

Typografický systém T_EX

Petr Hanuš (Píta)



Brkosí zimmění 2009

- Typografický systém s velmi kvalitní sazbou

Co je to T_EX?

- Typografický systém s velmi kvalitní sazbou
- Makroprocesor nabízející uživatelům mocné programovací schopnosti, které umožňují proces sazby výrazně ovlivňovat

Co je to T_EX?

- Typografický systém s velmi kvalitní sazbou
- Makroprocesor nabízející uživatelům mocné programovací schopnosti, které umožňují proces sazby výrazně ovlivňovat
- Zřejmě jediný systém, který v dnešní době může zaručit, že dokument bude možné ze stejných podkladů vysázet i za padesát let

Co je to T_EX?

- Typografický systém s velmi kvalitní sazbou
- Makroprocesor nabízející uživatelům mocné programovací schopnosti, které umožňují proces sazby výrazně ovlivňovat
- Zřejmě jediný systém, který v dnešní době může zaručit, že dokument bude možné ze stejných podkladů vysázet i za padesát let
- Autor Donald E. Knuth o něm říká: „Je to nástroj určený pro tvorbu hezkých knih – a obzvláště knih obsahujících spoustu matematiky“

- Editor

Co T_EX není?

- Editor
- „Kanón na vrabce“ – nevhodný nástroj pro malé dokumenty (vzkazy, letáky apod.)

- Editor
- „Kanón na vrabce“ – nevhodný nástroj pro malé dokumenty (vzkazy, letáky apod.)
- Program typu WYSIWYG (existují však nadstavby jako např. LyX)

- Editor
- „Kanón na vrabce“ – nevhodný nástroj pro malé dokumenty (vzkazy, letáky apod.)
- Program typu WYSIWYG (existují však nadstavby jako např. LyX)
- Nástroj pro zpracování grafiky (k tomu je určen komplementární program METAFONT)

- Editor
- „Kanón na vrabce“ – nevhodný nástroj pro malé dokumenty (vzkazy, letáky apod.)
- Program typu WYSIWYG (existují však nadstavby jako např. LyX)
- Nástroj pro zpracování grafiky (k tomu je určen komplementární program METAFONT)
- Program, který se ovládá myší :-)

- Výborný typografický výstup

Proč psát v T_EXu?

- Výborný typografický výstup
- Vynikající nástroj pro zpracování matematických výrazů

Proč psát v T_EXu?

- Výborný typografický výstup
- Vynikající nástroj pro zpracování matematických výrazů
- Zdrojový kód si uživatel může napsat ve svém oblíbeném editoru

Proč psát v T_EXu?

- Výborný typografický výstup
- Vynikající nástroj pro zpracování matematických výrazů
- Zdrojový kód si uživatel může napsat ve svém oblíbeném editoru
- Každý si může definovat své vlastní příkazy

Proč psát v T_EXu?

- Výborný typografický výstup
- Vynikající nástroj pro zpracování matematických výrazů
- Zdrojový kód si uživatel může napsat ve svém oblíbeném editoru
- Každý si může definovat své vlastní příkazy
- Je volně šiřitelný

V roce 1977 měl prof. Donald E. Knuth v úmyslu vydat knihu *The Art of Computer Programming*. Když však zjistil, že sazba není příliš a kvalitní a že lepší způsob neexistuje, rozhodl se, že vytvoří nový sazecí systém.

V roce 1977 měl prof. Donald E. Knuth v úmyslu vydat knihu *The Art of Computer Programming*. Když však zjistil, že sazba není příliš a kvalitní a že lepší způsob neexistuje, rozhodl se, že vytvoří nový sázecí systém.

Z rok už se T_EX dostal k prvním uživatelům. Vývoj pokračoval a vznikaly tak další verze. S verzí 3.0 se však vývoj z rozhodnutí Donalda Knutha zastavil. Každé další vydání pouze opravuje nalezené chyby a jeho číslo je doplněno o jednu další číslici tak, že konverguje k číslu π . Aktuální verze je 3.1415926.

V roce 1977 měl prof. Donald E. Knuth v úmyslu vydat knihu *The Art of Computer Programming*. Když však zjistil, že sazba není příliš a kvalitní a že lepší způsob neexistuje, rozhodl se, že vytvoří nový sázecí systém.

Z rok už se T_EX dostal k prvním uživatelům. Vývoj pokračoval a vznikaly tak další verze. S verzí 3.0 se však vývoj z rozhodnutí Donalda Knutha zastavil. Každé další vydání pouze opravuje nalezené chyby a jeho číslo je doplněno o jednu další číslici tak, že konverguje k číslu π . Aktuální verze je 3.1415926.

Knuth se dokonce nechal slyšet, že si přeje, aby po jeho smrti nebyly prováděny žádné další úpravy v jeho zdrojovém kódu. Verze, která v té době bude aktuální, se stane verzí finální a bude se označovat číslem π .

Vznik a vývoj (2)

Sám o sobě je $\text{T}_{\text{E}}\text{X}$ natolik obtížný, že Knuth napsal balík maker pro použití s $\text{T}_{\text{E}}\text{X}$ em, který nazval $\text{PlainT}_{\text{E}}\text{X}$. Využití veškerého potencionálu $\text{PlainT}_{\text{E}}\text{X}$ u však vyžaduje značné programátorské schopnosti, a proto v roce 1985 Leslie Lamport sestavil sadu maker s názvem $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$. Pomocí tohoto systému pro zpracování dokumentu může i uživatel bez znalostí sazby či programování využívat rozsáhlých výhod nabízených programem $\text{T}_{\text{E}}\text{X}$ a je schopný v krátkém čase vytvořit rozmanité textové výstupy v knižní kvalitě.

Sám o sobě je $\text{T}_{\text{E}}\text{X}$ natolik obtížný, že Knuth napsal balík maker pro použití s $\text{T}_{\text{E}}\text{X}$ em, který nazval $\text{PlainT}_{\text{E}}\text{X}$. Využití veškerého potencionálu $\text{PlainT}_{\text{E}}\text{X}$ u však vyžaduje značné programátorské schopnosti, a proto v roce 1985 Leslie Lamport sestavil sadu maker s názvem $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$. Pomocí tohoto systému pro zpracování dokumentu může i uživatel bez znalostí sazby či programování využívat rozsáhlých výhod nabízených programem $\text{T}_{\text{E}}\text{X}$ a je schopný v krátkém čase vytvořit rozmanité textové výstupy v knižní kvalitě.

Zdokonalování $\text{T}_{\text{E}}\text{X}$ u došlo dokonce tak daleko, že Donald E. Knuth nabízí odměnu tomu, kdo by v jeho zdrojovém kódu našel chybu. Pokud by se Vám to podařilo, získali byste od něho odměnu, která v současné době činí \$327,68.

Jaký formát zvolit?

- Tato otázka nemá jednoznačnou odpověď

Jaký formát zvolit?

- Tato otázka nemá jednoznačnou odpověď
- Rozdíly jsou pouze v tom, co který formát uživateli nabízí, a jak snadno lze defaultní chování modifikovat

Jaký formát zvolit?

- Tato otázka nemá jednoznačnou odpověď
- Rozdíly jsou pouze v tom, co který formát uživateli nabízí, a jak snadno lze defaultní chování modifikovat

1 plainT_EX

Jaký formát zvolit?

- Tato otázka nemá jednoznačnou odpověď
 - Rozdíly jsou pouze v tom, co který formát uživateli nabízí, a jak snadno lze defaultní chování modifikovat
- 1 **plainTeX**
- Poskytuje pouze základní možnosti (vše ostatní si musí uživatel nadefinovat sám)

Jaký formát zvolit?

- Tato otázka nemá jednoznačnou odpověď
- Rozdíly jsou pouze v tom, co který formát uživateli nabízí, a jak snadno lze defaultní chování modifikovat

1 plainTeX

- Poskytuje pouze základní možnosti (vše ostatní si musí uživatel nadefinovat sám)
- Je tedy vhodný především pro uživatele s duší programátora

Jaký formát zvolit?

- Tato otázka nemá jednoznačnou odpověď
- Rozdíly jsou pouze v tom, co který formát uživateli nabízí, a jak snadno lze defaultní chování modifikovat

1 plain $\text{T}_{\text{E}}\text{X}$

- Poskytuje pouze základní možnosti (vše ostatní si musí uživatel nadefinovat sám)
- Je tedy vhodný především pro uživatele s duší programátora

2 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

Jaký formát zvolit?

- Tato otázka nemá jednoznačnou odpověď
- Rozdíly jsou pouze v tom, co který formát uživateli nabízí, a jak snadno lze defaultní chování modifikovat

1 plain $\text{T}_\text{E}\text{X}$

- Poskytuje pouze základní možnosti (vše ostatní si musí uživatel nadefinovat sám)
- Je tedy vhodný především pro uživatele s duší programátora

2 $\text{L}\text{A}\text{T}_\text{E}\text{X}$

- Poskytuje sadu maker pro vytváření komplexních dokumentů

Jaký formát zvolit?

- Tato otázka nemá jednoznačnou odpověď
- Rozdíly jsou pouze v tom, co který formát uživateli nabízí, a jak snadno lze defaultní chování modifikovat

1 plain $\text{T}_{\text{E}}\text{X}$

- Poskytuje pouze základní možnosti (vše ostatní si musí uživatel nadefinovat sám)
- Je tedy vhodný především pro uživatele s duší programátora

2 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$

- Poskytuje sadu maker pro vytváření komplexních dokumentů
- Vhodný i pro uživatele bez znalostí sazby či programování

Jaký formát zvolit?

- Tato otázka nemá jednoznačnou odpověď
- Rozdíly jsou pouze v tom, co který formát uživateli nabízí, a jak snadno lze defaultní chování modifikovat

1 plain $\text{T}_{\text{E}}\text{X}$

- Poskytuje pouze základní možnosti (vše ostatní si musí uživatel nadefinovat sám)
- Je tedy vhodný především pro uživatele s duší programátora

2 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$

- Poskytuje sadu maker pro vytváření komplexních dokumentů
- Vhodný i pro uživatele bez znalostí sazby či programování

3 Další formáty

Jaký formát zvolit?

- Tato otázka nemá jednoznačnou odpověď
- Rozdíly jsou pouze v tom, co který formát uživateli nabízí, a jak snadno lze defaultní chování modifikovat

1 plainTeX

- Poskytuje pouze základní možnosti (vše ostatní si musí uživatel nadefinovat sám)
- Je tedy vhodný především pro uživatele s duší programátora

2 L^AT_EX

- Poskytuje sadu maker pro vytváření komplexních dokumentů
- Vhodný i pro uživatele bez znalostí sazby či programování

3 Další formáty

- AmsTeX

Jaký formát zvolit?

- Tato otázka nemá jednoznačnou odpověď
- Rozdíly jsou pouze v tom, co který formát uživateli nabízí, a jak snadno lze defaultní chování modifikovat

1 plain $\text{T}_\text{E}\text{X}$

- Poskytuje pouze základní možnosti (vše ostatní si musí uživatel nadefinovat sám)
- Je tedy vhodný především pro uživatele s duší programátora

2 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$

- Poskytuje sadu maker pro vytváření komplexních dokumentů
- Vhodný i pro uživatele bez znalostí sazby či programování

3 Další formáty

- Ams $\text{T}_\text{E}\text{X}$
- Sli $\text{T}_\text{E}\text{X}$

Jaký formát zvolit?

- Tato otázka nemá jednoznačnou odpověď
- Rozdíly jsou pouze v tom, co který formát uživateli nabízí, a jak snadno lze defaultní chování modifikovat

1 plain $\text{T}_{\text{E}}\text{X}$

- Poskytuje pouze základní možnosti (vše ostatní si musí uživatel nadefinovat sám)
- Je tedy vhodný především pro uživatele s duší programátora

2 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$

- Poskytuje sadu maker pro vytváření komplexních dokumentů
- Vhodný i pro uživatele bez znalostí sazby či programování

3 Další formáty

- Ams $\text{T}_{\text{E}}\text{X}$
- Sli $\text{T}_{\text{E}}\text{X}$
- ...

- Každý \LaTeX ový dokument (zdrojový text s příkazy) musí obsahovat preambuli a tělo

- Každý \LaTeX ový dokument (zdrojový text s příkazy) musí obsahovat preambuli a tělo
- Preambule je sekvence příkazů, které specifikují globální parametry pro zpracování textu (např.: formát papíru, výška a šířka textu, vzhled výstupní stránky,...)

- Každý \LaTeX ový dokument (zdrojový text s příkazy) musí obsahovat preambuli a tělo
- Preambule je sekvence příkazů, které specifikují globální parametry pro zpracování textu (např.: formát papíru, výška a šířka textu, vzhled výstupní stránky,...)
- Všechny příkazy začínají znakem \backslash (zpětné lomítko), za nímž následuje jméno příkazu

- Každý \LaTeX ový dokument (zdrojový text s příkazy) musí obsahovat preambuli a tělo
- Preambule je sekvence příkazů, které specifikují globální parametry pro zpracování textu (např.: formát papíru, výška a šířka textu, vzhled výstupní stránky,...)
- Všechny příkazy začínají znakem \backslash (zpětné lomítko), za nímž následuje jméno příkazu
- Pokud příkaz umožňuje zadání parametrů, píší se do hranatých závorek, jedná-li se o parametry nepovinné, a do závorek složených, pokud jde o parametry povinné

- Každý \LaTeX ový dokument (zdrojový text s příkazy) musí obsahovat preambuli a tělo
- Preambule je sekvence příkazů, které specifikují globální parametry pro zpracování textu (např.: formát papíru, výška a šířka textu, vzhled výstupní stránky,...)
- Všechny příkazy začínají znakem `\` (zpětné lomítko), za nímž následuje jméno příkazu
- Pokud příkaz umožňuje zadání parametrů, píší se do hranatých závorek, jedná-li se o parametry nepovinné, a do závorek složených, pokud jde o parametry povinné
- Prvním příkazem v dokumentu je informace o stylu (třídě) psaného dokumentu

- Každý \LaTeX ový dokument (zdrojový text s příkazy) musí obsahovat preambuli a tělo
- Preambule je sekvence příkazů, které specifikují globální parametry pro zpracování textu (např.: formát papíru, výška a šířka textu, vzhled výstupní stránky,...)
- Všechny příkazy začínají znakem `\` (zpětné lomítko), za nímž následuje jméno příkazu
- Pokud příkaz umožňuje zadání parametrů, píší se do hranatých závorek, jedná-li se o parametry nepovinné, a do závorek složených, pokud jde o parametry povinné
- Prvním příkazem v dokumentu je informace o stylu (třídě) psaného dokumentu
- Za tímto úvodním příkazem většinou následují příkazy pro zavedení balíčků

- Pro zavedení balíčku se používá příkaz `\usepackage` s tím, že jako povinný parametr se uvádí jméno balíčku, který chceme zavést, jako nepovinný parametr pak nějaká bližší specifikace

Struktura dokumentu v L^AT_EXu (2)

- Pro zavedení balíčku se používá příkaz `\usepackage` s tím, že jako povinný parametr se uvádí jméno balíčku, který chceme zavést, jako nepovinný parametr pak nějaká bližší specifikace
- Vlastní text je uvozen příkazem `\begin{document}` a ukončen `\end{document}`

- Pro zavedení balíčku se používá příkaz `\usepackage` s tím, že jako povinný parametr se uvádí jméno balíčku, který chceme zavést, jako nepovinný parametr pak nějaká bližší specifikace
- Vlastní text je uvozen příkazem `\begin{document}` a ukončen `\end{document}`
- Mezi tyto dva příkazy se vepíše text, který může být formátován příkazy

- Pro zavedení balíčku se používá příkaz `\usepackage` s tím, že jako povinný parametr se uvádí jméno balíčku, který chceme zavést, jako nepovinný parametr pak nějaká bližší specifikace
- Vlastní text je uvozen příkazem `\begin{document}` a ukončen `\end{document}`
- Mezi tyto dva příkazy se vepíše text, který může být formátován příkazy

Ukázka programu

```
\documentclass[11pt]{article}
\usepackage[cp1250]{inputenc}      %vstupní kódování dokumentu
\usepackage[czech]{babel}         %dokument bude psán v češtině
\begin{document}
  hello world...
\end{document}
```

- Příkaz představuje instrukci udávající, že se provede něco specifického (např. vysázel speciální symbol)

- Příkaz představuje instrukci udávající, že se provede něco specifického (např. vysázel speciální symbol)
- Rozlišujeme tři typy příkazů

- Příkaz představuje instrukci udávající, že se provede něco specifického (např. vysázel speciální symbol)
- Rozlišujeme tři typy příkazů
 - Samostatné znaky # \$ & ~ _ ^ % { } z nichž každý má svůj specifický význam

- Příkaz představuje instrukci udávající, že se provede něco specifického (např. vysázel speciální symbol)
- Rozlišujeme tři typy příkazů
 - Samostatné znaky # \$ & ~ _ ^ % { } z nichž každý má svůj specifický význam
 - Zpětné lomítko \ plus jediný znak, který není písmeno (např. \\$ vysází \$)

- Příkaz představuje instrukci udávající, že se provede něco specifického (např. vysázel speciální symbol)
- Rozlišujeme tři typy příkazů
 - Samostatné znaky # \$ & ~ _ ^ % { } z nichž každý má svůj specifický význam
 - Zpětné lomítko \ plus jediný znak, který není písmeno (např. \\$ vysází \$)
 - Zpětné lomítko \ plus posloupnost písmen, která končí prvním nepísmenným znakem (např. příkaz \TeX vysází logo T_EX)

- Příkaz představuje instrukci udávající, že se provede něco specifického (např. vysázel speciální symbol)
- Rozlišujeme tři typy příkazů
 - Samostatné znaky # \$ & ~ _ ^ % { } z nichž každý má svůj specifický význam
 - Zpětné lomítko \ plus jediný znak, který není písmeno (např. \\$ vysází \$)
 - Zpětné lomítko \ plus posloupnost písmen, která končí prvním nepísmenným znakem (např. příkaz \TeX vysází logo T_EX)
- Povinným parametrům říkáme argumenty

- Uvozuje příkazem `\begin{prostředí}` a ukončen `\end{prostředí}`

- Uvozuje příkazem `\begin{prostředí}` a ukončen `\end{prostředí}`
- Text uvnitř prostředí se zpracuje jiným způsobem než ostatní text, a to v závislosti na parametrech prostředí

- Uvozuje příkazem `\begin{prostředí}` a ukončen `\end{prostředí}`
- Text uvnitř prostředí se zpracuje jiným způsobem než ostatní text, a to v závislosti na parametrech prostředí
- Zvláštním případem prostředí je tzv. skupina, která se zapisuje pomocí páru složených závorek `{...}`

- Uvozuje příkazem `\begin{prostředí}` a ukončen `\end{prostředí}`
- Text uvnitř prostředí se zpracuje jiným způsobem než ostatní text, a to v závislosti na parametrech prostředí
- Zvláštním případem prostředí je tzv. skupina, která se zapisuje pomocí páru složených závorek `{...}`
- Platnost jakéhokoliv příkazu umístěného uvnitř prostředí končí příkazem `\end{prostředí}`, resp. uzavírající složenou závorkou

- Příkaz, který mění hodnoty nebo význam určitých parametrů nebo příkazů

- Příkaz, který mění hodnoty nebo význam určitých parametrů nebo příkazů
- Neslouží k sázení textu

- Příkaz, který mění hodnoty nebo význam určitých parametrů nebo příkazů
- Neslouží k sázení textu
- Platnost deklaráce začíná v okamžiku jejího uvedení a končí, jakmile se vyskytne jiná deklaráce stejného typu

- Příkaz, který mění hodnoty nebo význam určitých parametrů nebo příkazů
- Neslouží k sázení textu
- Platnost deklaráce začíná v okamžiku jejího uvedení a končí, jakmile se vyskytne jiná deklaráce stejného typu
- Vyskytuje-li se deklaráce uvnitř prostředí či skupiny, její platnost se rozpíná pouze po odpovídající příkaz `\end`, resp. uzavírající složenou závorku `}`

- Příkaz, který mění hodnoty nebo význam určitých parametrů nebo příkazů
- Neslouží k sázení textu
- Platnost deklaráce začíná v okamžiku jejího uvedení a končí, jakmile se vyskytne jiná deklaráce stejného typu
- Vyskytuje-li se deklaráce uvnitř prostředí či skupiny, její platnost se rozpíná pouze po odpovídající příkaz `\end`, resp. uzavírající složenou závorku `}`
- Zvláštním případem jsou globální deklaráce, které nejsou omezeny na aktuální prostředí

- Příkaz, který mění hodnoty nebo význam určitých parametrů nebo příkazů
- Neslouží k sázení textu
- Platnost deklaráce začíná v okamžiku jejího uvedení a končí, jakmile se vyskytne jiná deklaráce stejného typu
- Vyskytuje-li se deklaráce uvnitř prostředí či skupiny, její platnost se rozpíná pouze po odpovídající příkaz `\end`, resp. uzavírající složenou závorku `}`
- Zvláštním případem jsou globální deklaráce, které nejsou omezeny na aktuální prostředí
 - např.: `\pagenumbering{arabic}`

Zadání:

Vytvořte dokument třídy article s velikostí písma 12pt a následujícím obsahem: „Podle dnešního kursu je britská libra \$1,63, což je nárůst o 2% oproti včerejšku.“

Zadání:

Vytvořte dokument třídy article s velikostí písma 12pt a následujícím obsahem: „Podle dnešního kursu je britská libra \$1,63, což je nárůst o 2% oproti včerejšku.“

Vzorové řešení

```
\documentclass[12pt]{article}
\usepackage[cp1250]{inputenc}
\usepackage[czech]{babel}
\begin{document}
  Podle dnešního kursu je britská libra \$1,63, což je nárůst o 2\% oproti včerejšku.
\end{document}
```

- Na začátku dokumentu velmi často bývá titulní stránka, kterou vytvoříme pomocí prostředí

```
\begin{titlepage}   Text titulní stránky   \end{titlepage}
```

- Na začátku dokumentu velmi často bývá titulní stránka, kterou vytvoříme pomocí prostředí
`\begin{titlepage}` *Text titulní stránky* `\end{titlepage}`
- Po titulní stránce se běžně objevuje obsah. Ten vygenerujeme pomocí jediného příkazu `\tableofcontents`

- Na začátku dokumentu velmi často bývá titulní stránka, kterou vytvoříme pomocí prostředí `\begin{titlepage}` *Text titulní stránky* `\end{titlepage}`
- Po titulní stránce se běžně objevuje obsah. Ten vygenerujeme pomocí jediného příkazu `\tableofcontents`
- Samotný text dokumentu se obvykle člení do kapitol, sekcí, podsekcí atd. K tomu slouží postupně tyto příkazy `\chapter`, `\section`, `\subsection`, `\paragraph` ...

- Na začátku dokumentu velmi často bývá titulní stránka, kterou vytvoříme pomocí prostředí `\begin{titlepage}` *Text titulní stránky* `\end{titlepage}`
- Po titulní stránce se běžně objevuje obsah. Ten vygenerujeme pomocí jediného příkazu `\tableofcontents`
- Samotný text dokumentu se obvykle člení do kapitol, sekcí, podsekcí atd. K tomu slouží postupně tyto příkazy `\chapter`, `\section`, `\subsection`, `\paragraph` ...
- Na konci dokumentu může být dodatek, který se uvede pomocí deklarace `\appendix`

- Základním fontem pro hlavní tělo textu v \LaTeX u je antikva střední šířky o velikosti specifikované úvodním příkazem celého dokumentu (`\documentclass[11pt]{...}`)

- Základním fontem pro hlavní tělo textu v \LaTeX u je antikva střední šířky o velikosti specifikované úvodním příkazem celého dokumentu (`\documentclass[11pt]{...}`)
- Pro změnu velikosti písma poskytuje \LaTeX 10 přepínačů (pro příklad ty nejpoužívanější: `\tiny`, `\small`, `\large`, `\LARGE`, `\huge`)

- Základním fontem pro hlavní tělo textu v \LaTeX u je antikva střední šířky o velikosti specifikované úvodním příkazem celého dokumentu (`\documentclass[11pt]{...}`)
- Pro změnu velikosti písma poskytuje \LaTeX 10 přepínačů (pro příklad ty nejpoužívanější: `\tiny`, `\small`, `\large`, `\LARGE`, `\huge`)
- Atributy písma

- Základním fontem pro hlavní tělo textu v \LaTeX u je antikva střední šířky o velikosti specifikované úvodním příkazem celého dokumentu (`\documentclass[11pt]{...}`)
- Pro změnu velikosti písma poskytuje \LaTeX 10 přepínačů (pro příklad ty nejpoužívanější: `\tiny`, `\small`, `\large`, `\LARGE`, `\huge`)
- Atributy písma
 - `\textrm{text}` latinka

- Základním fontem pro hlavní tělo textu v \LaTeX u je antikva střední šířky o velikosti specifikované úvodním příkazem celého dokumentu (`\documentclass[11pt]{...}`)
- Pro změnu velikosti písma poskytuje \LaTeX 10 přepínačů (pro příklad ty nejpoužívanější: `\tiny`, `\small`, `\large`, `\LARGE`, `\huge`)
- Atributy písma
 - `\textrm{text}` latinka
 - `\textbf{text}` **tučné**

- Základním fontem pro hlavní tělo textu v \LaTeX u je antikva střední šířky o velikosti specifikované úvodním příkazem celého dokumentu (`\documentclass[11pt]{...}`)
- Pro změnu velikosti písma poskytuje \LaTeX 10 přepínačů (pro příklad ty nejpoužívanější: `\tiny`, `\small`, `\large`, `\LARGE`, `\huge`)
- Atributy písma
 - `\textrm{text}` latinka
 - `\textbf{text}` **tučné**
 - `\texttt{text}` strojopisné

- Základním fontem pro hlavní tělo textu v \LaTeX u je antikva střední šířky o velikosti specifikované úvodním příkazem celého dokumentu ($\text{\documentclass[11pt]\{...\}}$)
- Pro změnu velikosti písma poskytuje \LaTeX 10 prepínačů (pro příklad ty nejpoužívanější: \tiny , \small , \large , \LARGE , \huge)
- Atributy písma
 - $\text{\textrm\{text\}}$ latinka
 - $\text{\textbf\{text\}}$ **tučné**
 - $\text{\texttt\{text\}}$ strojopisné
 - $\text{\textit\{text\}}$ *kurziva*

- Základním fontem pro hlavní tělo textu v \LaTeX u je antikva střední šířky o velikosti specifikované úvodním příkazem celého dokumentu (`\documentclass[11pt]{...}`)
- Pro změnu velikosti písma poskytuje \LaTeX 10 prepínačů (pro příklad ty nejpoužívanější: `\tiny`, `\small`, `\large`, `\LARGE`, `\huge`)
- Atributy písma
 - `\textrm{text}` latinka
 - `\textbf{text}` **tučné**
 - `\texttt{text}` strojopisné
 - `\textit{text}` *kurziva*
 - `\textsc{text}` KAPITÁLKY

- Základním fontem pro hlavní tělo textu v \LaTeX u je antikva střední šířky o velikosti specifikované úvodním příkazem celého dokumentu (`\documentclass[11pt]{...}`)
- Pro změnu velikosti písma poskytuje \LaTeX 10 přepínačů (pro příklad ty nejpoužívanější: `\tiny`, `\small`, `\large`, `\LARGE`, `\huge`)
- Atributy písma
 - `\textrm{text}` latinka
 - `\textbf{text}` **tučné**
 - `\texttt{text}` strojopisné
 - `\textit{text}` *kurziva*
 - `\textsc{text}` KAPITÁLKY
 - `\textsf{text}` bezpatkové písmo

Matematický text se vkládá do normálního textu dvěma možnými způsoby:

Matematický text se vkládá do normálního textu dvěma možnými způsoby:

- Začleněním do řádků (tvoří část obyčejného textu)

Matematický text se vkládá do normálního textu dvěma možnými způsoby:

- Začleněním do řádků (tvoří část obyčejného textu)
 - Mezi dva znaky \$ (stabilní a velmi často používané)

Matematický text se vkládá do normálního textu dvěma možnými způsoby:

- Začleněním do řádků (tvoří část obyčejného textu)
 - Mezi dva znaky \$ (stabilní a velmi často používané)
 - Ve tvaru `\(vzorec \)` (nestabilní)

Matematický text se vkládá do normálního textu dvěma možnými způsoby:

- Začleněním do řádků (tvoří část obyčejného textu)
 - Mezi dva znaky \$ (stabilní a velmi často používané)
 - Ve tvaru `\(vzorec \)` (nestabilní)
 - Prostředím `\begin{math} vzorec \end{math}`

Matematický text se vkládá do normálního textu dvěma možnými způsoby:

- Začleněním do řádků (tvoří část obyčejného textu)
 - Mezi dva znaky $$$ (stabilní a velmi často používané)
 - Ve tvaru `\(vzorec \)` (nestabilní)
 - Prostředím `\begin{math} vzorec \end{math}`
- Umístěním na samostatné řádky vždy doprostřed

Matematický text se vkládá do normálního textu dvěma možnými způsoby:

- Začleněním do řádků (tvoří část obyčejného textu)
 - Mezi dva znaky $$$ (stabilní a velmi často používané)
 - Ve tvaru `\(vzorec \)` (nestabilní)
 - Prostředím `\begin{math} vzorec \end{math}`
- Umístěním na samostatné řádky vždy doprostřed
 - Mezi „dvojitě dolary“ (sazba matematiky je drahá :-)

Matematický text se vkládá do normálního textu dvěma možnými způsoby:

- Začleněním do řádků (tvoří část obyčejného textu)
 - Mezi dva znaky $$$ (stabilní a velmi často používané)
 - Ve tvaru `\(vzorec \)` (nestabilní)
 - Prostředím `\begin{math} vzorec \end{math}`
- Umístěním na samostatné řádky vždy doprostřed
 - Mezi „dvojitě dolary“ (sazba matematiky je drahá :-)
 - Ve tvaru `\[vzorec \]`

Matematický text se vkládá do normálního textu dvěma možnými způsoby:

- Začleněním do řádků (tvoří část obyčejného textu)
 - Mezi dva znaky `$` (stabilní a velmi často používané)
 - Ve tvaru `\(vzorec \)` (nestabilní)
 - Prostředím `\begin{math} vzorec \end{math}`
- Umístěním na samostatné řádky vždy doprostřed
 - Mezi „dvojitě dolary“ (sazba matematiky je drahá :-)
 - Ve tvaru `\[vzorec \]`
 - Prostředím `\begin{displaymath} vzorec \end{displaymath}`

Matematický text se vkládá do normálního textu dvěma možnými způsoby:

- Začleněním do řádků (tvoří část obyčejného textu)
 - Mezi dva znaky `$` (stabilní a velmi často používané)
 - Ve tvaru `\(vzorec \)` (nestabilní)
 - Prostředím `\begin{math} vzorec \end{math}`
- Umístěním na samostatné řádky vždy doprostřed
 - Mezi „dvojitě dolary“ (sazba matematiky je drahá :-)
 - Ve tvaru `\[vzorec \]`
 - Prostředím `\begin{displaymath} vzorec \end{displaymath}`
 - Pomocí dalších speciálních prostředí

- Horní a dolní indexy jsou sázeny pomocí symbolů \wedge a $_$

- Horní a dolní indexy jsou sázeny pomocí symbolů $\hat{\ }$ a $_$
- Pro zlomky je využíván příkaz $\frac{\text{čitateľ}}{\text{jmenovatel}}$

- Horní a dolní indexy jsou sázeny pomocí symbolů $\hat{\ }$ a $_$
- Pro zlomky je využíván příkaz $\frac{\text{čítatel}}{\text{jmenovatel}}$
- Odmocniny se sází příkazem $\sqrt[n]{arg}$

- Horní a dolní indexy jsou sázeny pomocí symbolů $\hat{\ }$ a $_$
- Pro zlomky je využíván příkaz $\frac{\text{čítatel}}{\text{jmenovatel}}$
- Odmocniny se sází příkazem $\sqrt[n]{arg}$
- Suma s horní a dolní mezí vypadá takto: $\sum_{i=1}^n$

Zadání:

Vytvořte následující výstup:

$$\frac{\sqrt{ax + b^3}}{x}$$

Zadání:

Vytvořte následující výstup:

$$\frac{\sqrt{ax + b^3}}{x}$$

Vzorové řešení

```
\documentclass[12pt]{article}
\usepackage[cp1250]{inputenc}
\usepackage[czech]{babel}
\begin{document}
  $$ \frac{\sqrt{ax+b^3}}{x} $$
\end{document}
```

Zadání:

Vytvořte následující výstup:

$$s = \sqrt{\frac{1}{n} \sum_{i=1}^r p_i (x_i - x)^2}$$

Zadání:

Vytvořte následující výstup:

$$s = \sqrt{\frac{1}{n} \sum_{i=1}^r p_i (x_i - x)^2}$$

Vzorové řešení

```
\documentclass[12pt]{article}
\usepackage[cp1250]{inputenc}
\usepackage[czech]{babel}
\begin{document}
  $$ \sqrt{\frac{1}{n} \sum_{i=1}^r p_i (x_i - x)^2} $$
\end{document}
```

Jako základní prostředí pro vytváření tabulek se v \LaTeX u používá tabular. Existují však i další prostředí, převážně určené pro sazbu rozsáhlých tabulek.

Jako základní prostředí pro vytváření tabulek se v \LaTeX u používá tabular. Existují však i další prostředí, převážně určené pro sazbu rozsáhlých tabulek.

Pro to, abychom tabulku vůbec mohli dostat na výstup, použijeme prostředí tabular následovně:

Jako základní prostředí pro vytváření tabulek se v \LaTeX u používá `tabular`. Existují však i další prostředí, převážně určené pro sazbu rozsáhlých tabulek.

Pro to, abychom tabulku vůbec mohli dostat na výstup, použijeme prostředí `tabular` následovně:

Ukázka

```
\begin{tabular}[vert. umístění]{sloupce}  
...  
\end{tabular}
```

Jako základní prostředí pro vytváření tabulek se v \LaTeX používá `tabular`. Existují však i další prostředí, převážně určené pro sazbu rozsáhlých tabulek.

Pro to, abychom tabulku vůbec mohli dostat na výstup, použijeme prostředí `tabular` následovně:

Ukázka

```
\begin{tabular}[vert. umístění]{sloupce}  
...  
\end{tabular}
```

Nepovinný parametr *vert. umístění* určuje, jak má být zarovnán obsah buněk tabulky ve vertikálním směru. Implicitně je nastaveno centrování, pro změnu lze použít tyto dvě volby:

Jako základní prostředí pro vytváření tabulek se v \LaTeX používá `tabular`. Existují však i další prostředí, převážně určené pro sazbu rozsáhlých tabulek.

Pro to, abychom tabulku vůbec mohli dostat na výstup, použijeme prostředí `tabular` následovně:

Ukázka

```
\begin{tabular}[vert. umístění]{sloupce}  
...  
\end{tabular}
```

Nepovinný parametr *vert. umístění* určuje, jak má být zarovnán obsah buněk tabulky ve vertikálním směru. Implicitně je nastaveno centrování, pro změnu lze použít tyto dvě volby:

- `t` zarovnání k horní části buňky (top)

Jako základní prostředí pro vytváření tabulek se v \LaTeX používá `tabular`. Existují však i další prostředí, převážně určené pro sazbu rozsáhlých tabulek.

Pro to, abychom tabulku vůbec mohli dostat na výstup, použijeme prostředí `tabular` následovně:

Ukázka

```
\begin{tabular}[vert. umístění]{sloupce}  
...  
\end{tabular}
```

Nepovinný parametr *vert. umístění* určuje, jak má být zarovnán obsah buněk tabulky ve vertikálním směru. Implicitně je nastaveno centrování, pro změnu lze použít tyto dvě volby:

- `t` zarovnání k horní části buňky (top)
- `b` zarovnání dospod buňky (bottom)

Tabulky (2)

Místo položky sloupce vložíme řetězec vytvořený z níže uvedených znaků. Tento řetězec oznamuje \LaTeX u, jak má zpracovat jednotlivé sloupce.

Místo položky sloupce vložíme řetězec vytvořený z níže uvedených znaků. Tento řetězec oznamuje \LaTeX u, jak má zpracovat jednotlivé sloupce.

- | Sloupec, jehož buňky budou zarovnány k levé straně

Místo položky sloupce vložíme řetězec vytvořený z níže uvedených znaků. Tento řetězec oznamuje \LaTeX u, jak má zpracovat jednotlivé sloupce.

- l Sloupec, jehož buňky budou zarovnány k levé straně
- r Sloupec s buňkami zarovnanými na pravou stranu

Místo položky sloupce vložíme řetězec vytvořený z níže uvedených znaků. Tento řetězec oznamuje \LaTeX u, jak má zpracovat jednotlivé sloupce.

- l Sloupec, jehož buňky budou zarovnány k levé straně
- r Sloupec s buňkami zarovnanými na pravou stranu
- c Sloupec s buňkami zarovnanými na střed

Místo položky sloupce vložíme řetězec vytvořený z níže uvedených znaků. Tento řetězec oznamuje \LaTeX u, jak má zpracovat jednotlivé sloupce.

- l Sloupec, jehož buňky budou zarovnány k levé straně
- r Sloupec s buňkami zarovnanými na pravou stranu
- c Sloupec s buňkami zarovnanými na střed
- p{šířka} Sloupec s buňkami, které mají danou *šířku* a text se do nich bude lámat (ve výše uvedených typech sloupců zůstává text na jediné řádce)

Místo položky sloupce vložíme řetězec vytvořený z níže uvedených znaků. Tento řetězec oznamuje L^AT_EXu, jak má zpracovat jednotlivé sloupce.

- l Sloupec, jehož buňky budou zarovnány k levé straně
- r Sloupec s buňkami zarovnanými na pravou stranu
- c Sloupec s buňkami zarovnanými na střed
- p{šířka} Sloupec s buňkami, které mají danou šířku a text se do nich bude lámat (ve výše uvedených typech sloupců zůstává text na jediné řádce)
- | Neurčuje sloupec, ale slouží k oddělení jednotlivých sloupců tabulky jednou svislou čárou (Pokud uvedeme dvě vedle sebe, bude tam čára dvojitá atd.)

Tabulky (3)

Do prostoru prostředí (tam, co byly uvedeny tři tečky) se vpisují data. Přejít na nový řádek tabulky zajistíme příkazem `\\` (dvě opačná lomítka). Pokud ovšem chceme, aby v tabulce byly oddělovány řádky linkou, musíme za dvojitá lomítka připsat příkaz `\hline`. Jednotlivé sloupce se oddělují znakem `&`.

Tabulky (3)

Do prostoru prostředí (tam, co byly uvedeny tři tečky) se vpisují data. Přejít na nový řádek tabulky zajistíme příkazem `\\` (dvě opačná lomítka). Pokud ovšem chceme, aby v tabulce byly oddělovány řádky linkou, musíme za dvojitá lomítka připsat příkaz `\hline`. Jednotlivé sloupce se oddělují znakem `&`.

Tabulky (3)

Do prostoru prostředí (tam, co byly uvedeny tři tečky) se vpisují data. Přejít na nový řádek tabulky zajistíme příkazem `\\` (dvě opačná lomítka). Pokud ovšem chceme, aby v tabulce byly oddělovány řádky linkou, musíme za dvojité lomítka připsat příkaz `\hline`. Jednotlivé sloupce se oddělují znakem `&`.

Příklad zdrojového kódu tabulky

```
\begin{tabular}{|l|c|c|} \hline
  Jméno a příjmení & Ročník & Body \\ \hline
  Jan Novák & 2. & 15.213 \\ \hline
  Tomáš Černý & 4. & 14.996 \\ \hline
\end{tabular}
```

Tabulky (3)

Do prostoru prostředí (tam, co byly uvedeny tři tečky) se vpisují data. Přejít na nový řádek tabulky zajistíme příkazem `\\` (dvě opačná lomítka). Pokud ovšem chceme, aby v tabulce byly oddělovány řádky linkou, musíme za dvojitá lomítka připsat příkaz `\hline`. Jednotlivé sloupce se oddělují znakem `&`.

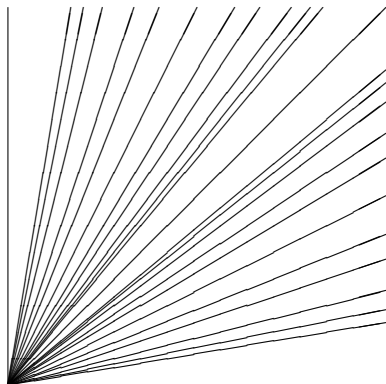
Příklad zdrojového kódu tabulky

```
\begin{tabular}{|l|c|c|} \hline
  Jméno a příjmení & Ročník & Body \\ \hline
  Jan Novák & 2. & 15.213 \\ \hline
  Tomáš Černý & 4. & 14.996 \\ \hline
\end{tabular}
```

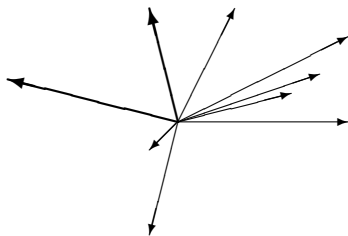
Výsledkem je následující tabulka:

Jméno a příjmení	Ročník	Body
Jan Novák	2.	15.213
Tomáš Černý	4.	14.996

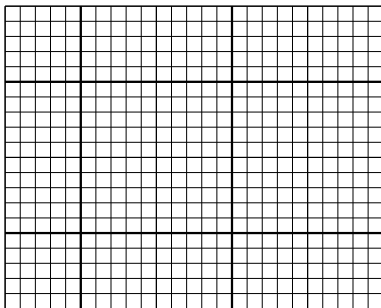
Opusťme nyní nezáživnou teorii :-)) a podívejme se, co vše je v \LaTeX u možné stvořit.



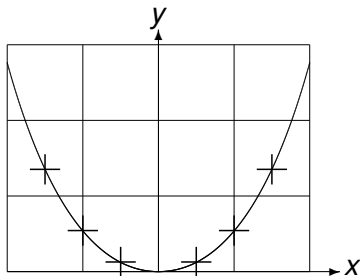
Vektory



Milimetrový „papír“



Body tvořící parabolu v souřadnicovém systému



I tato prezentace je vysázena v \TeX u :-)