



Wojciech Myszka

Historia pewnej strony WWW...
...albo czy \LaTeX 2HTML nadaje się do tworzenia
serwisów WWW?

Bachotek, 1–3 maja 2003

Strona główna

Strona tytułowa

Spis treści



Strona 1 z 21

Powrót

Full Screen

Zamknij

Koniec

Spis treści

1. Konwersja z Worda	3
2. Konwersja do HTML	5
2.1. TTH	6
2.2. HEVEA	7
2.3. PyL ^A T _E X	8
2.4. L ^A T _E X2HTML	9
2.5. T _E X4ht	10
3. Możliwości L^AT_EX2HTML	11
4. Wsparcie dla CSS w L^AT_EX2HTML	13
5. Projekt strony	19
6. Realizacja	21



Strona główna

Strona tytułowa

Spis treści



Strona 1 z 21

Powrót

Full Screen

Zamknij

Koniec



[Strona główna](#)

[Strona tytułowa](#)

[Spis treści](#)



[Strona 2 z 21](#)

[Powrót](#)

[Full Screen](#)

[Zamknij](#)

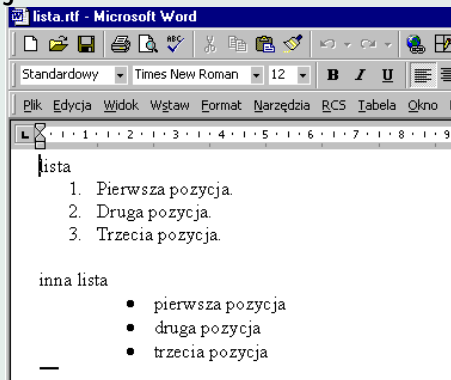
[Koniec](#)

Uzasadnienie

1. Chęć przyjazdu do Bachotka.
2. Apel organizatorów żeby coś wygłosić.
3. Wnioski (Smętne) z rozważań o tym czym są dziś strony (serwisy) WWW.
4. Chęć podzielenia się pewnymi przemyśleniami osoby, która **musi** zajmować się utrzymaniem stron WWW, a nie zna się na tym zupełnie...

1. Konwersja z Worda

- `rtf2LATEX2ε`. Niezły, ale... Najprostsza konstrukcja listowa wyglądająca w Wordzie jak na poniższym rysunku, tłumaczona jest...



```
lista\\  
1.\tab  
Pierwsza pozycja.\\  
2.\tab Druga pozycja.\\  
3.\tab Trzecia pozycja.
```

```
inna lista\\  
{\textbullet}\tab  
pierwsza pozycja\\  
{\textbullet}\tab druga pozycja\\  
{\textbullet}\tab trzecia pozycja
```

Nieco lepiej jest z tytułami (ale wprowadzanimi z użyciem stylów).

- Konkurentem może być `Word2TEX`; nie jest darmowy. Większe możliwości.
- Całkiem przyzwoicie (na prostych dokumentach) sprawdza się `wvWare` (zwłaszcza jak się wybierze konwersję do tekstu lub HTMLa).



Strona główna

Strona tytułowa

Spis treści



Strona 3 z 21

Powrót

Full Screen

Zamknij

Koniec



- AbiWord. Podobno pozwala na eksport do \LaTeX a.
- Antiword – pozwala na konwersje do tekstu.
- Jeżeli z jakichś względów nie zechcemy z powyższych metod korzystać – pozostaje tylko metoda „Cut&Paste” (z której, w istocie, skorzystałem).
- Trzeba będzie popatrzeć też na Writer2 \LaTeX – program, który ma pozwolić na konwersję z OpenOffice/StarOffice do \LaTeX a. Na razie wersja 0.2, ale już ma zestaw stylów, który powinien pozwalać na bezproblemowy(?) eksport dokumentów z OpenOffice do \LaTeX .

Strona główna

Strona tytułowa

Spis treści



Strona 4 z 21

Powrót

Full Screen

Zamknij

Koniec

2. Konwersja do HTML

Gdzie szukać informacji:

— (L^A)T_EX conversion to HTML w UK List of T_EX Frequently Asked Questions.

<http://www.tex.ac.uk/cgi-bin/texfaq2html?label=LaTeX2HTML>

— W zestawieniu narzędzi na serwerze W3C „Word Processors Filters”.

http://www.w3.org/Tools/Word_proc_filters.html

— W katalogu Google: Computers → Software → Typesetting → TeX → Converters. <http://directory.google.com/Top/Computers/Software/Typesetting/TeX/Converters/>

Do najpopularniejszych należą:

— TTH <http://hutchinson.belmont.ma.us/tth/>

— T_EX4ht <http://www.cis.ohio-state.edu/~gurari/TeX4ht/mn.html>

— L^AT_EX2HTML <http://www.latex2html.org/>

— HEVEA <http://para.inria.fr/~maranget/hevea/>



Strona główna

Strona tytułowa

Spis treści



Strona 5 z 21

Powrót

Full Screen

Zamknij

Koniec

2.1. TTH

- Jeden z podstawowych problemów to licencja.
- Drugi – wersja darmowa tworzy jeden długi plik HTMLowy (bez podziału na fragmenty).
- Większe możliwości daje wersja komercyjna.
- Dostępne wersje pracujące zarówno w środowisku Windows jak i Linux/Unix.



[Strona główna](#)

[Strona tytułowa](#)

[Spis treści](#)



[Strona 6 z 21](#)

[Powrót](#)

[Full Screen](#)

[Zamknij](#)

[Koniec](#)

2.2. HEVEA

W pewnych sytuacjach może wymagać dodatkowej konfiguracji przeglądarki – niestety widać wyraźnie różnicę w prezentacji symboli pomiędzy IE a Mozillą (na korzyść tej ostatniej). Nie przypominam sobie aby oferowała wsparcie dla naszych literek. Dostępne wersje pracujące zarówno w środowisku Windows jak i Linux/Unix.



[Strona główna](#)

[Strona tytułowa](#)

[Spis treści](#)



[Strona 7 z 21](#)

[Powrót](#)

[Full Screen](#)

[Zamknij](#)

[Koniec](#)

2.3. Py~~L~~TeX

Bardzo ciekawe rozwiązanie – konwerter napisany w języku Python. Jak go oglądałem – był w bardzo wstępnej fazie rozwoju. Jak dziś patrzę, nie rozwinął się zbyt wiele. Teoretycznie daje szansę na konwersję do postaci HTML lub XML. Autor jako jedną z głównych zalet wskazuje fakt zaprogramowania go w języku Python, który ma być łatwiejszy od Perla.



[Strona główna](#)

[Strona tytułowa](#)

[Spis treści](#)



[Strona 8 z 21](#)

[Powrót](#)

[Full Screen](#)

[Zamknij](#)

[Koniec](#)

2.4. \LaTeX 2HTML

Najstarszy: 10lat! Ale okres najbardziej bujnego rozwoju ma już poza sobą. Dający ogromne możliwości rozbudowy, ale napisany w Perlu, co dla niektórych (w tym i mnie) jest przeszkodą nie do przebycia. Dobrze obsługuje język polski, zawiera wsparcie dla CSSa. Dostępne wersje pracujące zarówno w środowisku Windows jak i Linux/Unix (wymaga instalacji Perla).

Daje duże możliwości konfiguracji z poziomu parametrów. No, i używam go już od ładnych paru lat.



[Strona główna](#)

[Strona tytułowa](#)

[Spis treści](#)



[Strona 9 z 21](#)

[Powrót](#)

[Full Screen](#)

[Zamknij](#)

[Koniec](#)

2.5. T_EX4ht

Z pewnych względów wydaje się najbardziej elastycznym narzędziem. Co więcej wykorzystuje natywne właściwości systemu L^AT_EX (choć trzeba przyznać, że tworzone pliki DVI są tak przetadowane poleceniami `special`, że mój yap odmawiał posłuszeństwa. Z drugiej strony – **niemożliwy** do zainstalowania: w MiKTeXu, którego używam, istnieje, co prawda, odpowiedni pakiet, ale pozbawiony plików wykonywalnych. Te trzeba ściągać ze stron Autorów. A i tak nie działał. . . Dokumentacja zagmatwana. . . Wersja z TL7 (Linux) odmawiała współpracy z posiadanym ImageMagickiem. Szybko dałem sobie spokój.



[Strona główna](#)

[Strona tytułowa](#)

[Spis treści](#)



[Strona 10 z 21](#)

[Powrót](#)

[Full Screen](#)

[Zamknij](#)

[Koniec](#)

3. Możliwości \LaTeX 2HTML

- Bardzo dobry do konwersji „książka” → HTML.
- Ogromna możliwość konfiguracji:
 - zmienne środowiska,
 - parametry w „wierszu polecenia” ,
 - zmienne „z wnętrza pliku źródłowego” ,
 - pliki stylu i odpowiadające im procedury w języku Perl.
 - możliwość przedefiniowywania procedur istniejących i pisania własnych.
- Podstawowe wsparcie dla różnych języków.
- Generacja HTML w wersjach 2.0, 3.0, 3.2, 4.0, 4.1.
- Wsparcie dla CSS
- „Sympatyczna” konwersja wzorów matematycznych (do obrazków i symboli).



Strona główna

Strona tytułowa

Spis treści



Strona 11 z 21

Powrót

Full Screen

Zamknij

Koniec



Język polski

- Zarówno **polski.sty** jak i **polish.sty**.
- (Teoretycznie) można używać zarówno notacji „ciachowej” /a, /c... jak i „babelowej” "a, "c,....
- Najlepiej używać **iso-8859-2**.
- Jeden plik, gdzie można wszystko zmienić.
- Kiepsko z sortowaniem (indeksu).

Notacja „ciachowa” stwarza szereg problemów. W zasadzie łatwo to poprawić, ale...

```
sub main'polski_translation {
  local($_) = 0;
  s/^(^|\G|[^<]|[^\\] (\\|\\)+)\\s*(''|;SPMlt;|;SPMgt;|\\|-|\\/|=|\\| | [aelxcnoszAELXCNOsz])/
  $1.&get_polski_specials($3)/geom;
  $_;
}
```

Strona główna

Strona tytułowa

Spis treści



Strona 12 z 21

Powrót

Full Screen

Zamknij

Koniec

4. Wsparcie dla CSS w $\text{\LaTeX}2\text{HTML}$

- $\text{\LaTeX}2\text{HTML}$ automatycznie tworzy plik zawierający bardzo podstawowe informacje; można go ręcznie modyfikować; w kolejnych przebiegach – nie będzie on nadpisany.
- W przypadku gdy generujemy HTML w wersji 4.0 lub wyższej pojawiają się dodatkowe możliwości.
 - Zmiany kroju i wielkości czcionki oraz wszystkie środowiska wewnętrzne (*inline*) – tłumaczone są na elementy SPAN.
 - Środowiska i wszystko to co jest składane w trybie blokowym – tłumaczone jest na elementy DIV. Tablice (środowiska `table` i `array` tłumaczone jest na elementy TABLE; środowiska typu `verbatim` na PRE.
 - Z jakichś powodów listy tłumaczone są na HTMLowe `...`, `...`,
- Za pomocą dodatkowych parametrów środowisk można dodawać i definiować selektor klasy albo specyfikację stylu.

Nie jest to zbyt konsekwentnie zrealizowane, ale to tylko moja prywatna opinia.



[Strona główna](#)

[Strona tytułowa](#)

[Spis treści](#)



[Strona 13 z 21](#)

[Powrót](#)

[Full Screen](#)

[Zamknij](#)

[Koniec](#)



[Strona główna](#)

[Strona tytułowa](#)

[Spis treści](#)



[Strona 14 z 21](#)

[Powrót](#)

[Full Screen](#)

[Zamknij](#)

[Koniec](#)

Konstrukcja źródłowa:

```
\textbf{Ala ma kota}  
\large Ala ma kota
```

generuje w pliku HTML:

```
<SPAN CLASS="textbf">Ala ma kota</SPAN>  
<BIG CLASS="LARGE">Ala ma kota </BIG>
```

a w pliku stylu:

```
BIG.LARGE      { }  
SPAN.textbf    { font-weight: bold }
```



Konstrukcja źródłowa:

```
\begin{flushright}  
zawartość  
\end{flushright}
```

generuje w pliku HTML:

```
<DIV ALIGN="RIGHT">zawartość</DIV>
```

a w pliku CSS:

```
DIV.flushright { }
```

ale to jeszcze nie to o co nam chodzi...

[Strona główna](#)

[Strona tytułowa](#)

[Spis treści](#)



Strona 15 z 21

[Powrót](#)

[Full Screen](#)

[Zamknij](#)

[Koniec](#)



Konstrukcja źródłowa:

```
\begin[]{\flushright}  
zawartość  
\end{\flushright}
```

generuje w pliku HTML:

```
<DIV ALIGN="RIGHT" ID="flushright27"  
CLASS="flushright">zawartość</DIV>
```

a w pliku CSS:

```
DIV.flushright      {   }  
#flushright27      {   }
```

Jest już znacznie lepiej...

[Strona główna](#)

[Strona tytułowa](#)

[Spis treści](#)



[Strona 16 z 21](#)

[Powrót](#)

[Full Screen](#)

[Zamknij](#)

[Koniec](#)



Konstrukcja źródłowa:

```
\begin[ala|color=red]{flushright}  
zawartość  
\end{flushright}
```

generuje w pliku HTML:

```
<DIV ALIGN="RIGHT" ID="flushright29"  
CLASS="ala">zawartość </DIV>
```

a w pliku CSS:

```
DIV.ala          {   }  
#flushright29   { color : red }
```

Co już zaczyna mieć sens, zwłaszcza że posiadamy dodatkowe polecenia pozwalające definiować styl klasy:

```
\htmlsetstyle [DIV] {ala} {text-decoration=underline}
```

Strona główna

Strona tytułowa

Spis treści



Strona 17 z 21

Powrót

Full Screen

Zamknij

Koniec



Dodatkowo program elementom nawigacyjnym również nadaje odpowiednie klasy i identyfikatory. I tak:

— Panel Nawigacyjny opisany jest jako:

```
<DIV CLASS="navigation">  
  <!--Navigation Panel-->  
</DIV>
```

— dodatkowo Spis Treści oznaczony jest jako:

```
<UL CLASS="TofC">  
  <!--Table of Contents-->  
</UL>
```

— oraz „małe spisy treści” jako:

```
<UL CLASS="ChildLinks">  
  <!--Table of Child-Links-->  
</UL>
```

Strona główna

Strona tytułowa

Spis treści



Strona 18 z 21

Powrót

Full Screen

Zamknij

Koniec

5. Projekt strony



Strona główna

Strona tytułowa

Spis treści



Strona 19 z 21

Powrót

Full Screen

Zamknij

Koniec



- Spis treści (ale tylko na najwyższym poziomie).
- Kolejne „podmenu” nawigacyjne w ramach mniejszych fragmentów.
- Indeks nazwisk.

- Bez ramek.
- Bez tabelek.
- Jednakowy (na ile się da) wygląd i w IE i w Mozilli
- Znośny wygląd w przeglądarkach tekstowych

Strona główna

Strona tytułowa

Spis treści



Strona 20 z 21

Powrót

Full Screen

Zamknij

Koniec

6. Realizacja

- Niestety nie udało się tego zrealizować za pomocą standardowo dostępnych możliwości pakietu \LaTeX 2HTML.
- Skończyło się „przemieszcaniem” zawartości z kodem odpowiedzialnym za wygląd i nawigację (wszystko co się dało zostało przeniesione do CSS)
- W miarę wolnego czasu dam szansę pakietowi \TeX 4ht



[Strona główna](#)

[Strona tytułowa](#)

[Spis treści](#)



[Strona 21 z 21](#)

[Powrót](#)

[Full Screen](#)

[Zamknij](#)

[Koniec](#)