

Zpravodaj Československého sdružení uživatelů TeXu

Michal Hoftich

Responzivní design a automatická sazba s LuaLaTeXem

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 34 (2024), No. 1-4, 28–38

Persistent URL: <http://dml.cz/dmlcz/152648>

Terms of use:

© Československé sdružení uživatelů TeXu, 2024

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>

Responzivní design a automatická sazba s Lua \LaTeX em

MICHAL HOFTICH

Tento článek představuje využití metod responzivního designu a pokročilých vlastností Lua \LaTeX u pro automatickou sazbu dokumentů určených pro různé cílové výstupy, jak tištěné, tak elektronické, například mobilní telefony, tablety nebo čtečky e-knih.

Konkrétně se zaměřuje na využití Lua \LaTeX u pro automatizovanou sazbu s pomocí balíčků *Responsive* [1] pro nastavení velikosti písma a řádkování podle velikosti stránky, *Luavlna* [2] pro zamezení výskytu jednopísmenných předložek na koncích řádků, *Lua-widow-control* [3] pro omezení osamocených řádků na koncích a začátcích stránek a *Linebreaker* [4], který brání přetečení řádků.

Klíčová slova: automatická sazba, responzivní design, Lua \LaTeX

Úvod

Před časem jsem si pořídil čtečku elektronických knih, ale i přesto většinu textů čtu z obrazovky svého PC, protože pochází z webových zdrojů. Napadlo mě tedy, že bych si delší články ukládal pro pozdější přečtení na čtečce. K tomuto účelu samozřejmě existuje řada aplikací, ale já jsem se rozhodl vytvořit si vlastní, kterou si přizpůsobím přesně svým potřebám a preferencím. Další motivací je možnost naučit se něco nového a vytvořit balíčky, které budou užitečné i pro další uživatele \TeX u.

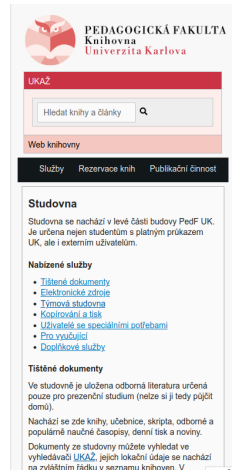
Mým cílem je, aby řešení bylo co nejvíce automatizované, abych například nemusel řešit přetečené řádky nebo jiné chyby, které by vyžadovaly ruční zásah. Díky možnostem Lua \LaTeX u je takové řešení již v současnosti možné, jak si ukážeme v následujícím textu.

Responzivní design

Jedním z problémů, který je třeba řešit, je nastavení správné velikosti písma z hlediska čitelnosti. Výchozí velikost písma v \LaTeX u je 10 bodů bez ohledu na velikost stránky. To je vhodná velikost písma pro stránku formátu A5. Pro formát A4 by měla být velikost písma větší, pro menší displeje čteček a mobilních telefonů může být naopak menší. Stejně tak můžeme změnit řádkování, které také ovlivňuje čitelnost textu v závislosti na velikosti písma a stránky.



(a) Ukázka stránky na velkém monitoru



(b) Ukázka stránky na malém displeji

Obrázek 1: Ukázka zobrazení webové stránky s využitím responzivního designu

Podobný problém řeší webové prohlížeče, které musí zobrazit text jak na velkých monitorech PC, tak na menších displejích notebooků, tabletů a mobilních telefonů. Řešení, které používají, se nazývá *responzivní design*.

Responzivní design je způsob návrhu webových stránek, který umožňuje pružné a dynamické přizpůsobení vzhledu a uspořádání obsahu stránky různým zobrazovacími zařízeními. Jedním z klíčových prvků responzivního designu je flexibilní struktura, která umožňuje přizpůsobit velikost prvků na stránce zobrazovacímu zařízení.

Dalším důležitým prvkem jsou *media queries*. Ty umožňují definovat pravidla, která se aplikují na základě vlastností zobrazovacího zařízení, jako jsou například šířka a výška obrazovky nebo typ výstupu (papír, displej). Díky těmto pravidlům může být stejný kód stránky dobře zobrazen jak na velkém monitoru, tak na mobilních zařízeních nebo při tisku. Ukázkou reálného využití můžete vidět na Obrázku 1.

Balíček *Responsive* [1] se těmito zásadami inspiruje. Jeho hlavní funkcí je nastavení velikosti písma podle velikosti stránky a přibližného počtu znaků, které by se měly vejít na stránku. Dále nastavuje typografickou stupnici (ovlivňuje velikost písma například u nadpisů nebo poznámek pod čarou), písmovou osnovu a podporuje jednoduchou verzi *media queries*.

Nastavení balíčku *Responsive*

Responsive automaticky nastavuje velikost písma, řádkování a typografickou stupnici na začátku dokumentu. Výchozí hodnoty můžeme změnit pomocí voleb balíčku (`\usepackage[<volby>]{responsive}`) nebo příkazem `\ResponsiveSetup<volby>`. Příkaz `\ResponsiveSetup` můžeme využít i přímo v textu dokumentu, například pro lokální změny nastavení písma.

Balíček *Responsive* nabízí následující volby:

noautomatic – zabrání nastavení velikosti písma a řádkování automaticky na začátku dokumentu.

characters – počet znaků při automatickém nastavení velikosti písma.

scale – typografická stupnice použitá pro velikosti řezů písma.

lineratio – poměr využitý při výpočtu řádkování.

Základní velikost písma

Velikost písma můžeme nastavit pomocí příkazu `\setsizes{<počet znaků na řádek>}`. *Responsive* se snaží nastavit velikost písma tak, aby na řádku byl v průměru požadovaný počet znaků. Skutečný počet znaků ovšem záleží na použitém textu, neboť při použití proporcionálních fontů má každé písmeno jinou šířku. Ve skutečnosti může být počet znaků zobrazených na řádce mírně vyšší.

Pokud v příkazu `\setsizes` nevedeme počet znaků, použije se hodnota volby **characters**. Následující příklad využívá právě nastavení této volby. Obrázek 2 ukazuje, jak se stejný text může v daném rámci zobrazit rozdílně v závislosti na nastavených volbách.

```
\begin{minipage}{5cm}
\ResponsiveSetup{characters=55}
\setsizes{}
\lipsum[1]
\end{minipage}}
```

Řádkování

V základním nastavení L^AT_EXu je řádkování nastaveno na velikost písma násobenou hodnotou 1,2. Pro jiná písma a velikosti stránky může být vhodná jiná velikost řádkování. Stejně tak mohou být vhodné rozdílné hodnoty pro tištěnou a elektronickou verzi dokumentu.

Inspiroval jsem se článkem Edoarda Cavazza [5] o čitelnosti a přidal podporu pro nastavení řádkování na základě poměru výšky malých písmen a proměnné **lineratio**. Ten se získá následujícím výpočtem:

$$\text{řádkování} = \frac{1\text{ex}}{\text{lineratio}/100}$$

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

(a) characters=55

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque.

(b) characters=25, lineratio=38

Obrázek 2: Rozdíl velikosti písma v závislosti na počtu znaků na řádku

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

(a) lineratio=38

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

(b) lineratio=34

Obrázek 3: Změna řádkování změnou hodnoty lineratio

Jaký vliv mají změny hodnoty `lineratio`, můžete pozorovat na Obrázku 3. Volba její optimální hodnoty závisí na použitém písmu i velikosti stránky. Pro dosažení maximální čitelnosti výstupu je vhodné porovnat výstup při použití různých hodnot.

Typografická stupnice

Typografická stupnice je soubor předem stanovených velikostí písma, které jsou používány pro vytvoření jednotného vizuálního stylu dokumentu nebo webové stránky. Tyto velikosti jsou obvykle vyjádřeny v bodové jednotce a postupně se zvětšují nebo zmenšují o konkrétní interval, který leží na stupnici.

Typografická stupnice může obsahovat velikosti pro nadpisy, poznámky pod čarou a běžný text. Správné používání typografické stupnice pomáhá vytvořit

vizuální hierarchii, která zlepšuje čitelnost a estetický dojem textu. Více informací o typografických stupnicích naleznete v článku Spencera Mortensena [6].

V \LaTeX u je typografická stupnice dostupná pomocí příkazů jako `\large`, `\Large`, `\LARGE`, `\huge`, nebo třeba `\scriptsize`. Každý z těchto příkazů je od předešlé velikosti vzdálený o jeden interval. Výchozí stupnice v balíčku *Responsive* je nejbližší stupnici použité v \LaTeX u. V Mortensenově článku se nazývá *tetratonická*. Balíček nabízí i další stupnice popsané v článku, například *golden*, založenou na zlatém řezu. Efekt použití stupnice je zobrazen na Obrázku 4.

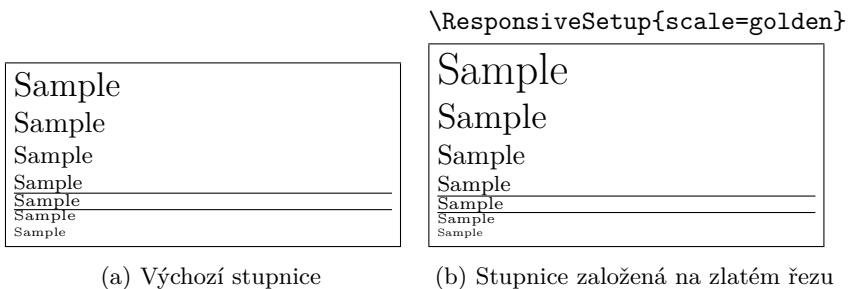
Můžeme však definovat i vlastní stupnici. Například následující kód definuje stupnici, na které se velikost textu zdvojnásobí (`ratio=2`) za tři kroky (`number=3`). Při definici vlastních voleb stupnice je třeba zadat hodnotu `scale=none`. Příkaz `\setsizes` poté redefinuje velikosti písem.

```
\ResponsiveSetup{ratio=2, number=3, scale=none}
\setsizes{}
```

Media queries

Media queries jsou technikou, která umožňuje webovým vývojářům dynamicky přizpůsobovat vzhled a chování webových stránek na základě různých vlastností zařízení, jako jsou například šířka a výška obrazovky, orientace zařízení, podpora barev a mnoho dalších. Díky těmto podmínkám lze vytvářet responzivní a flexibilní webové stránky, které se dokážou automaticky přizpůsobovat různým typům a velikostem zařízení, na kterých jsou zobrazovány.

Jak může být tato technika užitečná pro autory \LaTeX ových balíčků? Mohli by například nastavit velikost písma, řádkování a dalších prvků pro určité rozměry stránky. Poté, co uživatel zvolí velikost stránky podle velikosti zařízení, pro které chce dokument zkompileovat, nastaví se tyto prvky automaticky. Autor balíčku může například definovat, že pokud je šířka textového řádku menší než určitý



Obrázek 4: Ukázka typografických stupnic (výchozí velikost písma je zvýrazněna linkami)

```
\mediaquery{max-textwidth=4cm}
{\ResponsiveSetup{characters=45}}
{\ResponsiveSetup{characters=60}}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque.

(a) Šířka textu 5 cm

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque.

(b) Šířka textu 3,9 cm

Obrázek 5: Příklad Media Query, která zobrazí méně znaků na řádku, pokud je šířka textu menší než 4 cm

rozměr, zobrazí se na něm méně znaků než na řádcích delších. Výsledek je zobrazen na Obrázku 5.

Media query můžeme deklarovat pomocí příkazu `\mediaquery`, který očekává tři parametry – první je seznam testů, další parametr poté očekává kód, který se vykoná, pokud se testy vyhodnotí jako pravdivé, poslední pak kód spuštěný při nesplnění podmínce. Kód může obsahovat příkaz `\ResponsiveSetup`, ale i libovolné jiné příkazy. Například nastavení velikosti textového bloku, záhlaví a zápatí pomocí balíčku *Geometry*.

Můžeme testovat následující vlastnosti stránky: `paperwidth` a `paperheight` pro rozměry stránky, `textwidth` a `textheight` pro rozměry textu, `orientation` pro orientaci textu a `twocolumn` pro detekci použití dvou sloupců textu v dokumentu.

Testy pro rozměry textu a stránky podporují také předpony `max-` a `min-`. Pomocí nich můžeme testovat, zda daný rozměr je menší nebo větší než zadaná hodnota.

Například následující příkaz změní barvu textu na modrou za podmínky, že dokument má orientaci stránky na šířku, šířka textu je menší než 20 cm a používají se dva sloupce.

```
\mediaquery{orientation=landscape,
max-textwidth=20cm,
twocolumn=true}{\color{blue}}{}
```

Automatická sazba

Lua_T_E_X nám umožňuje automaticky řešit některé problémy, které bylo v minulosti třeba řešit ručním zásahem do dokumentu. Díky tomu, že proces sazby lze ovlivňovat pomocí callbacků v jazyce Lua, můžeme například omezit výskyt

Text s krátkými souhláskami a samohláskami i dalšími jevy z nabídky možností (v textu možnými).
 Co třeba í znaky š diakritikou?
 Různé možnosti [v závorkách < i jiných znacích
 Podpora iniciál a titulů: M. J. Hegel, Ing. Běháková, Ph.D., Ž. Zíbrt, Ch. Borner.
 Podpora jednotek: 100,5 MN-s, 100,5 kJ, 200 μA,
 – 1 dag, 1 MB.
 Uvnitř matematiky by mělo být zpracování vypnuté: $k \in \mathbb{N}$.

Obrázek 6: Ukázka užití balíčku Luavlna

takových typografických chyb, jako jsou vdovy a sirotci, jednopísmenné předložky na koncích řádků nebo špatně zalomené odstavce. V této sekci si ukážeme několik balíčků, které tyto chyby řeší.

Balíček *Luavlna*

Podle českých a slovenských typografických zvyklostí by se na koncích řádků neměly vyskytovat jednopísmenné předložky nebo spojky. Tomu se dá v $\text{T}_{\text{E}}\text{X}$ u zabránit vložením znaku ~ mezi předložku a následující slovo. Pro zjednodušení této činnosti existuje několik nástrojů, například program *Vlna* Petra Olšáka nebo balíček *Encavlna* vytvořený také Petrem Olšákem a Zdeňkem Wagnerem.

Balíček *Luavlna* [2] využívá možnosti $\text{LuaT}_{\text{E}}\text{X}$ u pro úpravu plně zpracovaných uzlů textu před zalomením do odstavců. V tomto stadiu jsou již všechna makra expandována, nezlomitelná mezera tedy může být snadno vložena na všechna místa, kde by se měla nacházet.

Luavlna zamezuje zalomení řádků v následujících situacích:

- za jednoznakovými předložkami a spojkami
- u iniciál
- u akademických titulů
- mezi čísla a jednotkami

Ukázku použití balíčku můžete vidět na Obrázku 6. Volby se zadávají pouze pomocí voleb balíčku: `\usepackage[<volby>]{luavlna}`. Nezlomitelné mezery jsou zvýrazněné růžově díky použití volby `debug`. Balíček nabízí řadu dalších voleb, například pro zakázání vkládání nezlomitelných mezer v určitých situacích:

noproccess – nespouštět zpracování dokumentu automaticky

noinitials – nezpracovávat iniciály

nounits – nezpracovávat SI jednotky

nopredegrees – nezpracovávat tituly před jménem

nosufdegrees – nezpracovávat tituly za jménem

Nastavení jednotlivých jazyků se provádí pomocí těchto příkazů:

- `\singlechars{jazyk}{písmena}` – seznam písmen, u kterých se potlačuje zalamování řádků.
- `\enablesplithypens{jazyk}` – nastaví podporu zalamování spojovníků pro daný jazyk.
- `\preventsingleng{jazyk}` – nastaví pravidla pro daný jazyk pro celý dokument.

Jazyk lze vybrat pomocí balíčků *Babel* nebo *Polyglossia*. Následující příklad ukazuje, že v českém textu se vloží nezlomitelná mezera, po změně jazyka na angličtinu pomocí `\selectlanguage{english}` se zpracování mezer vypne.

A příklad česky.

```
\selectlanguage{english}
```

A thing.

A příklad česky. A thing.

Zpracování celého dokumentu podle českých pravidel bez ohledu na aktuálně zvolený jazyk lze dosáhnout pomocí příkazu `\preventsingleng{czech}`.

```
\preventsingleng{czech}
```

Příklad v češtině.

```
\selectlanguage{english}
```

A thing.

Příklad v češtině. A thing.

Balíček *Linebreaker*

Balíček *Linebreaker* [4] brání přetečení textu boxů a odstavců. Příklad jeho použití můžete vidět na Obrázku 7, kde zamezí přetečení několika řádků při sazbě do úzkého sloupce.

Linebreaker využívá callback Lua_{TeX}u `linebreak_filter`, který řídí řádkový zlom. *Linebreaker* nahrazuje výchozí funkci pro řádkový zlom upravenou verzí, která detekuje přetečení nebo podtečení lámaného textu. Při detekci tohoto problému jej sází znovu s většími hodnotami `\tolerance` a `\emergencystretch`, dokud se přetečení nepotlačí nebo dokud se nedosáhne maximální hranice `\tolerance`.

Výhodou je, že tyto změny `\tolerance` a `\emergencystretch` jsou lokální pouze pro aktuálně lámaný odstavec, zbytek textu nijak neovlivní.

Balíček *Linebreaker* můžeme konfigurovat zadáním voleb balíčku `\usepackage[<volby>]{linebreaker}` nebo později i v těle dokumentu s využitím příkazu `\linebreakersetup{<volby>}`:

maxcycles – počet pokusů o přesázení odstavce.
maxemergencystretch – maximální hodnota `\emergencystretch`.
maxtolerance – maximální hodnota `tolerance`.

```
\linebreakersetup{
maxtolerance = 90,           % výchozí 8189
maxemergencystretch = 1em,  % výchozí 3em
maxcycles = 4                % výchozí 30
}
```

Balíček *lua-widow-control*

Vdovy a sirotci (souhrně také parchanty) jsou typografickou chybou, která vzniká při zalomení stránek. Osamocený řádek odstavce se někdy ocitne na začátku nebo konci stránky. Sirotek je osamocený řádek na začátku stránky, vdova pak na konci. V TeXu existují dvě penalty, `\clubpenalty` a `\widowpenalty`, které při nastavení vdovy a sirotky potlačují, ovšem za cenu rozšíření vertikálních mezer. Balíček *Lua-widow-control* [3] používá jiný přístup. Každý odstavec sází dvakrát – jednou s výchozími parametry, druhý o jeden řádek delší. To může mít vliv na rychlost kompilace dokumentu, v praxi by ale měl být malý. V případě, že se při zalomení stránky detekuje parchant, vymění se předposlední odstavec na stránce za jeho verzi s delším řádkem, čímž dojde k posunutí problematického řádku.

Balíček *Lua-widow-control* lze nastavit pomocí několika voleb. Ty je možné zadat jako čárkou oddělené volby v příkazu `\usepackage[<volby>]{lua-widow-control}`, nebo pomocí příkazu `\lwcsetup{<volby>}`.

The example document given below creates two pages by using Lua code alone. You will learn how to access TeX's boxes and counters from the Lua side, shipout a page into the PDF file, create horizontal and vertical boxes (hbox and vbox), create new nodes and manipulate the nodes links structure.

The example document given below creates two pages by using Lua code alone. You will learn how to access TeX's boxes and counters from the Lua side, shipout a page into the PDF file, create horizontal and vertical boxes (hbox and vbox), create new nodes and manipulate the nodes links structure.

(a) Bez balíčku *Linebreaker*

(b) S balíčkem *Linebreaker*

Obrázek 7: Příklad využití balíčku *Linebreaker*

Následující volby umožňují změnit metodu použitou pro zamezení parchantů: **default** – nepřidává žádné vertikální mezery, používá pouze prodloužení odstavců.

Tato metoda by měla odstranit 95 % parchantů, může ovšem vést k příliš vzdušným odstavcům.

strict – přísnější nastavení, které dbá, aby změny zalomení odstavců byly co nejméně postřehnutelné. Odstraní ovšem pouze třetinu parchantů.

balanced – nejdříve přidává vertikální mezery, pokud to nestačí, prodlouží odstavce. Tato metoda odstraní až 90 % parchantů.

Tento výčet je pouze malým úvodem k balíčku *Lua-widow-control*. Podrobný popis naleznete v článku Maxe Chernoffa [7], který vyšel v loňském Zpravodaji.

Odkazy

1. HOFTICH, Michal. *The Responsive package* [online]. 2023. Ver. 0.1 [cit. 2023-03-30]. Dostupné z: <https://www.ctan.org/pkg/responsive>.
2. HOFTICH, Michal. *The Luavlna package: Prevent line breaks after single letter words, units, or academic titles* [online]. 2023. 0.1k [cit. 2023-03-30]. Dostupné z: <https://ctan.org/pkg/luavlna>.
3. CHERNOFF, Max. *The Lua-widow-control package: Automatically remove widows and orphans from any document* [online]. 2022. Ver. 3.0.0 [cit. 2023-03-30]. Dostupné z: <https://ctan.org/pkg/lua-widow-control>.
4. HOFTICH, Michal. *The Linebreaker package: Prevent overflow boxes with LuaLaTeX* [online]. 2023. Ver. 0.1c [cit. 2023-03-30]. Dostupné z: <https://ctan.org/pkg/linebreaker>.
5. CAVAZZA, Edoardo. Modern CSS Techniques To Improve Legibility. *Smashing Magazine* [online]. 2020 [cit. 2023-03-30]. Dostupné z: <https://www.smashingmagazine.com/2020/07/css-techniques-legibility/>.
6. MORTENSEN, Spencer. *The Typographic Scale* [online]. 2011. [cit. 2023-03-30]. Dostupné z: <https://spencermortensen.com/articles/typographic-scale/>.
7. CHERNOFF, Max. Automatically Removing Widows and Orphans with lua-widow-control. *Zpravodaj Československého sdružení uživatelů TeXu*. 2022, roč. 32, č. 1, s. 49–76. Dostupné z DOI: 10.5300/2022-1-4/49.

Summary: Responsive Design and Automatic Typesetting with Lua^AT_EX

This article focuses on the use of responsive design techniques to display web pages on devices with different display sizes, such as mobile phones, tablets, large monitors and printers. These methods allow optimizing the readability of

a document on all devices by using different font sizes, individual page elements, and margins.

We present how similar functionality can be achieved using L^AT_EX. Specifically, it focuses on the use of LuaL^AT_EX for automated typesetting, using packages *Responsive* for setting font size and line spacing according to page size, *Luavlna* to prevent the occurrence of single-letter prepositions at line breaks, *Lua-widow-control* to reduce orphan lines at page breaks and page starts, and *Linebreaker* to prevent line overflow.

With these methods, a single source document can be used for different outputs, such as print versions, e-book readers, and web pages, and achieve optimal document display on all devices.

Keywords: automatic typesetting, responsive design, LuaL^AT_EX

Michal Hoftich, michal.h21@gmail.com