



# Politechnika Wroclawska

Jak przeżyć w nieprzyjaznym środowisku  
WYSIWYG



Wojciech Myszka  
BachTeX 2006



# O czym będzie. . . I

O czym będzie. . .

Geneza

Cele

Prerekwizyty

Metodologia

Kilka słów o Instrukcji...

Fonty

Tytularia

Tytuł



# O czym będzie. . . II

Dwujęzyczne podpisy pod rysunkami

Podsumowanie I

Konwersja  $\LaTeX$  do. . .

$\TeX$ 4ht

Konwersja do WYSIWYG

Konwersja do formatu MS Word

$\TeX$ 4ht – osvajanie

Formatowanie wizualne

Podsumowanie II



# O czym będzie... III

Kiedy warto



# Geneza

Przychodzi student do doktora. . .



# Cele

1. Pokazać, że można style (klasy) modyfikować.
2. Pokazać jak się za to zabrać.
3. Pokazać, że nie trzeba (bardzo) dużo wiedzieć.
4. Wskazać możliwości (słabości) różnych podejść do zagadnienia.

Celem nie było przedstawienie **gotowego** rozwiązania!



# Prerekwizyty

1. Umiejętność czytania ze zrozumieniem.
2. Ogólna znajomość  $\LaTeX$ a (TDS, style, klasy, podstawowe polecenia: `newcommand`, `renewcommand`, `newenvironment`, `renewenvironment`, . . .).
3. Niezłe rozeznanie (różnych) języków programowania (makra, konstrukcje if-then-else, bloki, zmienne lokalne, globalne, . . .) w zakresie czytania i bardzo ogólnego rozumienia kodu.
4. Jakieś doświadczenie programistyczne.
5. Język angielski.



# Metodologia postępowania

1. Należy przeczytać (ze zrozumieniem!) Instrukcję.
2. Zaopatrzyć się w ulubiony edytor tekstu (pozwalający na raz przeglądać kilka(naście) plików).
3. Na podorędziu zgromadzić literaturę (The T<sub>E</sub>XBook, Podręcznik L<sup>A</sup>T<sub>E</sub>Xa, T<sub>E</sub>X for the impatient, pewno coś jeszcze)





## Kilka słów o Instrukcji. . .

1. Zazwyczaj dostaje się ją na papierze, choć coraz częściej dostępna jest elektronicznie.
2. Jest obrazowa, to znaczy sformatowana zgodnie z wymaganiami, które opisuje.
3. Instrukcja zakłada że praca musi być przygotowana w edytorze Word (nie dostrzegając niczego innego).
4. Zazwyczaj **nie jest** to Wordowy plik .dot.
5. Bardzo często napisana jest przez fachowców – więc zawiera trudne sformułowania.



# Metodologia

## Założenia projektowe

1. Użyjemy klasy `mwart` z pakietu `mwcls` Marcina Wolińskiego.
2. Podstawowa wielkość czcionki to 11pt.
3. Budować będziemy „klasę pochodną” (wystarczający opis jest w dokumentacji `mwcls`).



# Fonty

Instrukcja nakłada szereg wymagań dotyczących,

- ▶ kroju,
- ▶ wielkości czcionki,
- ▶ „interlinii”



# Fonty

## Krój

- ▶ Zgodnie z oczekiwaniem – wymaganym krojem czcionki jest Times New Roman.
- ▶ Font już mocno spowszedniał, ale najprawdopodobniej zakłada się, że jest wszędzie tam gdzie MS Office
- ▶ Standardowo używane przez  $\text{\LaTeX}$  fonty CMR/PLR/LMR są równie opatrzone. . .
- ▶ . . . i nie różnią się specjalnie od TNR



# Fonty

## Krój

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.



# Fonty

## Krój

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

LMR

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

QTM

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

TNR

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

PTM



# Fonty

## Krój – wnioski

Decyzja o wyborze fontu powinna należeć do realizatora: najłatwiej będzie zastosować font QuasiTimes, Można próbować użyć pakietu **winfonts** i zastosować Times New Roman (licząc się z tym, że  $\LaTeX$  ciągle ma problemy z dołączaniem fontów TTF/OTF.

Widać też, że użycie fontu LMR/CMR/PLR – nie powoduje straszego zamieszania



# Fonty

## Wielkość

Instrukcja bardzo precyzyjnie definiuje użyte wielkości czcionek podając dwa parametry:

*. . . wielkość pisma 11 punktów, odstęp między wierszami przynajmniej 13 punktów. . .*





# Interlinia – dygresja

Zainteresował mnie problem ustawiania wielkości fontu w Wordzie. Po naiwnym zastosowaniu wielkości 11pt uzyskałem następujący efekt:

Garamond	TNR	CMR10	Tahoma
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.



# Fonty

## Wielkość

W  $\text{\LaTeX}$ owych klasach wszystkie „sprawy” związane z wielkością fontów:

```
\documentclass[11pt]{article}
```

zawarte są zadeklarowane są w plikach .clo (mw11.clo w klasach mwcls).

```
\renewcommand\normalsize{%  
  \@setfontsize\normalsize\@xipt{13.6}%  
  \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@  
  \abovedisplayshortskip \z@ \@plus3\p@  
  \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@  
  \belowdisplayskip \abovedisplayskip  
}  
\normalsize
```



# Fonty

## Wielkość cd – dla początkujących

- ▶ `\renewcommand` to polecenie pozwalające przedefiniować inne polecenie, wcześniej zdefiniowane
- ▶ `\normalsize` to polecenie definiujące wszystko to co związane jest z domyślną wielkością pisma
- ▶ `@` znaczek, którego użycie w nazwie czyni polecenie niedostępnym dla „zwykłego użytkownika”
- ▶ `\p@`, `\z@`, `\@xipt` to (często używane) stałe. . . (zdefiniowane w pliku `latex.ltx`)
- ▶ polecenie `\@setfontsize` ma trzy parametry: drugi to wielkość fontu, trzeci interlinia
- ▶ `%` to znak komentarza – jego „nadużywanie” wymaga dalszych wyjaśnień



# Fonty

## Wielkość

Podobnie trzeba przyrzeć się i innym wielkościom pisma wymaganym przez Instrukcję:

- ▶ tytuł pracy powinien być składany fontem 13/15 punktów,
- ▶ treści „poboczne” (streszczenie, literatura, słowa kluczowe) fontem 9/11 punktów.

i poszukać najbliższych odpowiedników wśród standardowych wielkości. Później należy dokonać ewentualnych korekt definicji. . .

Przeddefiniujemy zatem polecenie `\large`, żeby generowało tekst w wielkości 13/15 punktów.



# Fonty

## Wielkość – secondarysize

Sympatyczną cechą klas mwcls jest wprowadzenie polecenia `\secondarysize` którym składane są „poboczne” części dokumentu (oraz ilustracje i tabele).

Wskazana wielkość pisma to 9/11 punktów czemu odpowiada w (w pliku `mw11.clo`) `\footnotesize`.



# Tytularia

Zmiany w tytulariach (w klasach mwcls) to prawdziwa przyjemność.

Instrukcja dpuszcza dwa poziomy tytułów `\section` i `\subsection`, nakazując centrowanie, pisanie wersalikami i zadając odstępy przed i po tytule.

Odpowiedni efekt uzyskujemy bardzo łatwo

```
\renewcommand{\sectsetup}{\rmfamily}
                    % standardowo bfseries!
\SetSectionFormatting{section}
    {1cm}
    {\FormatBlockHeading{\centering\normalsize
    \MakeUppercase}}
    {0.7cm}
```



# Tytuł

- ▶ Tytuł generowany jest na podstawie metainformacji (nazwisko Autora, data, tytuł pracy) poleceniem `\maketitle`.
- ▶ Instrukcja nakłada obowiązek umieszczenia na początku również słów kluczowych, które nie są standardowym atrybutem tytułu klas  $\LaTeX$ owych.
- ▶ Potrzebna będzie modyfikacja.



# Tytuł

## Modyfikacja istniejącego polecenia

(Naiwne) postępowanie jest następujące:

1. Znajdujemy polecenie, które chcemy modyfikować. . .
  - ▶ . . . szukając go najpierw w pliku klasy. . .
  - ▶ sięgając do źródeł  $\LaTeX$ a (plik `latex.ltx`) gdy to nie da rezultatu.
2. W miarę możliwości staramy się zasadę działania polecenia zrozumieć.
3. Projektujemy i wdramy zmiany – to znaczy wkładamy odpowiedni fragment kodu „do siebie” i zaczynamy go przerabiać.





# Tytuł

## Modyfikacja polecenia maketitle

- ▶ Polecenie `\maketitle` korzysta z metainformacji definiowanych poleceniami pomocniczymi: `\date`, `\author`, `\title`.
- ▶ Przez analogię dodamy polecenie `\keywords`

```
\def\keywords#1{\gdef\@keywords{#1}}  
\def\@keywords{\@latex@error{No  
  \noexpand\keywords given}\@ehc}
```



# Tytuł

## Realizacja layoutu

Jak przypatrzeć się poleceniu `\maketitle` to (pomijając szczegóły) wygląda ono tak:

```
\newcommand\maketitle{
  \begingroup
    \renewcommand\thefootnote{...}%
    \long\def\@makefnote##1{...}%
    \if@twocolumn ...
    \else ...
    \@maketitle
    \fi
  \endgroup
  \global\let\maketitle\relax
  ...}
```



# Tytuł

## Realizacja layoutu – cd

- ▶ Cała „czarna” robota wykonywana jest przez `\@maketitle`.
- ▶ W ten sposób realizowanych jest bardzo wiele poleceń!
- ▶ Jeżeli nie potrzebujemy ingerować w „czynności ogólne” – nie musimy ich przeddefiniowywać. (My musieliśmy zaingerować. . . )
- ▶ Natomiast same `\@maketitle` – to już programowanie „wizualne”(?) gdzie z pudełek i odstępów trzeba poskładać sensowną całość – co wcale nie musi być bardzo proste.



# Podpisy pod rysunkami I

- ▶ Instrukcja nakłada obowiązek realizowania dwujęzycznych podpisów pod rysunkami.
- ▶ Jest to trudniejszy przykład modyfikacji polecenia – w tym wypadku – `\caption`
- ▶ Polecenie to nie jest zdefiniowane w pliku klasy – znajdziemy je w pliku `latex.ltx` – to jest ten nieszczęsny „format”, który czasami trzeba generować.



## Podpisy pod rysunkami II

- ▶ Polecenie zrealizowane jest bardzo podobnie do poprzedniego – w podstawowej definicji wykonuje się różne czynności organizacyjne.
- ▶ Czarną robotę wykonuje polecenie `\@caption` – zdefiniowane w pliku klasy.
- ▶ Ponieważ `\caption` działa tylko w otoczeniach `figure` i `table` – analiza nie jest prosta. . .



## Podpisy pod rysunkami III

- ▶ Co najważniejsze – zdefiniowane polecenie `\caption` nie wskazuje ile parametrów powinno mieć – co jest ważne: chcemy dodać drugi, obowiązkowy parametr – podpis po angielsku.



# Podpisy pod rysunkami

- ▶ Jak obejrzymy polecenie `\caption` to od razu zauważymy „trzecie(?) dno” – samym formatowaniem podpisu zajmuje się kolejne polecenie: `\makecaption`
- ▶ Proponowane zmiany sprowadzają się do tego aby polecenie `\makecaption` wywołać dwa razy z parametrami podpisu polskiego i angielskiego.
- ▶ Trudność(?) polega na tym, że to samo polecenie `\caption` obsługuje podpisy tabel i rysunków – pomaga na to odpowiednia „parametryzacja” poleceń.



# Podsumowanie I

1. Postępując konsekwentnie według podanych przepisów można uzyskać dzieło, które – w warstwie wizualnej – odpowiada wymaganiom „Instrukcji”.
2. Mamy przykład, dla niektórych idei prezentowanych przez Marcina Wolińskiego („Adiustacja logiczna w praktyce”).
3. Ale co dalej?





# Konwersja $\LaTeX$ do... I

1. Należy przypuszczać, że „Instrukcja” powstała nie po to, żeby ją implementować w  $\LaTeX$ u, ale raczej po to by ułatwić wydawcy szybkie stworzenie dzieła, które mimo, że przygotowane przez wielu autorów – zachowuje jednolity wygląd.
2. Można – w związku z tym – zadać pytanie, czemu zamiast przygotowywać „Instrukcję. . .” prozą nie przygotował [wydawca] odpowiedniego pliku stylu dla Jedynie Słusznego edytora WYSIWYG?



# Konwersja $\LaTeX$ do... II

- Pytanie „Co dalej?” pozostaje w mocy (i sprowadza się, w istocie, do znalezienia konwertera  $\LaTeX \rightarrow ??$  który przeniesie:
  - ▶ zawartość,
  - ▶ strukturę,
  - ▶ atrybuty formatowania.



# T<sub>E</sub>X4ht I

- ▶ Jeden z ciekawszych programów do konwersji T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X do HTML, XHTML i XML (MathML)
- ▶ Na system składa się rozbudowany zestaw makr T<sub>E</sub>Xowych, które „obudowują” konstrukcje T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>Xowe poleceniami special (zapisywanymi do pliku DVI). . .
- ▶ . . . oraz interpreter DVI konwertujący speciale do plików tekstowych.



## T<sub>E</sub>X4ht II

- ▶ Program (System) jest bardzo aktywnie rozwijany, a jego Autor aktywnie udziela się na grupie comp.text.tex rozwiązując różne problemy.
- ▶ System jest rozbudowany i czasami sprawia problemy z instalacją. Na szczęście wersje T<sub>E</sub>XLive i proT<sub>E</sub>Xt działają bardzo dobrze.
- ▶ Dokumentacja istnieje, choć – czasami – jest mocno zagmatwana.
- ▶ Wsparcie dla klas mwcls stwarza drobne problemy.



# TeX4ht III

- ▶ **Najważniejszą cechą systemu jest to, że (stosunkowo) łatwo może on być rozbudowywany.**



# Konwersja do WYSIWYG I

- ▶ Program może być wykorzystany do konwersji  $\TeX$ a i  $\LaTeX$ a do postaci WYSIWYG. Zostały nawet zdefiniowane odpowiednie polecenia.
- ▶ Systemem docelowym może być bądź OpenOffice (najnowsza wersja produkuje dokumenty w formacie ODT) bądź MS Word (via specjalnie przygotowany plik HTML).
- ▶ W pierwszym przypadku wzory matematyczne konwertowane są via MathML (całkiem niezłe); w drugim – konwertowane do plików rastrowych (różnej jakości).



## Konwersja do WYSIWYG II

- ▶ Z różnych względów wybrałem drugą możliwość, choć bardziej obiecująca jest metoda pierwsza i konwersja do formatu DOC wykonywana przez OO.
- ▶ Tak na marginesie: próba współpracy przy pisaniu artykułu trzech osób (OO i dwie różne wersje Worda) skończyła się przepisaniem tekstu z ostatniego wydruku. . .



# Konwersja do formatu MS Word

- ▶ Zasadnicza część konwersji dokonywana jest przez odpowiednią modyfikację plików CSS, na podstawie których Word buduje kategorie stylu.
- ▶ Część informacji na temat formatowania tekstu zapisywana jest przez Word w dziwacznych, niestandardowych komentarzach – zapewne metodą reverse engineering można to odtworzyć, ale. . .
- ▶ Niestety wzory konwertowane są do plików graficznych – ich jakość jest ograniczona.





# T<sub>E</sub>X4ht – oswajanie

- ▶ Oswajanie T<sub>E</sub>X4ht na pierwszy rzut oka nie jest specjalnie proste.
- ▶ Dokumentacja albo na stronach Autora albo pliki przychodzące wraz z systemem.
- ▶ Bardziej szczegółową dokumentację można uzyskać kompilując jakiś prosty dokument (L<sup>A</sup>T<sub>E</sub>Xowy) poleceniem:  

```
htlatex plik "html,info"
```

i przeglądając powstały plik `.log`

- ▶ Jeszcze bardziej szczegółowa dokumentacja wymaga skompilowania znajdującego się w dystrybucji pliku `mktex4ht.4ht`:

```
htlatex mktex4ht.4ht
```



# Konfiguracja I

Konfiguracja sposobu tłumaczenia poleceń dokonywana jest za pomocą makra `\Configure{co}` o kilku parametrach.

Aby próbować zobaczyć efekt działania poleceń konfiguracyjnych czary są następujące:

```
htllatex filename "html,0.0,hooks"
```



## Konfiguracja II

generuje to plik HTML zawierający „podpowiedzi” pozwalające (częściowo) wyobrazić sobie znaczenie parametrów konfiguracyjnych poszczególnych poleceń.

W przypadku konfiguracji wszystkich tytułarów parametry są cztery:

- ▶ pierwszy i drugi określają zawartość, która umieszczona będzie przed i po jednostce tekstu.
- ▶ parametry trzeci i czwarty określają zawartość, która umieszczona będzie przed i po tytule.



## Konfiguracja III

Aby Dokonać zmian w konfiguracji – przygotowujemy plik o rozszerzeniu `.cfg`. Niech jego zawartość będzie taka:

```
\Preamble{html}
\begin{document}

\Configure{section}{POCZATEK\ }{\ KONIEC}
                {PRZED\ }{\ PO}

\EndPreamble
```



# Konfiguracja IV

Plik źródłowy:

```
\documentclass[12pt]{article}
\begin{document}
  \section{Tytul}
  Ala ma kota
\end{document}
```

Kompilujemy tak (pierwszy parametr to nazwa pliku źródłowego), drugi – parametry dodatkowe, w tym przypadku plik konfiguracyjny):



# Konfiguracja V

```
htlatex aa my
```

A uzyskany wynik (plik HTML) jest następujący:

```
POCZATEK PRZED Tytuł PO
```

```
Ala ma kota KONIEC
```



# Formatowanie wizualne

Formatowanie wizualne „załatwiamy” modyfikacjami pliku CSS.  
(Większość tych informacji jest interpretowana przez Worda)

W przypadku konfiguracji rozdziału sprowadza się do następujących wpisów w pliku konfiguracyjnym:

```
\Css{h3 {margin-top:1cm; margin-bottom:0.7cm;  
text-transform:uppercase; text-align:center;  
font-size:11pt;}}
```

```
\Css{h4 {margin-top:0.7cm; margin-bottom:0.5cm;  
text-transform:uppercase; text-align:center;  
font-size:9pt;}}
```



## Podsumowanie II

1. Można dużo.
2. Konfiguracja  $\text{T}\text{E}\text{X}4\text{ht}$  nie jest prosta, ale możliwa.
3. Droga via HTML jest to ślepa uliczka, ale jest to rodzaj konwertera struktura ( $\text{L}\text{A}\text{T}\text{E}\text{X}$ )  $\rightarrow$  struktura (HTML) i wygląd (styl  $\text{L}\text{A}\text{T}\text{E}\text{X}$ owy)  $\rightarrow$  wygląd (arkusze stylów).





# Kiedy warto?

Ciągle pozostaje otwarte pytanie: Kiedy warto podejmować taką pracę?

**Odpowiedź jest dokładnie znana wszystkim programistom: wtedy, gdy czas potrzebny przygotowanie i uruchomienie systemu jest krótszy niż czas który zaoszczędzimy stosując system.**