

To demo prezentuje zagadnienia związane z formatowaniem tekstu i bazuje na funkcjonalnościach dostarczonych przez klasę dokumentów `pdfArticle`. Więcej informacji (i innych możliwości formatowań) w dokumentacji klas `fontspec`, `xcolor`, `ulem`, `contour` i `shadowtext` używanych przez `pdfArticle`.

- Rozmiar fonu:
 - względem bazowego fonu dokumentu:
 - * pomniejszone: `\tiny \scriptsize \footnotesize \small`
 - * rozmiar bazowy: `\normalsize`
 - * powiększone: `\large \Large \LARGE \huge \Huge`
 - bezwzględny: 17 punktów:
`{\fontsize{17pt}{1.2em}\selectfont tekst}`
 - skalowany: półtora raza:
`{\addfontfeatures{Scale=1.5} tekst}`
- Krój fonu (rodzina, typ):
 - domyślny szeryfowy (roman) (używany standardowo, dopóki nie wybierzemy innego):
`\textrm{tekst}` lub `{\rmfamily tekst}` lub `\rmfamily tekst \normalfont`^{1, 2}
 - domyślny bez szeryfowy (sans):
`\textsf{tekst}` lub `{\sffamily tekst}` lub `\sffamily tekst \normalfont`
 - domyślny stałej szerokości:
`\texttt{tekst}` lub `{\ttfamily tekst}` lub `\ttfamily tekst \normalfont`
 - dowolnie wybrany:
`{\fontspec{DejaVu Sans Mono} tekst}`³
- Odmiany (seria i kształt) fonu (dla aktualnie aktywnej rodziny fontów)⁴:
 - **pogrubiony (Bold)**:
`\textbf{tekst}` lub `{\bfseries tekst}` lub `\bfseries tekst \mdseries`⁵
 - *kursywa (Italic)*:
`\textit{tekst}` lub `{\itshape tekst}` lub `\itshape tekst \upshape`⁶
 - *pochyły (Slanted aka Oblique)*:
`\textsl{tekst}` lub `{\slshape tekst}` lub `\slshape tekst \upshape`
 - KAPITALIKI (SMALL CAPITALS):
`\textsc{tekst}` lub `{\scshape tekst}` lub `\scshape tekst \upshape`
 - r o z r z e d z o n y:
`{\addfontfeatures{LetterSpace=25} tekst}`
 - cyfry nautyczne: 0123456789:

1. `\normalfont` resetuje oprócz kroju także serię i kształt fonu.

2. Dostępna jest także komenda `\textnormal{}` przywracająca domyślny font dokumentu dla napisu podanego w argumentcie.

3. W preambule dokumentu możemy użyć komendy `\newfontfamily`, np. `\newfontfamily{\symf}{Symbola}` i używać go poprzez `\symf tekst`.

Jeżeli użyjemy argumentu `NFSSFamilj`, np. `\newfontfamily{\symf}[NFSSFamilj=mysym]{Symbola}`, font będzie można także wybrać standardowym poleceniem `\fontfamily{mysym}\selectfont` (czego oczekują niektóre pakiety).

4. Odmiana musi być udostępniana przez dany krój pisma i dostępna w systemie LaTeX.

5. Jest też `\textmd{}` zdejmujący serię fonu (pogrubienie) dla napisu podanego w argumentcie.

6. Jest też `\textup{}` zdejmujący kształt fonu (kursywę, pochylenie i kapitaliki) dla napisu podanego w argumentcie.

`\oldstylenums{tekst}` lub `{\addfontfeatures{Numbers=OldStyle} tekst}`

– odmiany mogą być łączone⁷ np.:

* **pogrubiona kursywa:**

`\textbf{\textit{tekst}}` lub `{\bfseries\itshape tekst}`

* **pogrubiony pochyły**

`\textbf{\textsl{tekst}}` lub `{\bfseries\slshape tekst}`

* **POCHYLONE KAPITALIKI:**

`\textit{\textsc{tekst}}` lub `{\itshape\scshape tekst}`

* Stałej szerokości: **Kursywa z pogrubieniem** oraz **pochylony ...**

- Kolor:

– **poprzez RGB :**

`\textcolor[rgb]{1,.4,0}{tekst}` lub `{\color[rgb]{1,.4,0} tekst}`
lub `{\addfontfeatures{Color=ff6600} tekst}`^{8, 9}

– **poprzez nazwę :**

`\textcolor{red}{tekst}` lub `{\color{red} tekst}`
lub `{\addfontfeatures{Color=red} tekst}`¹⁰

- Efekty:

– pakiet `ulem` – komendy postaci `\xxx{tekst}`, gdzie `\xxx`:

`\sout` ~~`\xout`~~ `\uline` `\uuline` `\uwave` `\dashuline` `\dotuline`

– makra oparte na pakiecie `ulem` zdefiniowane w `pdfArticle`:

* **kolorowe podkreślenie:** `\ul[kolor]{tekst}`

* ~~kolorowe przekreślenie:~~ `\so[kolor]{tekst}`

* **kolorowe podświetlenie:** `\hl[kolor]{tekst}`

– pakiet `contour` – **konturowe tło liter:**

`\contourlength{0.03em}\contour[10]{red}{tekst}`

– pakiet `shadowtext` – **cień liter:**

`\shadowoffset{-1.3pt}\shadowrgb{0.6, 0.6, 0.8}\shadowtext{tekst}`

– pochylenie tekstu:

`{\addfontfeatures{FakeSlant=-0.4} tekst}`

– **rozciągnięcie tekstu:**

`{\addfontfeatures{FakeStretch=1.8} tekst}`

– efekty fontów pdfowych — **(udawane) pogrubienie i konturowe litery**

– przezroczystość: **Najstprawdziwa!!!**

`{\addfontfeatures{Opacity=0.5} tekst}`¹¹

– indeks ^{górny} i _{dolny}:

`{górny \textsubscript{dolny}}`

7. Odpowiednia kombinacja musi być udostępniana przez dany krój pisma i dostępna w systemie LaTeX

8. Kolor ustawiany poprzez `\addfontfeatures` nie zostanie zmieniony poprzez `\textcolor` i `\color`.

9. W `\addfontfeatures` kolor może zostać podany także jako 8 znakowy ciąg kodujący hexalnie wartości RGBA.

10. Dla użycia w `\addfontfeatures` nazwany kolor powinien być definiowany jako RGB, a nie CMYK.

11. Ten sam efekt można uzyskać także przez podanie koloru jako RGBA, z odpowiednią wartością składowej A.

Ligatures

font załadowany z Ligatures=TeX
« « « abc » » » ‘ ‘ ‘ abc ’ ’ ’
„ „ „ abc “ “ “ “ abc ” ” ” ” ”
-- abc — — abc - (U+002D) - (U+2010) — (U+2212)

font załadowany bez Ligatures=TeX
« « << abc » » >> ‘ ‘ ` abc ’ ’ ’
„ „ „ abc “ “ ` “ “ abc ” ” ’ ’ ’
-- abc — --- abc - (U+002D) - (U+2010) — (U+2212)

Verbatim

L^AT_EX umożliwia wprowadzenie tekstu, którego nie będzie interpretował a jedynie przedrukuje go (np. listing kodu źródłowego). Domyślnie tekst taki wypisywany jest fontem o stałej szerokości. Jednak jest to co innego niż `\texttt{}`, w którym wszystkie polecenia są normalnie interpretowane. Zaawansowaną wersję komendy i środowiska pozwalającego na takie wprowadzenie tekstu oferuje pakiet [fancyvrb](#) oraz rozszerzający go m.in. o łamanie linii pakiet [fvextra](#).

```
1 % ustawiamy łamanie linii w środowiskach Verbatim i komendzie \Verb
2 \fvset{
3     breaklines=true, breakafter={/}, breakaftersymbolpre={},
4     breakaftersymbolpost={\tiny\ensuremath{\hookrightarrow} }
5 }
6 Lorem ipsum dolor sit amet, consectetur adipiscing elit, \Verb$/usr/share/texmf/fonts/open__type/public/
7 lm/lmroman10-regular$ sed do eiusmod tempor incididunt ut labore et ...
8 \begin{Verbatim}
9 Lorem ipsum dolor sit amet, $consectetur adipiscing elit$, sed do eiusmod tempor incididunt ut labore et
10 dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ...
11 \end{Verbatim}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, /usr/share/texmf/fonts/open__type/
↪ public/lm/lmroman10-regular sed do eiusmod tempor incididunt ut labore et ...

Lorem ipsum dolor sit amet, \$consectetur adipiscing elit\$, sed do eiusmod tempor
↪ incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis
↪ nostrud exercitation ullamco laboris nisi ...

Więcej informacji w dokumentacji klas [fancyvrb](#) i [fvextra](#) używanych przez [pdfArticle](#).

kolorowanie kodu

Najbardziej popularnymi pakietami umożliwiającymi kolorowanie kodu źródłowego są [minted](#) i [listings](#).

Pakiet [minted](#) korzysta z zewnętrznego skryptu do generowania „kolorwanego” kodu, w związku z czym wymaga uruchamiania `lualatex` (lub innych kompilatorów T_EXa) z opcją `-shell-escape`. Posiada on też pewne ograniczenia co do możliwości włączenia interpretacji kodu T_EXowego w przetwarzanym kodzie źródłowym (nie jest to możliwe np. wewnątrz napisów).

Pakiet [listings](#) nie wymaga zewnętrznych narzędzi i nie ma takich ograniczeń co do włączenia interpretacji kodu T_EXowego, jednak nie wspiera UTF-8 oraz ma problemy z kolorowaniem niektórych elementów składniowych (np. `#$` w bashu jest nazwą zmiennej i nie rozpoczyna komentarza).

Klasa [pdfArticle](#) z opcją `extra` włącza pakiet [minted](#) pozwalając na kolorowanie kodu z użyciem środowiska `minted`:

```
1 \begin{minted}{python}
2 for i in ll:
3     print(i)
4 \end{minted}
```

```
for i in ll:
    print(i)
```

