `\lecture`. Rámce, v nichž následuje jiný příkaz než `\lecture`, se potlačí.

Standardně nemá příkaz `\lecture` jiný efekt. Nevytváří žádné rámce ani nevkládá položky do přehledu kapitol. Chcete-li však, aby třída BEAMER vložila např. titulní stránku na začátek (každé) přednášky, můžete použít příkaz `\AtBeginLecture`.

```
\lecture["short lecture name"]{"lecture name"}{"lecture label"}
```

Zahájí přednášku. Název *"lecture name"* bude k dispozici pomocí příkazu `\insertlecture`. Název *"short lecture name"* bude k dispozici pomocí příkazu `\insertshortlecture`.

Příklad:

```
\begin{document}
\lecture{Vektorové prostory}{týden 1}

\section{Úvod}...

\section{Souhrn}

\lecture{Skalární součiny}{týden 2}

\section{Úvod}
...
\section{Souhrn}

\end{document}
```

```
\includeonlylecture "lecture label"
```

Potlačí všechny příkazy `\frame`, `frame`, `\section`, `\subsection` a `\part`, které následují pro příkazu `\lecture` s výjimkou případu, kdy se popisek přednášky shoduje s *"lecture label"*. Rámce před veškerými příkazy `\lecture` se zahrnou vždy. Tento příkaz byste měli zadat v úvodu.

Příklad:

```
\includeonlylecture{týden 1}

\AtBeginLecture{"text"}
```

Zadaný text se vloží na začátek každé přednášky.

Příklad:

```
\AtBeginLecture{\frame{\Dnešní dlouhá přednáška: \insertlec-
ture}}
```

5.3.3.5 Přidání přehledu kapitol

Přehled kapitol můžete vytvořit použitím příkazu `\tableofcontents`. Na rozdíl od normálního příkazu `LATEX` pro vytvoření přehledu kapitol přebírá tento příkaz volitelný parametr v hranatých závorkách, jímž můžete vytvořit určité speciální efekty.

```
\tableofcontents["comma-separated option list"]
```

Vloží do aktuálního rámce přehled kapitol.

Příklad:

```
\section*{Přehled}
\frame{\tableofcontents}

\section{Úvod}
\frame{\tableofcontents[currentsection]}
\subsection{Proč?}
\frame{...}
\frame{...}
\subsection{Kde?}
\frame{...}
```

```

\section{Výsledky}
\frame{\tableofcontents[currentsection]}
\subsection{Protože}
\frame{...}
\subsection{Zde}
\frame{...}

```

Je možné zadat následující volby:

- `currentsection` s výjimkou aktuálního oddílu zobrazí všechny oddíly poloprůhledně. S výjimkou pododdílů v aktuálním oddílu se všechny pododdíly zobrazí poloprůhledně. Tento příkaz je zkráceným zadáním následujících voleb `sectionstyle=show/shaded`, `subsectionstyle=show/show/shaded`
- `currentsubsection` s výjimkou aktuálního pododdílu zobrazí všechny pododdíly poloprůhledně. Tento příkaz je zkráceným zadáním volby `subsectionstyle=show/shaded`.
- `firstsection="section number"` specifikuje oddíl, který se má očíslovat jako oddíl "1". To je užitečné v případě, že máte první oddíl (třeba oddíl s přehledem), který by neměl mít žádné číslo. Čísla oddílů se standardně nezobrazují. Chcete-li je zobrazit, je nutné nejdříve instalovat jiné šablony s přehledem kapitol.
- `hideallsubsections` skryje všechny pododdíly. Tento příkaz je zkráceným zadáním volby `subsectionstyle=hide`.
- `hideothersubsections` s výjimkou aktuálního pododdílu skryje pododdíly oddílů. Tento příkaz je zkráceným zadáním volby `subsectionstyle=show/show/hide`.
- `part="part number"` zobrazí přehled kapitol části "*part number*" namísto zobrazení přehledu kapitol aktuální části (což je výchozí nastavení). Tuto volbu je možné kombinovat s dalšími volbami, ačkoliv kombinace s volbou `current` očividně nemá smysl.
- `pausesections` zadá příkaz `\pause` před každý oddíl. To je užitečné, chcete-li zobrazit přehled kapitol přírůstkově.
- `pausesubsections` zadá příkaz `\pause` před každým pododdílem.

- `sections={"overlay specification"}` zobrazí se pouze oddíly označené ve specifikaci překrytí "*overlay specification*". Například `sections={<2-4| handout:0>}` zobrazí pouze druhý, třetí a čtvrtý oddíl v normální verzi. Ve verzi prospektu se nezobrazí nic a ve zbývajících verzích se zobrazí vše. Vynecháte-li špičaté závorky, specifikace se použije na všechny verze. Příkaz `sections={2-4}` tedy zobrazí oddíly dva, tři a čtyři ve všech verzích.
- `sectionstyle="style for current section" / "style for other sections"` (styl aktuálního oddílu / styl dalších oddílů) specifikuje způsob zobrazení oddílů. Povolené styly "*styles*" jsou `show`, `shaded` a `hide`. První zobrazí titul oddílu normálně, druhý jej zobrazí poloprůhledně a třetí jej úplně potlačí. Druhý styl můžete také vynechat. V tom případě se první styl použije na všechny oddíly (to není moc užitečné).
- `subsectionstyle="style for current subsection" / "style for other subsections in current section" / "style for subsections in other sections"` (styl aktuálního pododdílu / styl dalších pododdílů v aktuálním oddílu / styl pododdílů v dalších oddílech) udává způsob zobrazení pododdílů. Je možné zadat stejné volby jako pro volbu `sectionstyle`. Poslední styl můžete vynechat. V tom případě se na druhý styl aplikuje i na ten poslední. Také můžete vynechat poslední dva styly, a první styl se tak použije na všechny ostatní.

Příklad: `subsectionstyle=shaded` všechny pododdíly se zobrazí stínovaně.

Příklad: `subsectionstyle=hide` skryje všechny pododdíly.

Příklad: `subsectionstyle=show/shaded` s výjimkou aktuálního pododdílu v aktuálním oddílu zobrazí všechny pododdíly poloprůhledně.

Příklad: `subsectionstyle=show/show/hide` s výjimkou aktuálního oddílu potlačí všechny pododdíly.

Příklad: `subsectionstyle=show/shaded/hide` s výjimkou aktuálního oddílu potlačí všechny pododdíly a zvýrazní pouze aktuální pododdílu v aktuálním oddílu.

Poslední příkazy jsou užitečné v případě, že během představení přehledu své prezentace nechcete zobrazit příliš mnoho podrobností.

Parent Beamer-Template sections/subsections in toc (oddíly /pododdíly v toc)

Toto je nadřazená šablona, jejíž podřízenými prvky jsou `section in toc` a `subsection in toc`. To znamená, že pokud v této šabloně použijete příkaz

`\setbeamertemplate`, příkaz se místo toho zavolá na oba podřízené prvky (se stejnými argumenty).

Jsou předdefinovány níže uvedené volby šablony:

- `[default]` Standardně se oddíly a pododdíly vysázejí pomocí fontů a barev šablon `section in toc` a `subsection in toc`. Barvy pozadí se však ignorují. Pododdíly jsou odsazeny.
- `[sections numbered]` Podobné jako výchozí nastavení. Avšak zobrazují se i čísla oddílů. Pododdíly nejsou číslovány.
- `[subsections numbered]` Tentokrát se pododdíly číslovají, avšak oddíly nikoliv. Jelikož jsou však pododdíly "plně očíslovány", např. "1.2" nebo "3.2", tak v případě, že každý oddíl má alespoň jeden pododdíl, očíslovaný oddíl nebude ve skutečnosti chybět.
- `[circle]` Před oddíly vykreslí malé kroužky. Uvnitř kroužků se zobrazí číslo oddílu. Pro vysázení kroužků se použije font a barva `BEAMER section number projected`, tj. kroužek bude mít barvu pozadí a text uvnitř rámečku bude mít barvu popředí.
- `[square]` Podobné jako volba `circle` s tou výjimkou, že místo kroužků se použijí čtverečky. Před pododdíly se zobrazí malé neočíslované čtverečky.
- `[ball]` Stejně jako volba `square`. Jediným rozdílem je to, že se použijí vyplněné kroužky namísto čtverečků.
- `[ball unnumbered]` Podobné jako volba `ball`. Nepoužije se však žádné číslování. Touto volbou se přehled kapitol zobrazí podobně jako při `itemize`.

Jestliže vám žádná z výše uvedených voleb nevyhovuje, je nutné změnit šablony `section in toc` a `subsection in toc` přímo.

Parent Beamer-Template sections/subsections in toc shaded
(stínované oddíly/pododdíly v toc)

Nadřazená šablona s podřízenými šablonami `section in toc shaded` a `subsection in toc shaded`. Používají se k vykreslení položek oddílů a pododdílů, které jsou aktuálně zastíněné stejně jako všechny neaktuální pododdíly v `\tableofcontents[current`

Jsou předdefinovány níže uvedené volby šablony:

- [default] ["opaqueness"] Standardně zobrazují šablony `section in toc shaded` a `subsection in toc shaded` svůj obsah zastíněně, avšak pouze s "`opaqueness`"% krytím. Výchozí hodnotou je 20%.

Příklad:

```
\setbeamertemplate{table of contents shaded}[default][50]
```

zobrazí zašedlé položky s 50% průhledností.

5.3.3.6 Přidání bibliografie

V prezentaci BEAMER můžete použít prostředí bibliography a příkazy \LaTeX `\cite`. Obvykle je nutné vysadit bibliografické položky částečně ručně. Můžete však použít bibtex a vytvořit "první odhad" bibliografie. Do prezentace zkopírujte obsah souboru `main.bbl`. Je to mocný nástroj pro vysoce kvalitní tvorbu citací.

Pomocí nástroje bibtex nebo editoru umístěte bibliografické reference do prostředí `thebibliography`. Toto prostředí (standardní \LaTeX) přebírá jeden parametr, který by měl být nejdelsí popiskou `\bibitem` v níže uvedeném seznamu bibliografických položek.

```
\begin{thebibliography}{"longest label text"}
  "environment contents"
\end{thebibliography}
```

Do aktuálního rámce vloží bibliografii. Text nejdelsí popisky "*longest label text*" se použije k nastavení odsazení seznamu. Některé předdefinované volby pro vysazení bibliografie však tento parametr ignorují, jelikož reference nahrazují symbolem.

V daném prostředí použijte příkaz `\bibitem` (standardní \LaTeX) pro každou referenční položku. Uvnitř každé položky použijte příkaz `\newblock` (standardní \LaTeX), který oddělí jména autorů, titul, referenci na knihu/časopis a případné poznámky. Každý z těchto příkazů může zavést novou řádku či barvu nebo jiné formátování, což specifikuje bibliografická šablona.

Prostředí je nutné vložit do rámce. Jestliže se bibliografie nevejde do jednoho rámce, měli byste ji rozdělit (vytvořit nový rámec a druhé prostředí `thebibliography`) nebo použít volbu `allowframebreaks`.

Příklad:

```
\begin{frame}
  \frametitle{Další četba}
```

```

\begin{thebibliography}{Dijkstra, 1982}
\bibitem[Solomaa, 1973]{Solomaa1973}
  A.~Salomaa.
  \newblock {\em Formal Languages}.
  \newblock Academic Press, 1973.

\bibitem[Dijkstra, 1982]{Dijkstra1982}
  E.~Dijkstra.
  \newblock Smoothsort, an alternative for sorting
in situ.
  \newblock {\em Science of Computer Programming},
1(3):223--233, 1982.
\end{thebibliography}
\end{frame}

```

Vzhled textu s popisem autora, titulu, časopisu a poznámek popisují čtyři šablony. Tyto autorské šablony se vloží před první blok položky (první blok je veškerý text před prvním výskytem příkazu `\newblock`). Šablona titulu se vloží před druhý blok (text mezi prvním a druhým výskytem příkazu `\newblock`). To samé platí pro šablony časopisu a poznámek. Tyto šablony se vloží před bloky a k těmto samotným blokům nemáte přístup přes příkazy vložení. Před bloky se také instalují odpovídající barvy a fonty BEAMER.

Beamer-Template/-Color/-Font bibliography entry author
(autor bibliografické položky)

Tato šablona se vloží před autora bibliografické položky. Instaluje se také barva a font. Efekt této šablony přetrvává až do konce bibliografické položky nebo dokud jedna z níže uvedených šablon efekt nezruší. Ve výchozím nastavení tato šablona nedělá nic. Výchozí barvou je barva struktury.

Beamer-Template/-Color/-Font bibliography entry title (titul bibliografické položky)

Tato šablona se vloží před titul bibliografické položky (přesněji řečeno se vloží po prvním výskytu příkazu `\newblock`). Standardně tato šablona zahájí nový odstavec, a způsobí tak zalomení řádku. Výchozí barvou je normální barva textu.

Beamer-Template/-Color/-Font bibliography entry journal (časopis bibliografické položky)

Tato šablona se vloží před časopis bibliografické položky (druhý příkaz `\newblock`). Ve výchozím nastavení tato šablona zahájí nový odstavec. Výchozí barvou je poněkud průhlednější verze barvy struktury.

Beamer-Template/-Color/-Font bibliography entry note (poznámka bibliografické položky)

Tato šablona se vloží před text poznámky na konec bibliografické položky (vloží se před třetí příkaz `\newblock`). Ve výchozím nastavení tato šablona zahájí nový odstavec. Výchozí barva je stejná jako barva časopisu.

```
\bibitem<"overlay specification">["citation text"]{"label
name"}
```

Text citace "citation text" se vloží do textu, je-li položka citována pomocí příkazu `\cite{label name}` v hlavním textu prezentace. V prezentaci BEAMER by měl obvykle být co nejdelší. Příkazy `\newblock` oddělíte jména autorů, titul, reference na knihy/časopisy a poznámky. Je-li zadána specifikace překrytí "overlay specification", zobrazí se tato položka pouze na zadaných snímcích.

Příklad:

```
\bibitem[Dijkstra, 1982]{Dijkstra1982}
  E.~Dijkstra.
  \newblock Smoothsort, an alternative for sorting in situ.
  \newblock {\em Science of Computer Programming},
  1(3):223--233, 1982.
```

Beamer-Template/-Color/-Font bibliography item (bibliografická položka)

Nadřazený prvek pro volbu `color/font`: `item`.

Tato šablona se používá k vykreslení bibliografické položky. Na rozdíl od normálního systému L^AT_EX neopakuje výchozí šablona bibliografie před každou položkou bibliografie text citace (např. "[Dijkstra, 1982]"). Namísto toho se vykreslí roztomilý, malý symbol článku. Důvodem je to, že publikum si nezapamatuje žádné zkrácené texty citací až do konce prezentace.

Jsou předdefinovány níže uvedené volby šablony:

- `[default]` Vykreslí roztomilou, malou referenční ikonu článku. Použijte pro články z časopisů, knih (například zápis z jednání konference) nebo technických zpráv.

- `[article]` Alias výchozího nastavení.
- `[book]` Vykreslí roztomilou, malou referenční ikonu knihy. Používejte pro knihy.
- `[triangle]` Vykreslí jako referenci trojúhelník. Slouží pro odkazování na standardní výčtové položky.
- `[text]` Používá referenční text (např. "[Dijkstra, 1982]") jako text reference. Chcete-li použít tuto volbu, ujistěte se, že víte, co děláte.

Pro tuto šablonu je užitečná níže uvedená vsuvka:

- `\insertbiblabel` vloží aktuální popisku citace.

5.3.3.7 Přidání dodatku

Ke své prezentaci můžete přidat dodatek příkazem `\appendix`. Rámce a možná i celé pododdíly byste měli vložit do dodatku, který neplánujete uvést během prezentace, avšak může být užitečný pro zodpovězení dotazů. Příkaz `\appendix` v podstatě spustí novou část s názvem `\appendixname`. Nastaví však i určité hyperlinkové odkazy. Stejně jako ostatní části, i dodatek se uchovává odděleně od vlastní prezentace.

```
\appendix<"mode specification">
```

Zahájí dodatek v zadaných režimech. Žádné příkazy `\subsection` a `\section` použité po tomto příkazu nebudou zobrazeny jako součást normálních navigačních lišt.

Příklad:

```
\begin{document}
\frame{\titlepage}
\section*{Přehled}
\frame{\tableofcontents}
\section{Hlavní text}
\frame{Nějaký text}
\section*{Souhrn}
\frame{Text souhrnu}

\appendix
\section{\appendixname}
```

```

\frame{\tableofcontents}
\subsection{Další materiály}
\frame{Podrobnosti}
\frame{Text, který jste vynechali v hlavní prezentaci}
\subsection{Další dodatečný materiál}
\frame{Další podrobnosti}
\end{document}

```

5.3.4 Strukturujeme prezentaci: Interaktivní globální struktura

5.3.4.1 Přidání odkazů a tlačítek

Chcete-li ve struktuře prezentace vytvořit předpokládané nelineární odsoky, můžete do ní vložit odkazy. Odkazem je text (většinou vykreslený jako tlačítko), po jehož stisknutí odskočíte v prezentaci na nějaký jiný snímek. Takové tlačítko vytvoříte ve třech krocích:

1. Pomocí příkazu `\hypertarget` či (jednoduššího) `\label` zadáte cíl odkazu. V některých případech (viz níže) můžete tento krok vynechat.
2. Tlačítko vykreslíte volbou `\beamerbutton` či jiným podobným příkazem. Tím vykreslíte tlačítko. Pokud na něj však nyní kliknete, nebude to mít ještě žádný efekt.
3. Tlačítko vložte do příkazu `\hyperlink`. Teď na něj můžete kliknout a odskočíte do cíle odkazu.

```
\hypertarget<"overlay specification">{"target name"}{"text"}
```

Použijete-li specifikaci překrytí *"overlay specification"*, bude volba *"text"* cílem skoku na *"target name"* pouze na zadaném snímku. Na všech ostatních snímcích se text zobrazí normálně. Upozorňujeme, že specifikaci překrytí je nutné zadat do příkazu `\hypertarget` vždy, když ji použijete v rámcích, jež mají několik snímků (v opačném případě si systém pdflatex oprávněně stěžuje, že jste definovali stejný cíl na různých snímcích).

Příklad:

```

\begin{frame}
\begin{itemize}
\item<1-> První položka.
\item<2-> Druhá položka.

```

```

\item<3-> Třetí položka.
\end{itemize}

\hyperlink{jumptosecond}{\beamergotobutton{Skok na druhý snímek}}
\hypertarget<2>{jumptosecond}{
\end{frame}

\begin{frame}[label=threeitems]
\begin{itemize}
\item<1-> První položka.
\item<2-> Druhá položka.
\item<3-> Třetí položka.
\end{itemize}

\hyperlink{threeitems<2>}{\beamergotobutton{Skok na druhý snímek}}
\end{frame}

```

Níže uvedené příkazy je možné použít k abstraktnímu zadání akce, pro niž se tlačítko využije.

```
\beamerbutton{"button text"}
```

Vykreslí tlačítko s textem *"button text"*.

Příklad:

```
\hyperlink{somewhere}{\beamerbutton{Přejít na nějaký snímek}}
```

Beamer-Template/-Color/-Font button (tlačítko)

Zavoláte-li příkaz `\beamerbutton`, k vykreslení tlačítka se použije tato šablona. Uvnitř této šablony můžete použít příkaz `\insertbuttontext`, jenž vloží parametr předaný příkazu `\beamerbutton`.

Jsou předdefinovány níže uvedené volby šablony:

- [default] Vysází tlačítko se zakulacenými rohy. Použije se popředí i pozadí tlačítka `BEAMER-color button` a také `BEAMER-font button`. Okraj tlačítka získá popředí volby `BEAMER-color button border`.

U tohoto prvku jsou užitečné níže uvedené vsuvky:

- `\insertbuttontext` vloží text aktuálního tlačítka. Uvnitř tlačítek typu "Přejít na" (viz níže) předchází tomuto textu vsuvka `\insertgotosymbol`. Podobně je tomu v případě tlačítek pro přeskočení a pro návrat.
- `\insertgotosymbol` Tento text se vloží na začátek tlačítek "Přejít na". Opětovnou definicí tohoto příkazu `symbol` změníte.

Příklad:

```
\renewcommand{\insertgotosymbol}{\somearrowcommand}
```

- `\insertskipsymbol` Tento text se vloží na začátek přeskovacích tlačítek.
- `\insertreturnsymbol` Tento text se vloží na začátek tlačítek návratu.

Beamer-Color `button border` (okraj tlačítka) Popředí této barvy se použije k vykreslení okrajů tlačítek.

```
\beamergetobutton{"button text"}
```

Vykreslí tlačítko s textem "*button text*". Před text se vloží malý symbol (většinou šipka ukazující vpravo), jenž naznačuje, že stisknutím tohoto tlačítka přeskočíte do jiné "oblasti" prezentace.

Příklad:

```
\hyperlink{detour}{\beamergetobutton{Přejít k odbočce}}
```

```
\beamerskipbutton{"button text"}
```

Symbolem vykresleným na tomto tlačítku je obvykle dvojitá šipka vpravo. Použijte toto tlačítko, chcete-li přeskočit určitou, přesně definovanou část své prezentace.

Příklad:

```
\frame{
\begin{theorem}
...
\end{theorem}

\begin{overprint}
\onslide<1>
```



```

\hfill\hyperlinkframestartnext{\beamerskipbutton{Přeskočit
důkaz}}
\onslide<2>
\begin{proof}
...
\end{proof}
\end{overprint}
}

```

```
\beamerreturnbutton{"button text"}
```

Symbolem vykresleným na tomto tlačítku je obvykle šipka ukazující vlevo. Použijte toto tlačítko, chcete-li se po jeho stisknutí vrátit z odbočky.

Příklad:

```

\frame<1>[label=mytheorem]
{
\begin{theorem}
...
\end{theorem}

\begin{overprint}
\onslide<1>
\hfill\hyperlink{mytheorem<2>}{\beamergotobutton{Přejít
k podrobnostem důkazu}}
\onslide<2>
\begin{proof}
...
\end{proof}
\hfill\hyperlink{mytheorem<1>}{\beamerreturnbutton{Návrat}}
\end{overprint}
}
\appendix
\againframe<2>{mytheorem}

```

Chcete-li vytvořit "kliknutelné" tlačítko, je nutné jej umístit do příkazu, jako je třeba `\hyperlink`. Příkaz `\hyperlink` je standardním příkazem balíku `hyperref`. Třída BEAMER definuje celou řadu dalších příkazů `hyperlink`, které můžete také používat.

```

\hyperlink<"overlay specification">{"target name"}{"link text"}
<"overlay specification">

```

Je možné zadat pouze jednu specifikaci překrytí *"overlay specification"*. Text odkazu *"link text"* se vysází běžným způsobem. Kliknete-li na libovolné místo tohoto textu, přeskočíte na snímek, v němž jste použili příkaz `\hypertarget` s parametrem *"target name"*. Je-li zadána specifikace překrytí *"overlay specification"*, odkaz (včetně textu *"link text"*) se na snímcích, jež nebyly specifikovány, zcela potlačí.

Níže uvedené příkazy mají předem definovaný cíl, avšak jinak se chovají úplně stejně jako příkaz `\hyperlink`. Obzvláště pak upozorňujeme, že všechny tyto příkazy přijímají specifikaci překrytí. Příkazy přijímají specifikaci na konci, nikoliv na začátku.

```
\hyperlinkslideprev<"overlay specification">{"link text"}
```

Kliknutím na text přeskočíte o jeden snímek zpět.

```
\hyperlinkslidenext<"overlay specification">{"link text"}
```

Kliknutím na text přeskočíte o jeden snímek dopředu.

```
\hyperlinkframestart<"overlay specification">{"link text"}
```

Kliknutím na text přeskočíte na první snímek aktuálního rámce.

```
\hyperlinkframeend<"overlay specification">{"link text"}
```

Kliknutím na text přeskočíte na poslední snímek aktuálního rámce.

```
\hyperlinkframestartnext<"overlay specification">{"link text"}
```

Kliknutím na text přeskočíte na první snímek dalšího rámce.

```
\hyperlinkframeendprev<"overlay specification">{"link text"}
```

Kliknutím na text přeskočíte na poslední snímek předchozího rámce.

Předchozí čtyři příkazy existují také ve verzi, v níž je rámec *"frame"* nahrazen pododdílem *"subsection"* a ve verzi, kde je rámec *"frame"* nahrazen oddílem *"section"*.

```
\hyperlinkpresentationstart<"overlay specification">{"link text"}
```

Kliknutím na text přeskočíte na první snímek prezentace.

```
\hyperlinkpresentationend<"overlay specification">{"link text"}
```

Kliknutím na text přeskočíte na poslední snímek prezentace. To se nevztahuje na dodatek.

```
\hyperlinkappendixstart<"overlay specification">{"link text"}
```

Kliknutím na text přeskočíte na první snímek dodatku. Není-li dodatek k dispozici, přeskočíte na poslední snímek dokumentu.

```
\hyperlinkappendixend<"overlay specification">{"link text"}
```

Kliknutím na text přeskočíte na poslední snímek dodatku.

```
\hyperlinkdocumentstart<"overlay specification">{"link text"}
```

Kliknutím na text přeskočíte na první snímek prezentace.

```
\hyperlinkdocumentend<"overlay specification">{"link text"}
```

Kliknutím na text přeskočíte na poslední snímek prezentace. Je-li k dispozici dodatek, tak na poslední snímek dodatku.

5.3.4.2 Pozdější opakování rámce

Občas si budete přát, aby se některé snímky daného rámce zobrazily v hlavní prezentaci, avšak jiné, "doplňkové", snímky budete chtít zobrazit pouze v dodatku. V takovém případě využijete příkazy `\againframe`.

```
\againframe<"overlay specification">[<"default overlay specification">] ["options"] {"name"}
```

Příklad:

```
\frame<1-2>[label=myframe]
{
  \begin{itemize}
  \item<alert@1> První subjekt.
  \item<alert@2> Druhý subjekt.
  \item<alert@3> Třetí subjekt.
  \end{itemize}
}
```

```
\frame
{
  Nějaké informace s vysvětlivkami k druhé problematice.
}
\againframe<3>{myframe}
```

Výše uvedený kód vytvoří čtyři snímky. Na prvních dvou snímcích se zvýrazní položky 1 a 2. Třetí snímek obsahuje text "Nějaké informace s vysvětlivkami k druhé problematice." Čtvrtý snímek vypadá stejně jako první dva snímky s výjimkou toho, že je zvýrazněný třetí bod.

Příklad:

```

\frame<1>[label=Cantor]
{
\frametitle{Hlavní teorém}

\begin{Theorem}
 $\alpha < 2^\alpha$  pro všechny řadové číslovky  $\alpha$ .
\end{Theorem}

\begin{overprint}
\onslide<1>
\hyperlink{Cantor<2>}{\beamergotobutton{Podrobnosti důkazu}}

\onslide<2->
% toto se zobrazí pouze v dodatku, v němž se v tomto snímku
pokračuje.
\begin{proof}
Jak ukazuje Cantor...
\end{proof}

\hfill\hyperlink{Cantor<1>}{\beamerreturnbutton{Návrat}}
\end{overprint}
}

...
\appendix

\againframe<2>{Cantor}

```

V tomto příkladu jsou podrobnosti důkazu odloženy až na snímek v dodatku. Odkazy jsou nastaveny, a můžete proto odskočit na důkaz a zpět.

5.3.4.3 Přidání předpokládaného zvětšení

Předpokládané zvětšení je nutné používat v případě, že pracujete s velmi komplikovaným obrázkem, který nehodláte zjednodušit, protože chcete vy-

světlit všechny jeho složité detaily. V tomto případě použijte příkaz `\framezoom`.

Pomocí tohoto příkazu můžete zadat, že po kliknutí na určitou oblast rámce by se tato oblast měla zvětšit. Poté můžete vysvětlit všechny detaily. Kliknutím na zvětšený obrázek se vrátíte zpět k původní grafice.

```
\framezoom<"button overlay specification"><"zoomed overlay
specification">["options"]
("upper left x", "upper left y")("zoom area width", "zoom
area depth")
```

Tento příkaz byste měli zadat někde na začátku rámce. Jakmile jej zadáte, dojde ke dvěma skutečnostem, a to v závislosti na tom, zda se specifikace překrytí *"button overlay specification"* vztahuje na aktuální snímek daného rámce nebo zda je platná zvětšená specifikace překrytí *"zoomed overlay specification"*. Tyto specifikace překrytí by se neměly překrývat.

Platí-li specifikace překrytí *"button overlay specification"*, vytvoří se uvnitř rámce klikatelná oblast. Velikost této oblasti je zadána volbami *"zoom area width"* (šířka) a *"zoom area depth"* (výška), což jsou běžné jednotky velikosti \TeX (například 1cm nebo 20pt). Horní levý roh této oblasti se zadává volbou *"upper left x"* a *"upper left y"*, což jsou také jednotky velikosti \TeX . Tyto rozměry se zadávají relativně k místu, kde by se vysázel první řádek běžného textu daného rámce. Umístění (0pt, 0pt) je tedy na začátku běžného textu (což se nevztahuje na záhlaví ani na titul rámce).

Standardně je možné na tlačítko kliknout, avšak uživateli se to nikterak nesignalizuje. Pomocí níže uvedených voleb *"options"* můžete kolem tlačítka vykreslit okraj:

- `border="width in pixels"` vykreslí okraj kolem specifikované oblasti tlačítka. Výchozí šířkou je 1 pixel. Barvou tohoto tlačítka je `linkbordercolor` příkazu `hyperref`. Třída BEAMER standardně nastaví tuto barvu na 50% šedí. To můžete změnit příkazem `\hypersetup{linkbordercolor={ "red" "green" "blue" }}`, kde *"red"* *"green"* a *"blue"* jsou hodnotami mezi 0 a 1.

Stisknete-li takto vytvořené tlačítko, aplikace prohlížeče odskočí na první rámeček zadaný specifikací *"zoomed overlay specification"*. U snímků, na něž se vztahuje tato specifikace překrytí, dojde k níže uvedeným akcím:

- Naprosto stejná oblast, již jste zadali dříve, se "zvětší" a vyplní oblast s běžným textem rámce. Vše ostatní, a to včetně postranních panelů, záhlaví a zápatí a dokonce i titulu rámce, si uchová běžnou velikost.

Zvětšení se provede tak, že se zobrazí úplně celá zadaná oblast. Zachová se správný poměr stran, a ve zvětšené oblasti se tak mohou zobrazit další části, které se nacházejí mimo zadanou oblast, nesouhlasí-li poměr stran této oblasti s poměrem stran dostupné textové oblasti.

- Za tuto celou textovou oblast (jež obsahuje zvětšenou oblast) se vloží neviditelné tlačítko "Zpět". Kliknutím v libovolném místě textové oblasti se proto vrátíte k původnímu (nezvětšenému) obrázku.

U daného snímku můžete zadat několik oblastí zvětšení. V tomto případě byste měli zadat různé specifikace zvětšení *"zoomed overlay specification"*, avšak můžete zadat stejnou specifikaci *"button overlay specification"*. Zvětšení nelze skládat do sebe, tj. nelze mít tlačítko zvětšení na snímku, který je v nějaké specifikaci zvětšení *"zoomed overlay specification"*. Specifikace se však mohou překrývat a dokonce do sebe můžete skládat specifikace překrytí *"button overlay specification"*. Kliknete-li na oblast, jež náleží několika tlačítkům, tak tlačítko, které jste zadali nejpozději, "vyhraje" (mělo by tedy být tím nejmenším tlačítkem).

Nechcete-li, aby se na zvětšeném snímku zobrazoval titul rámce, můžete k příkazu `\frametitle` přidat specifikaci překrytí, jež na daném snímku jednoduše potlačí jeho titul. Použijete-li kromě toho volbu `plain`, může zvětšený snímek vyplnit celou stránku:

Příklad: Jednoduchý případ

```
\begin{frame}
\frametitle{Komplikovaný obrázek}

\framezoom<1><2>(0cm,0cm)(2cm,1.5cm)
\framezoom<1><3>(1cm,3cm)(2cm,1.5cm)
\framezoom<1><4>(3cm,2cm)(3cm,2cm)

\pgfimage[height=8cm]{complicatedimagefilename}
\end{frame}
```

Příklad: Složitější případ, v němž zvětšené části úplně zaplní rámce.

```
\begin{frame}<1>[label=zooms]
\frametitle<1>{Komplikovaný obrázek}

\framezoom<1><2>[border](0cm,0cm)(2cm,1.5cm)
\framezoom<1><3>[border](1cm,3cm)(2cm,1.5cm)
\framezoom<1><4>[border](3cm,2cm)(3cm,2cm)
```

```
\pgfimage[height=8cm]{complicatedimagefilename}  
\end{frame}  
\againframe<2->[plain]{zooms}
```

Kapitola 6

Závěr

V této práci jsem nejprve pro úplnost na úvod stručně představil historii a základní principy sázecího program $\text{T}_{\text{E}}\text{X}$, jeho nadstavby $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ a několik používaných tříd k prezentačním účelům.

Poté jsem uvedl užitečná pravidla pro strukturování prezentací a rady pro používání grafiky či fontů při jejich tvorbě, nejenom pro třídu BEAMER.

Nejobtavnější a na zpracování nejnáročnější část se stává z popisu příkazů a jejich atributů specifických pouze pro tuto třídu, v kterých bylo třeba se nejprve zorientovat. Tyto jsem sepsal za použití veřejně dostupných, zejména anglických pramenů, které jsem vždy uvedl před, případně za danými kapitolami.

Součástí celé práce jsou i názorné příklady kódu, které ukazují praktické použití jednotlivých příkazů a jejich možností. Samotná implementace příkazů je díky tomu potom podstatně jednodušší.

Svou práci jsem zpracoval systémem $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ a prezentaci k obhajobě vytvořil ve třídě BEAMER. Některé části této práce, může sloužit jako ucelený základ pro publikace u nás, zabývající se důležitými rozšiřujícími třídami $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ u a které by měl být BEAMER, dle mého názoru, pevnou součástí.

Literatura

- [1] RYBIČKA, Jiří. *ĚT_EX pro začátečníky*, 3.vydání. Brno : Konvoj, 2003. 238 s. ISBN 80-7302-049-1.

- [2] KOPKA, Helmut; DALY, Patrick W. *ĚT_EX Podrobný průvodce*. Brno : Computer Press, 2004. 576 s. ISBN 80-722-6973-9.

- [3] OLŠÁK, Petr. *Typografický systém T_EX*. 2. vydání, Brno: Konvoj, 2000, 300 s. ISBN 80-85615-91-6

- [4] OLŠÁK, Petr. *T_EXbook naruby*. 2. vyd. Brno : Konvoj, 2001. 468 s. ISBN 80-7302-007-6.

- [5] PARTL, Hubert et al. *Ne příliš stručný úvod do systému ĚT_EX 2_ε: Neboli ĚT_EX 2_ε v 73 minutách*. Michal Kočer. [s.l.] : [s.n.], 1998. 85 s. Dostupné z WWW: <<http://www.penguin.cz/~kocer/texty/lshort2e/lshort2e-cz.pdf>>.

- [6] ZELENKA, Petr. *Abelinuxu.cz* [online]. 2006-08-28 [cit. 2011-03-11]. *Beamer: ĚT_EX na prezentace*. Dostupné z WWW: <<http://www.abelinuxu.cz/clanky/navody/beamer-latex-na-prezentace>>. ISSN 1214-1267.

- [7] VAVŘÍČEK, Jan. *ĚT_EX* [online]. akt. 2010-03-10 [cit. 2011-03-11]. Dostupné z WWW: <<http://vavricek.cs.vsb.cz/index.php/LaTeX>>.

- [8] TANTAU, Till. *The Beamer class* [online]. [s.l.]: [s.n.]2007-03-11 [cit. 2011-04-15]. Dostupné z WWW:

<gking.harvard.edu/files/beameruserguide2.pdf>.

- [9] *TeX Users Group web site* [online]. 2009-04-04 [cit. 2011-03-10]. The history of \TeX . Dostupné z WWW: <<http://www.tug.org/whatis.html>>.
- [10] ŠÍN, Martin. *Linuxexpress.cz* [online]. 2007-06-07 [cit. 2011-03-11]. Dostupné z WWW: <<http://www.linuxexpres.cz/software/beamer-prezentace-v-latexu>>. ISSN 1801-3996.
- [11] MEDEK, Ondřej. *Root.cz* [online]. 2005-02-17 [cit. 2011-03-11]. Beamer - snadná a profesionální prezentace v (pdf) \LaTeX u. Dostupné z WWW: <<http://www.root.cz/clanky/beamer-prezentace-v-pdflatexu/>>.
- [12] ŘÍHA, Jozef; STRÍŽ, Pavel, . *Prezentačný software pre \LaTeX* . In *Zpravodaj Československého sdružení uživatelů \TeX u* [online]. [s.l.]: [s.n.]2007 [cit. 2011-03-11]. s. 52. Roč. 17, č. 2. Dostupné z WWW: <bulletin.cstug.cz/pdf/bul_072.pdf>. ISSN 1213-8185.

Přílohy

Zdrojové kódy jednotlivých rejstříků
Zdrojové kódy práce přiložené na CD

Zdrojový kód jednoduché prezentace

```
\documentclass[xcolor=dvipsnames]{beamer}
\usepackage[czech]{babel}
\usepackage[utf8]{inputenc}
\usepackage{times}
\usepackage[T1]{fontenc}
\usecolortheme{sidebartab}

\usetheme{PaloAlto}

\title[Tvorba prezentace v BEAMERu]{Tvorba prezentace v~\LaTeX u
ve třídě BEAMER}
\author{Josef Stern}\\
\institute[JČU]{Jihočeská univerzita v Českých Budějovicích}
\date{3. 6. 2011}

\pgfdeclareimage[height=1.55cm]{university-logo}{logo_jcu.jpg}
\logo{\pgfuseimage{university-logo}}

\begin{document}

\begin{frame}
\titlepage
\end{frame}

\begin{frame}{Přehled}
\tableofcontents[pausesections]
\end{frame}

\section{Základní pojmy}

\subsection{\TeX}

\begin{frame}{Co je to \TeX?}
\begin{block}{\TeX}
\begin{itemize}
\item <1-| alert@1>formátovací program pro zpracování textu
\item <2-| alert@2>sestavil Donald Ervin Knuth
\item <3-| alert@3>absence kvalitního sázecího systému\\
(zejména pro matematické výrazy)
```

```

\item <4-| alert@4>rozumí pouze sadě elementárních příkazů\\
(tzv. primitivů, počet okolo 300)
\item <5-| alert@5>na jejich základě lze definovat
komplexnější příkazy na vyšší úrovni - makra
\item <6-| alert@6>souhrnem (balíkem) maker je formát
\end{itemize}
\end{block}
\end{frame}

```

```

\subsection{\LaTeX}
\begin{frame}{Co je to \LaTeX?}
\begin{block}{\LaTeX}
\begin{itemize}
\item <1-| alert@1>je nejznámější a nejpoužívanější
formát \TeX u
\item <2-| alert@2>sestavil Leslie Lamport
\item <3-| alert@3>dovoluje tvorbu komplexních dokumentů
\item <4-| alert@4>uživatelsky přívětivý
\item <5-| alert@5>obsahuje třídy, které definují volby pro
specifické druhy textů
\item <6-| alert@6>příkazy pro tyto texty jsou uloženy v~
doplňkových souborech
\end{itemize}
\end{block}
\end{frame}

```

```

\subsection{BEAMER}
\begin{frame}{Co je to BEAMER?}
\begin{block}{BEAMER}
\begin{itemize}
\item <1-| alert@1>je třída formátu \LaTeX
\item <2-| alert@2>sestavil Till Tantau
\item <3-| alert@3>slouží k vytváření prezentací
\item <4-| alert@4>funguje na systému WYSIWYM
\item <5-| alert@5>je předmětem mé bakalářské práce
\end{itemize}
\end{block}
\end{frame}

```

```

\section{Cíle práce}
\begin{frame}{Cíle práce}

```

```

\begin{block}{Cíle práce}
  \begin{itemize}
    \item <1-| alert@1>stručně představit systémy
      \TeX~a~\LaTeX
    \item <2-| alert@2>popsat tvorbu prezentace v BEAMERu
    \item <3-| alert@3>shrnout příkazy třídy BEAMER
    \item <4-| alert@4>ukázat příklady použití
    \item <5-| alert@5>tato prezentace musí být vytvořena
      pomocí BEAMERu
    \item <6-| alert@6>nad rámec zadání: práce bude
      vytvořena v~\LaTeX u
  \end{itemize}
\end{block}
\end{frame}

\section{Metodika}
\subsection{Instalace}
\begin{frame}{Metodický postup}
  \begin{block}{Instalace}
    \begin{itemize}
      \item <alert@1>Instalace potřebných součástí:\\~
      \pause
      \begin{enumerate}
        \item <2-| alert@2>OS Linux, distribuce
          Ubuntu 10.10\\~
        \item <3-| alert@3>program \TeX~s nadstavbou
          \LaTeX , \\ distribuce \TeX Live, ver. 2009\\~
        \item <4-| alert@4>\LaTeX editor - multiplatformní
          \TeX Maker, ver. 2.0\\~
        \item <5-| alert@5>Beamer - součástí formátu
          \LaTeX, ver. 3.07\\~
      \end{enumerate}
    \end{itemize}
  \end{block}
\end{frame}

\subsection{Zdroje}
\begin{frame}{Metodický postup}
  \begin{block}{Literatura, zdroje informací}
    \begin{itemize}
      \item <alert@1>Tištěná literatura
    \end{itemize}
  \end{block}
\end{frame}

```

```

\pause
\begin{itemize}
  \item <2-| alert@2>o \LaTeX u u nás 2 knihy
  \item <3-| alert@3>o \TeX u u nás 1 kniha
  \item <4-| alert@4>o BEAMERu - v českém jazyce není
\end{itemize}
\item <5-| alert@5>Ostatní zdroje (elektronické)
\pause
\begin{itemize}
  \item <6-| alert@6>v češtině pouze krátké návody
    na rychlé prezentace\\ root.cz, linuxexpres.cz,
    abclinuxu.cz
  \item <7-| alert@7>CSTUG.cz - krátký report
    o BEAMERu
  \item <8-| alert@8>Anglická dokumentace
    k třídě BEAMER
  \item <9-| alert@9>další webové stránky
    v angličtině\\ (krátké návody, historie)
\end{itemize}
\end{itemize}
\end{block}
\end{frame}

\subsection{Práce}
\begin{frame}{Metodický postup}
\begin{block}{Vlastní práce}
\begin{itemize}
  \item <1-| alert@1>instalace potřebných součástí
    (viz. instalace)
  \item <2-| alert@2>seznamování se s \LaTeX em a~
    (pro mě) neznámou\\ třídou BEAMER
  \item <3-| alert@3>analýza a výběr informací ze zdrojů\\
    komparační metodou
  \item <4-| alert@4>překlad, zpracování a zakomponování
    do práce
  \item <5-| alert@5>testování příkazů
  \item <6-| alert@6>zdokumentování názorných příkladů,\\
    zapsání funkčních kódů do práce
  \item <7-| alert@7>vysázení celé práce systémem \LaTeX \\
    (první vlastní dokument)
\end{itemize}

```

```
\end{block}
\end{frame}

\section{Výsledky}
\begin{frame}{Výsledky}
\begin{block}{Výsledky práce}

\begin{itemize}
\item <1-| alert@1>seznámení s historií a principy
TeX u a jeho nadstaveb \pause
\item <2-| alert@2>postupy a rady pro strukturování
prezentace \pause
\item <3-| alert@3>popis příkazů třídy
BEAMER \pause
\item <4-| alert@4>v práci k příkazům
příklady a ukázky \pause
\item <5-| alert@5>práce vysázená v~\LaTeX u \pause
\item <6-| alert@6>tato prezentace
je vytvořena v BEAMERu \pause
\end{itemize}
\end{block}
\end{frame}

\section{Závěr}
\begin{frame}{Závěr}
\begin{block}{Několik slov závěrem}
\begin{itemize}
\item <1-| alert@1>objem práce\\~
\item <2-| alert@2>doba práce\\~
\item <3-| alert@3>hlavní vytyčené
cíle jsou splněny\\~
\end{itemize}
\end{block}
\end{frame}

\begin{frame}{Poděkování}
\begin{center}
\textbf{Děkuji za pozornost!}
\end{center}
\end{frame}
\end{document}
```