

# Zpravodaj Československého sdružení uživatelů TeXu

---

Peter Wilson; Jan Šustek

Mělo by to fungovat. IV - Tři otázky uživatelů

*Zpravodaj Československého sdružení uživatelů TeXu*, Vol. 23 (2013), No. 2, 100–106

Persistent URL: <http://dml.cz/dmlcz/150219>

## Terms of use:

© Československé sdružení uživatelů TeXu, 2013

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ*:  
*The Czech Digital Mathematics Library* <http://dml.cz>

Příspěvek ukazuje krátká makra, která řeší některé problémy začínajících uživatelů L<sup>A</sup>T<sub>E</sub>Xu. Konkrétně se jedná o makra pro testování prázdného argumentu, variantu L<sup>A</sup>T<sub>E</sub>Xového makra `\cleardoublepage` a o makra umožňující odlišně zpracovávat první znak a zbývající znaky daného řetězce.

## Klíčová slova

L<sup>A</sup>T<sub>E</sub>X, prázdný argument, `\cleardoublepage`

Seagulls scream upon the shorelines' wrack  
And seals abound  
Amid the setting sun's glistering track  
Across the Sound.

*Puget Sound*

Cílem tohoto seriálu je ukázat čtenáři krátké kousky kódu, které mohou vyřešit některé z jeho problémů. Doufám, že situaci ještě více nezkomplikuji v důsledku svých chyb.

Uvítám opravy, poznámky a návrhy na změny. Když mluvím o poznámkách, David Elliott byl první,<sup>1</sup> kdo si všiml, že jsem v posledním dílu [6] nesprávně přiřadil Tennysonovi báseň *Dover beach* od Matthewa Arnolda. Nemám ponětí, jak se to mohlo stát.

Obsahem dnešního dílu jsou tři problémy navržené čtenáři.

They cannot scare me with their empty spaces  
Between stars — on stars where no human race is.

*Desert Places*

ROBERT FROST

## 1. Prázdné argumenty

V jednom z předchozích dílů [5] jsem psal o tom, jak zjistit, zda jsou dva řetězce stejné. Nedávno v diskusní skupině `texhax` [2] zazněla otázka, jak testovat, zda je argument prázdný. To není nic jiného než porovnávání argumentu s prázdným řetězcem. Bohužel ale dřívější postup v tomto případě nefunguje.

---

Z anglického originálu *Glisterings* [7] přeložil Jan Šustek.

<sup>1</sup>Moje manželka byla těsně druhá.

V L<sup>A</sup>T<sub>E</sub>Xu často potřebujeme testovat, zda makro bylo, nebo nebylo voláno s nepovinným argumentem. Typický tvar definice makra je

```
1 \newcommand{\makro}[1][\@empty]{...
2   \ifx\@empty#1 ... % bez argumentu
3   \else ...          % s argumentem
4   \fi}
```

Otázkou pro tento díl je, jak přesně napsat řádek 6 v následujícím kódu.

```
5 \newcommand{\makro}[1]{...
6   \if(#1 je prázdný) ... % prázdný argument
7   \else ...             % neprázdný argument
8   \fi}
```

kde pod pojmem prázdný máme na mysli nula nebo více mezer. Takže jak {}, tak { } znamenají prázdný argument. Pokud používáte L<sup>A</sup>T<sub>E</sub>X, řešení poskytuje balíček ifmtarg dostupný na CTAN. Pro uživatele T<sub>E</sub>Xu zde uvádím ekvivalentní kód. Všimněte si, že všechny definice před řádkem 16 jsou součástí L<sup>A</sup>T<sub>E</sub>Xu.

```
9 \def\makeatletter{\catcode'\@11\relax}
10 \def\makeatother{\catcode'\@12\relax}
11 \makeatletter
12 \long\def\@gobble #1{}
13 \long\def\@firstofone#1{#1}
14 \long\def\@firstoftwo#1#2{#1}
15 \long\def\@secondoftwo#1#2{#2}
16 \begingroup
17 \catcode'\Q=3
18 \long\gdef\@ifmtarg#1{%
19   \@xifmtarg#1QQ\@secondoftwo\@firstoftwo\@nil}
20 \long\gdef\@xifmtarg#1#2Q#3#4#5\@nil{#4}
21 \long\gdef\@ifnotmtarg#1{%
22   \@xifmtarg#1QQ\@firstofone\@gobble\@nil}
23 \endgroup
24 \makeatother
```

Důležité části předchozího úseku kódu jsou

```
25 \@ifmtarg{<arg>}{<prázdný>}{<neprázdný>}
26 \@ifnotmtarg{<arg>}{<neprázdný>}
```

Tato makra můžeme použít následovně:

```
27 \def\jeprazdny#1{\@ifmtarg{#1}{prázdný}{neprázdný}}
28 \def\jeneprazdny#1{\@ifnotmtarg{#1}{neprázdný}}
```

29	<code>\def\nic{}</code>	
30	<code>\jeprazdny{}</code>	prázdný
31	<code>\jeprazdny{ }</code>	prázdný
32	<code>\jeprazdny{\nic}</code>	neprázdný
33	<code>\jeprazdny{ A }</code>	neprázdný
34	<code>\jeneprazdny{}</code>	
35	<code>\jeneprazdny{ }</code>	
36	<code>\jeneprazdny{\nic}</code>	neprázdný

Balíček `ifmtarg` byl původně mnohem jednodušší, ale Donald Arseneau v něm našel chyby. Problémy byly popsány v seriálu Michaela Downese *Around the Bend* a jsou dostupné na CTAN na `info/aro-bend/answer.002`

Faultily faultless, icily regular, splendidly null.  
Dead perfection, no more.

*Maud*  
ALFRED, LORD TENNYSON

## 2. Užitečnost ničeho

Další tazatelka v diskusní skupině `texhax` [1] potřebovala naprogramovat verzi  $\text{\LaTeX}$ ového makra `\cleardoublepage`, kde by další text začínal na sudé stránce. Na první pohled se zdá, že postačí použití  $\text{\LaTeX}$ ového makra `\cleardoublepage` (kterým se dostaneme na lichou stránku) a poté se makrem `\clearpage` nebo `\newpage` dostaneme na následující stranu. Tento postup však nefunguje. Nelze totiž ukončit stranu, na níž kromě záhlaví a zápatí nic není. V této situaci pomůže, když na stránku vložíme prázdný box. V  $\text{\TeX}$ u existuje makro `\null`, které se expanduje na prázdný horizontální box. Zde ukážu makro `\prazdnybox`, které je možné použít i v jiných případech.

```

37 \newcommand*{\prazdnybox}{\leavevmode\hbox{}}
38 \newcommand{\cleartoevenpage}[1][\@empty]{%
39   \clearpage
40   \ifodd\c@page
41     \prazdnybox\ifx\@empty#1\else#1\fi
42   \newpage
43   \fi}

```

Makro `\cleartoevenpage` ukončí současnou stranu, a pokud následující strana není lichá, je hotovo. Pokud je lichá, vloží na ni neviditelný `\prazdnybox` a přejde na následující stranu, která je sudá. Nepovinný argument slouží ke vložení textu nebo ilustrace na přeskočenou lichou stranu. Například

```

44 \cleartoevenpage[%
45   \vfill\centering PRÁZDNÁ STRANA\vfill
46   \thispagestyle{empty}]

```

vycentruje na prázdnou stranu text „PRÁZDNÁ STRANA“, přičemž na straně nebude ani záhlaví, ani zápatí.

Pokud jste někdy zkoušeli něco podobného jako

```

47 \begin{description}
48   \item[Nekonečný cyklus]\\
49   Viz Cyklus, nekonečný
50 ...

```

pravděpodobně jste obdrželi chybové hlášení

There's no line to end here.

Situaci je možné vyřešit použitím makra `\prazdnybox` těsně před makrem `\\`.

We may be in some degree whatever character we choose.

*London Journal*  
JAMES BOSWELL

### 3. Oddělení prvního znaku

Čtenář v diskusní skupině `texhax` [4] potřeboval makro, které vezme řetězec a vrátí řetězec, jehož první písmeno bude velké. Našlo se několik řešení. Tady ukážu několik svých. Všechna řešení jsou založena na tom, že neseparovaný argument makra je buď text ve složených závorkách, nebo jeden token, což může být buď znak, nebo název makra. Při definování makra v `TeXu` je seznam argumentů ukončen levou složenou závorkou, která uvozuje samotný text definice.

Tady je první možné řešení.

```

52 \def\vezmidva#1#2\konec{%
53   \gdef\prvniznak{#1}\gdef\zbytek{#2}}
54 \def\oddel#1{\vezmidva#1\konec}
55 \def\PrvniVelke#1{\oddel{#1}%
56   \MakeUppercase{\prvniznak}\zbytek}

```

Makro `\vezmidva` očekává dva argumenty, kde druhý je ukončen tokenem `\konec`. (Předpokládá se, že token `\konec` nebude součástí textu; je samozřejmě možné zvolit i jiný název.) Makro `\oddel` vezme jeden argument (řetězec) a pošle ho makru `\vezmidva`. To vezme první znak řetězce a uloží ho do makra `\prvniznak`.

Zbytek řetězce uloží do makra `\zbytek`. Makro `\PrvniVelke` vezme argument, zavolá makro `\oddel` (a tedy i makro `\vezmidva`) a zajistí, že z `\prvniznak` bude velké písmeno. Zbytek znaků pouze opíše.

Tento postup nebude fungovat, pokud argument makra `\PrvniVelke` bude makro, které se expanduje na řetězec (například `\def\arg{slovo}`). Tuto situaci je možné vyřešit použitím primitivu `\expandafter`, jímž zajistíme, že se argument makru `\oddel` pošle po jedné expanzi.

```
57 \def\PrvniVelke#1{%
58   \expandafter\oddel\expandafter{#1}%
59   \MakeUppercase{\prvniznak}\zbytek}
```

Další řešení není tak univerzální jako předchozí, protože řetězec převede rovnou a neukládá jeho části do maker `\prvniznak` a `\zbytek`.

```
60 \def\rozdel#1#2\konec{%
61   \MakeUppercase{#1}\MakeLowercase{#2}}
62 \def\PrvniVelke#1{\expandafter\rozdel#1\konec}
```

Základní myšlenka je stejná jako u prvního řešení. Další funkcí uvedeného makra je zajistit, aby pouze první znak byl velké písmeno, zatímco ostatní budou malé.

Žádné z uvedených řešení nebude fungovat, pokud první znak bude akcentovaný znak (například `\v{S}`) nebo znak definovaný makrem (například `\oe`). Uwe Lück [3] našel obecnější, ale složitější řešení.

```
63 \DeclareRobustCommand{\PrvniVelke}[1]{%
64   \protected@edef\PrvniVelke@rg{#1}%
65   \expandafter\navelke\PrvniVelke@arg\konec}
```

Příkaz `\DeclareRobustCommand` namísto `\def` nebo `\newcommand` umožní použít makro `\PrvniVelke` uvnitř pohyblivých argumentů, které navíc nemusí být chráněno makrem `\protect`. Makro `\protected@edef` je použito, aby expandovalo argument a přitom zachovalo všechny výskyty `\protect`. Aby bylo možné zpracovat případný akcentovaný první znak, je třeba řetězec rozdělit na tři části: první prvek (buď znak, nebo makro s akcentem), druhý prvek (druhý znak, nebo argument makra) a zbytek řetězce. Makro ale nebude fungovat, pokud řetězec bude mít méně než dva znaky.

```
66 \def\navelke#1#2#3\konec{%
67   \let\token@I#1%
68   \let\token@IIa@empty
69   \def\token@IIb{#2}%
70   \expandafter\testuj@akcent\seznam@akcentu\@zarazka
71   \MakeUppercase{#1\token@IIa}%
72   \MakeLowercase{\token@IIb#3}}
```

Makro `\navelke` vezme tři argumenty, které tvoří uvedené tři části vstupního řetězce. První dva argumenty se uloží do maker `\token@I` a `\token@IIb`. Makro `\testuj@akcent` zjistí, zda je první token nějaký akcent; v tom případě změní makra `\token@IIa` a `\token@IIb`.

```

73 \def\seznam@akcentu{'>"\'\'\'b\c}% a další
74 \def\testuj@akcent#1{%
75   \ifx#1@zarazka\else
76     \ifx\token@I#1
77       \let\token@IIa\token@IIb
78       \let\token@IIb\empty
79     \fi
80     \expandafter\testuj@akcent
81   \fi}

```

V makru `\seznam@akcentu` je seznam maker s akcenty. Seznam by měl také obsahovat neabecední znaky, jimiž by mohl začínat řetězec, například otevírací uvozovku.

Makro `\testuj@akcent` prochází uvedený seznam akcentů. Pokud se některý shoduje s makrem `\token@I`, změní se makra `\token@IIa` a `\token@IIb`. Pokud řetězec začíná akcentem, je v makru `\token@IIa` uložen znak s akcentem a `\token@IIb` je prázdné. Pokud řetězec nezačíná akcentem, je `\token@IIa` prázdný a v makru `\token@IIb` je uložen druhý znak řetězce.

Následuje několik příkladů použití takto definovaného makra `\PrvniVelke`.

```

82 mal\'e VELK\'E \& \PrvniVelke{mal\'e VELK\'E}
83                                     malé VELKÉ & Malé velké
84 \def\neco{n\v EjAk\'Y tExT}
85 \neco \& \PrvniVelke{\neco}          něJkÁY tExT & Nějaký text
86 {oe}{rstead} \& \PrvniVelke{{oe}{rstead}}      ørstead & Ørstead
87 {c}{cedilla} \& \PrvniVelke{{c}{cedilla}}        çedilla & Çedilla
88 \emph{kurz\'i va} \& \emph{\PrvniVelke{kurz\'i va}}
89                                     kurzíva & Kurzíva
90 'uvozovky' \& \PrvniVelke{'uvozovky'}          'uvozovky' & 'Uvozovky'
91 >que? \& \PrvniVelke{>que?}                    ¿que? & ¿Que?

```

Jako vždy, když používáme makra, jejichž název obsahuje znak `@`, musíme být uvnitř balíčku nebo třídy, anebo musíme kód uzavřít mezi `\makeatletter` a `\makeatother`.

Je možné, že někdy příště ukážu makra, která procházejí řetězec znak po znaku, případně užívají jiný druh cyklu. Ale je možné, že neukážu.

## Seznam literatury

- [1] Dittmar, Susan. Variant of `\cleardoublepage` starting on even page numbers. Příspěvek na `texhax`, 18. 8. 2005.
- [2] Fenn, Adam. Empty arguments. Příspěvek na `texhax`, 17. 8. 2005.
- [3] Lück, Uwe. Re: [texhax] read and process single characters. Příspěvek na `texhax`, 24. 6. 2005.
- [4] Wagner, Torsten. Read and process single characters. Příspěvek na `texhax`, 24. 6. 2005.
- [5] Wilson, Peter. Glisterings. *TUGboat*, 22(4):339–341, December 2001.
- [6] Wilson, Peter. Glisterings. *TUGboat*, 25(2):201–202, 2004.
- [7] Wilson, Peter. Glisterings. *TUGboat*, 26(3):253–255, 2005.

## Summary: It Might Work. IV – Three Users’ Questions

This paper describes short macros that solve several problems of  $\text{\LaTeX}$  beginners. There are macros testing whether an argument is empty, a variant of  $\text{\LaTeX}$  macro `\cleardoublepage` and macros that process differently the first character and the remaining characters of a given string.

### Key words:

$\text{\LaTeX}$ , empty argument, `\cleardoublepage`

*Peter Wilson, herries.press@earthlink.net  
18912 8th Ave. SW  
Normandy Park, WA 98166 USA*