XSL, tj. XSLT i XSL-FO
czyli
jak przekształcać i ładnie wyświetlać XML-e

Kuba Pochrybniak

1. XML → PDF

Jak?

Jak?

- LaTeX

Jak?

- LaTeX
- XSL-FO

Jak?

- LaTeX
- XSL-FO
- rzeźbienie ręczne (np. w php)

Jak?

- LaTeX
- XSL-FO
- rzeźbienie ręczne (np. w php)

LaTeX

Jak?

- LaTeX
- XSL-FO
- rzeźbienie ręczne (np. w php)

LaTeX

- ładny, potężny, ale…

Jak?

- LaTeX
- XSL-FO
- rzeźbienie ręczne (np. w php)

LaTeX

- ładny, potężny, ale…
- kobylasty

Jak?

- LaTeX
- XSL-FO
- rzeźbienie ręczne (np. w php)

LaTeX

- ładny, potężny, ale…
- kobylasty
- niekonsekwentny

Jak?

- LaTeX
- XSL-FO
- rzeźbienie ręczne (np. w php)

LaTeX

- ładny, potężny, ale…
- kobylasty
- niekonsekwentny
- chwilami trudny

Jak?

- LaTeX
- XSL-FO
- rzeźbienie ręczne (np. w php)

LaTeX

- ładny, potężny, ale…
- kobylasty
- niekonsekwentny
- chwilami trudny
- tabele — tragedia

Jak?

- LaTeX
- XSL-FO
- rzeźbienie ręczne (np. w php)

LaTeX

- ładny, potężny, ale…
- kobylasty
- niekonsekwentny
- chwilami trudny
- tabele — tragedia
- czasem irytujący

Jak?

- LaTeX
- XSL-FO
- rzeźbienie ręczne (np. w php)

LaTeX

- ładny, potężny, ale…
- kobylasty
- niekonsekwentny
- chwilami trudny
- tabele — tragedia
- czasem irytujący

XSL-FO

Jak?

- LaTeX
- XSL-FO
- rzeźbienie ręczne (np. w php)

LaTeX

- ładny, potężny, ale…
- kobylasty
- niekonsekwentny
- chwilami trudny
- tabele — tragedia
- czasem irytujący

XSL-FO

- prosty (w miarę)

Jak?

- LaTeX
- XSL-FO
- rzeźbienie ręczne (np. w php)

LaTeX

- ładny, potężny, ale…
- kobylasty
- niekonsekwentny
- chwilami trudny
- tabele — tragedia
- czasem irytujący

XSL-FO

- prosty (w miarę)
- zgodny z XML

Jak?

- LaTeX
- XSL-FO
- rzeźbienie ręczne (np. w php)

LaTeX

- ładny, potężny, ale…
- kobylasty
- niekonsekwentny
- chwilami trudny
- tabele — tragedia
- czasem irytujący

XSL-FO

- prosty (w miarę)
- zgodny z XML
- w miarę „małe” procesory

Jak?

- LaTeX
- XSL-FO
- rzeźbienie ręczne (np. w php)

LaTeX

- ładny, potężny, ale…
- kobylasty
- niekonsekwentny
- chwilami trudny
- tabele — tragedia
- czasem irytujący

XSL-FO

- prosty (w miarę)
- zgodny z XML
- w miarę „małe” procesory
- fajne drobiazgi (np. „przelewanie”)

Procesory XSL-FO (PDF)

FOP

XEP

Procesory XSL-FO (PDF)

FOP

- darmowy

XEP

- też, ale niekomercyjnie

Procesory XSL-FO (PDF)

FOP

- darmowy
- łatwy do ściągnięcia

XEP

- też, ale niekomercyjnie
- trudny do ściągnięcia

Procesory XSL-FO (PDF)

FOP

- darmowy
- łatwy do ściągnięcia
- trudny w instalacji

XEP

- też, ale niekomercyjnie
- trudny do ściągnięcia
- łatwy w instalacji

Procesory XSL-FO (PDF)

FOP

- darmowy
- łatwy do ściągnięcia
- trudny w instalacji
- mocno niekompletny

XEP

- też, ale niekomercyjnie
- trudny do ściągnięcia
- łatwy w instalacji
- raczej kompletny

Procesory XSL-FO (PDF)

FOP

- darmowy
- łatwy do ściągnięcia
- trudny w instalacji
- mocno niekompletny
- wkurzający

XEP

- też, ale niekomercyjnie
- trudny do ściągnięcia
- łatwy w instalacji
- raczej kompletny
- ?

Procesory XSL-FO (PDF)

FOP

- darmowy
- łatwy do ściągnięcia
- trudny w instalacji
- mocno niekompletny
- wkurzający

XEP

- też, ale niekomercyjnie
- trudny do ściągnięcia
- łatwy w instalacji
- raczej kompletny
- ?

Struktura prostego arkusza

```
⟨?xml version="1.0" encoding="iso-8859-2"?⟩
⟨fo:root version="1.0" xmlns:fo="http://www.w3.org/1999/XSL/Format"⟩
```

Struktura prostego arkusza

```
⟨?xml version="1.0" encoding="iso-8859-2"?⟩
⟨fo:root version="1.0" xmlns:fo="http://www.w3.org/1999/XSL/Format"⟩

   ⟨fo:layout-master-set⟩
      ⟨fo:simple-page-master master-reference="jakies-A4"
            page-width="21cm" page-height="29.7cm"⟩
         ⟨fo:region-body margin="10mm"/⟩
         ...
      ⟨/fo:simple-page-master⟩
   ⟨/fo:layout-master-set⟩
```

Struktura prostego arkusza

```
⟨?xml version="1.0" encoding="iso-8859-2"?⟩
⟨fo:root version="1.0" xmlns:fo="http://www.w3.org/1999/XSL/Format"⟩

   ⟨fo:layout-master-set⟩
      ⟨fo:simple-page-master master-reference="jakies-A4"
            page-width="21cm" page-height="29.7cm"⟩
         ⟨fo:region-body margin="10mm"/⟩
         ...
      ⟨/fo:simple-page-master⟩
   ⟨/fo:layout-master-set⟩

   ⟨fo:page-sequence master-reference="jakies-A4"⟩
      ⟨fo:flow flow-name="xsl-region-body"⟩
         ⟨fo:block⟩Elo ziom!⟨/fo:block⟩
         ⟨fo:block⟩Elo ⟨fo:inline font-style="italic"⟩ziom⟨/fo:inline⟩!⟨/fo:block⟩
         ...
      ⟨/fo:flow⟩
   ⟨/fo:page-sequence⟩

   ...

⟨/fo:root⟩
```

Niektóre elementy

Niektóre elementy

- `page-sequence` — ciąg stron (ustalony layout)

Niektóre elementy

- `page-sequence` — ciąg stron (ustalony layout)
- `flow` — zwykły tekst („przepływający”)

Niektóre elementy

- `page-sequence` — ciąg stron (ustalony layout)
- `flow` — zwykły tekst („przepływający”)
- `static-content` — tekst „stały” (paginy etc.)

Niektóre elementy

- `page-sequence` — ciąg stron (ustalony layout)
- `flow` — zwykły tekst („przepływający”)
- `static-content` — tekst „stały” (paginy etc.)
- `block` — blok „pionowy” (akapit)

Niektóre elementy

- `page-sequence` — ciąg stron (ustalony layout)
- `flow` — zwykły tekst („przepływający”)
- `static-content` — tekst „stały” (paginy etc.)
- `block` — blok „pionowy” (akapit)
- `table-and-caption` — tabela z podpisem

Niektóre elementy

- `page-sequence` — ciąg stron (ustalony layout)
- `flow` — zwykły tekst („przepływający”)
- `static-content` — tekst „stały” (paginy etc.)
- `block` — blok „pionowy” (akapit)
- `table-and-caption` — tabela z podpisem
- `list-block` — lista

Niektóre elementy

- `page-sequence` — ciąg stron (ustalony layout)
- `flow` — zwykły tekst („przepływający”)
- `static-content` — tekst „stały” (paginy etc.)
- `block` — blok „pionowy” (akapit)
- `table-and-caption` — tabela z podpisem
- `list-block` — lista
- `inline` — blok „poziomy” (fragment tekstu)

Niektóre elementy

- `page-sequence` — ciąg stron (ustalony layout)
- `flow` — zwykły tekst („przepływający”)
- `static-content` — tekst „stały” (paginy etc.)
- `block` — blok „pionowy” (akapit)
- `table-and-caption` — tabela z podpisem
- `list-block` — lista
- `inline` — blok „poziomy” (fragment tekstu)
- `...`

Przykładowe atrybuty

Przykładowe atrybuty

- `font-family`

Przykładowe atrybuty

- `font-family`
- `font-size`

Przykładowe atrybuty

- `font-family`
- `font-size`
- `border-right`

Przykładowe atrybuty

- `font-family`
- `font-size`
- `border-right`
- `margin-bottom`

Przykładowe atrybuty

- `font-family`
- `font-size`
- `border-right`
- `margin-bottom`
- `baseline-shift`

Przykładowe atrybuty

- `font-family`
- `font-size`
- `border-right`
- `margin-bottom`
- `baseline-shift`
- `last-line-end-indent`

Przykładowe atrybuty

- `font-family`
- `font-size`
- `border-right`
- `margin-bottom`
- `baseline-shift`
- `last-line-end-indent`
- `...`

Co jest fajne (w standardzie)

Co jest fajne (w standardzie)

- sensowna, ładnie zorganizowana struktura

Co jest fajne (w standardzie)

- sensowna, ładnie zorganizowana struktura
- tekst wielokolumnowy

Co jest fajne (w standardzie)

- sensowna, ładnie zorganizowana struktura
- tekst wielokolumnowy
- „przelewanie” tekstu

Co jest fajne (w standardzie)

- sensowna, ładnie zorganizowana struktura
- tekst wielokolumnowy
- „przelewanie” tekstu
- typowe: wstawianie grafiki etc.

Co jest fajne (w standardzie)

- sensowna, ładnie zorganizowana struktura
- tekst wielokolumnowy
- „przelewanie” tekstu
- typowe: wstawianie grafiki etc.
- obracanie tekstu

Co jest fajne (w standardzie)

- sensowna, ładnie zorganizowana struktura
- tekst wielokolumnowy
- „przelewanie” tekstu
- typowe: wstawianie grafiki etc.
- obracanie tekstu
- odnośniki (również typu „pageref”)

Co jest fajne (w standardzie)

- sensowna, ładnie zorganizowana struktura
- tekst wielokolumnowy
- „przelewanie” tekstu
- typowe: wstawianie grafiki etc.
- obracanie tekstu
- odnośniki (również typu „pageref”)
- duuuża kontrola nad wielkościami

Co jest fajne (w standardzie)

- sensowna, ładnie zorganizowana struktura
- tekst wielokolumnowy
- „przelewanie” tekstu
- typowe: wstawianie grafiki etc.
- obracanie tekstu
- odnośniki (również typu „pageref”)
- duuuża kontrola nad wielkościami
- metody robienia layoutu (również złożonego)

Co jest fajne (w standardzie)

- sensowna, ładnie zorganizowana struktura
- tekst wielokolumnowy
- „przelewanie” tekstu
- typowe: wstawianie grafiki etc.
- obracanie tekstu
- odnośniki (również typu „pageref”)
- duuuża kontrola nad wielkościami
- metody robienia layoutu (również złożonego)
- ogólna konsekwencja i przejrzystość (?)

Co jest fajne (w standardzie)

- sensowna, ładnie zorganizowana struktura
- tekst wielokolumnowy
- „przelewanie” tekstu
- typowe: wstawianie grafiki etc.
- obracanie tekstu
- odnośniki (również typu „pageref”)
- duuuża kontrola nad wielkościami
- metody robienia layoutu (również złożonego)
- ogólna konsekwencja i przejrzystość (?)
- można tym sensownie składać książki

Co jest niefajne

Co jest niefajne

- rozwlekłość zapisu

Co jest niefajne

- rozwlekłość zapisu
- przy wielu regułach trudno o konsekwencję (CSS ma lepiej)

Co jest niefajne

- rozwlekłość zapisu
- przy wielu regułach trudno o konsekwencję (CSS ma lepiej)
- mniejsza niż w LaTeX-u kontrola nad stronami

Co jest niefajne

- rozwlekłość zapisu
- przy wielu regułach trudno o konsekwencję (CSS ma lepiej)
- mniejsza niż w LaTeX-u kontrola nad stronami
- „zgodność” procesorów XSL-FO ze standardem

2. XSLT (XML → cokolwiek)

Do czego?

Do czego?

- wizualizacja

Do czego?

- wizualizacja
  - $\rightarrow$ HTML

Do czego?

- wizualizacja
  - $\rightarrow$ HTML
  - $\rightarrow$ LaTeX $\rightarrow$ PDF

Do czego?

- wizualizacja
  - $\rightarrow$ HTML
  - $\rightarrow$ LaTeX $\rightarrow$ PDF
  - $\rightarrow$ XSL-FO $\rightarrow$ ...

Do czego?

- wizualizacja
  - $\rightarrow$ HTML
  - $\rightarrow$ LaTeX $\rightarrow$ PDF
  - $\rightarrow$ XSL-FO $\rightarrow$ ...
- przerabianie

Do czego?

- wizualizacja
  - $\rightarrow$ HTML
  - $\rightarrow$ LaTeX $\rightarrow$ PDF
  - $\rightarrow$ XSL-FO $\rightarrow$ ...
- przerabianie
  - $\rightarrow$ inserty

Do czego?

- wizualizacja
  - $\rightarrow$ HTML
  - $\rightarrow$ LaTeX $\rightarrow$ PDF
  - $\rightarrow$ XSL-FO $\rightarrow$ ...
- przerabianie
  - $\rightarrow$ inserty
  - $\rightarrow$ Excel

Do czego?

- wizualizacja
  - $\rightarrow$ HTML
  - $\rightarrow$ LaTeX $\rightarrow$ PDF
  - $\rightarrow$ XSL-FO $\rightarrow$ ...
- przerabianie
  - $\rightarrow$ inserty
  - $\rightarrow$ Excel
  - $\rightarrow$ inny XML

Do czego?

- wizualizacja
  - $\rightarrow$ HTML
  - $\rightarrow$ LaTeX $\rightarrow$ PDF
  - $\rightarrow$ XSL-FO $\rightarrow$ ...
- przerabianie
  - $\rightarrow$ inserty
  - $\rightarrow$ Excel
  - $\rightarrow$ inny XML
- walidacja (XMLSchema wymięka)

Czym?

libxslt

Saxon

Czym?

libxslt

- jest linia poleceń

Saxon

- jest linia poleceń

Czym?

libxslt

- jest linia poleceń
- PHP

Saxon

- jest linia poleceń
- Java

Czym?

libxslt

- jest linia poleceń
- PHP
- źle sortuje pl

Saxon

- jest linia poleceń
- Java
- dobrze sortuje pl

Czym?

libxslt

- jest linia poleceń
- PHP
- źle sortuje pl
- wymięka przy dużych

Saxon

- jest linia poleceń
- Java
- dobrze sortuje pl
- radzi sobie z dużymi

Czym?

libxslt

- jest linia poleceń
- PHP
- źle sortuje pl
- wymięka przy dużych
- libexslt

Saxon

- jest linia poleceń
- Java
- dobrze sortuje pl
- radzi sobie z dużymi
- mało rozszerzeń

Czym?

libxslt

- jest linia poleceń
- PHP
- źle sortuje pl
- wymięka przy dużych
- libexslt
- XSLT 1.0

Saxon

- jest linia poleceń
- Java
- dobrze sortuje pl
- radzi sobie z dużymi
- mało rozszerzeń
- XSLT 2.0

Czym?

libxslt

- jest linia poleceń
- PHP
- źle sortuje pl
- wymięka przy dużych
- libexslt
- XSLT 1.0
- na bakier ze standardem

Saxon

- jest linia poleceń
- Java
- dobrze sortuje pl
- radzi sobie z dużymi
- mało rozszerzeń
- XSLT 2.0
- trzyma się standardu

Czym?

libxslt

- jest linia poleceń
- PHP
- źle sortuje pl
- wymięka przy dużych
- libexslt
- XSLT 1.0
- na bakier ze standardem
- nieco wkurzający

Saxon

- jest linia poleceń
- Java
- dobrze sortuje pl
- radzi sobie z dużymi
- mało rozszerzeń
- XSLT 2.0
- trzyma się standardu
- mało wkurzający

Czym?

libxslt

- jest linia poleceń
- PHP
- źle sortuje pl
- wymięka przy dużych
- libexslt
- XSLT 1.0
- na bakier ze standardem
- nieco wkurzający

Saxon

- jest linia poleceń
- Java
- dobrze sortuje pl
- radzi sobie z dużymi
- mało rozszerzeń
- XSLT 2.0
- trzyma się standardu
- mało wkurzający

. . . i parę innych.

Struktura prostego arkusza

```
⟨?xml version="1.0" encoding="iso-8859-2"?⟩
⟨xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"⟩

   ⟨xsl:template match="/"⟩
      Elo ziom!
   ⟨/xsl:template⟩

   ...

⟨/xsl:stylesheet⟩
```

```
⟨xsl:stylesheet ...⟩

   ⟨xsl:template match="osoba"⟩
      ⟨xsl:text⟩&#10;⟨/xsl:text⟩
      ⟨xsl:value-of select="imie"/⟩
      ⟨xsl:text⟩ ⟨/xsl:text⟩
      ⟨xsl:value-of select="nazwisko"/⟩
   ⟨/xsl:template⟩

⟨/xsl:stylesheet⟩
```

```
⟨xsl:stylesheet ...⟩

   ⟨xsl:template match="osoba"⟩
      ⟨xsl:text⟩&#10;⟨/xsl:text⟩
      ⟨xsl:value-of select="imie"/⟩
      ⟨xsl:text⟩ ⟨/xsl:text⟩
      ⟨xsl:value-of select="nazwisko"/⟩
   ⟨/xsl:template⟩

⟨/xsl:stylesheet⟩
```

```
⟨xsl:stylesheet ...⟩

   ⟨xsl:template match="osoba"⟩
      ⟨xsl:text⟩&#10;⟨/xsl:text⟩
      ⟨xsl:for-each select="imie"⟩
         ⟨xsl:value-of select="."/⟩
         ⟨xsl:text⟩ ⟨/xsl:text⟩
      ⟨/xsl:for-each⟩
      ⟨xsl:value-of select="nazwisko"/⟩
   ⟨/xsl:template⟩

⟨/xsl:stylesheet⟩
```

Główne elementy

Główne elementy

- `template[@match]` — samodopasowujące się szablony

Główne elementy

- `template[@match]` — samodopasowujące się szablony

  `⟨xsl:apply-templates select="..."/⟩`

Główne elementy

- `template[@match]` — samodopasowujące się szablony
  `⟨xsl:apply-templates select="..."/⟩`
- `template[@name]` — szablony wywoływane ręcznie

Główne elementy

- `template[@match]` — samodopasowujące się szablony

  `⟨xsl:apply-templates select="..."/⟩`
- `template[@name]` — szablony wywoływane ręcznie

  `⟨xsl:call-template name="..."/⟩`

Główne elementy

- `template[@match]` — samodopasowujące się szablony
  `⟨xsl:apply-templates select="..."/⟩`
- `template[@name]` — szablony wywoływane ręcznie
  `⟨xsl:call-template name="..."/⟩`

A w środku...

Główne elementy

- `template[@match]` — samodopasowujące się szablony
  `⟨xsl:apply-templates select="..."/⟩`
- `template[@name]` — szablony wywoływane ręcznie
  `⟨xsl:call-template name="..."/⟩`

A w środku...

- `text` — albo po prostu goły tekst

Główne elementy

- `template[@match]` — samodopasowujące się szablony
  `⟨xsl:apply-templates select="..."/⟩`
- `template[@name]` — szablony wywoływane ręcznie
  `⟨xsl:call-template name="..."/⟩`

A w środku...

- `text` — albo po prostu goły tekst
- węzły XML-owe

Główne elementy

- `template[@match]` — samodopasowujące się szablony
  `⟨xsl:apply-templates select="..."/⟩`
- `template[@name]` — szablony wywoływane ręcznie
  `⟨xsl:call-template name="..."/⟩`

A w środku...

- `text` — albo po prostu goły tekst
- węzły XML-owe
- `value-of` — jeśli w głąb, lepiej `apply-templates`

Główne elementy

- `template[@match]` — samodopasowujące się szablony
  `⟨xsl:apply-templates select="..."/⟩`
- `template[@name]` — szablony wywoływane ręcznie
  `⟨xsl:call-template name="..."/⟩`

A w środku...

- `text` — albo po prostu goły tekst
- węzły XML-owe
- `value-of` — jeśli w głąb, lepiej `apply-templates`
- `for-each` — jeśli w głąb, lepiej `apply-templates`

Głównych elementów c.d.

Głównych elementów c.d.

- `if` (bez else!)

Głównych elementów c.d.

- `if` (bez else!), `choose–when–otherwise`

Głównych elementów c.d.

- `if` (bez else!), `choose`–`when`–`otherwise`
- `variable`, `param`

Głównych elementów c.d.

- `if` (bez else!), `choose–when–otherwise`
- `variable`, `param`

  ⟨xsl:with-param name="rozmiar-buta"⟩9⟨/xsl:with-param⟩

  ↑ wołanie / odbiór → ⟨xsl:param name="rozmiar-buta"/⟩

  ⟨xsl:value-of select="$rozmiar-buta - 1"/⟩

Głównych elementów c.d.

- `if` (bez else!), `choose–when–otherwise`
- `variable`, `param`

  `⟨xsl:with-param name="rozmiar-buta"⟩9⟨/xsl:with-param⟩`

  ↑ wołanie / odbiór → `⟨xsl:param name="rozmiar-buta"/⟩`

  `⟨xsl:value-of select="$rozmiar-buta - 1"/⟩`
- `sort` — w `apply-templates`/`call-template`

Głównych elementów c.d.

- `if` (bez else!), `choose–when–otherwise`
- `variable`, `param`

  ⟨xsl:with-param name="rozmiar-buta"⟩9⟨/xsl:with-param⟩

  ↑ wołanie / odbiór → ⟨xsl:param name="rozmiar-buta"/⟩

  ⟨xsl:value-of select="$rozmiar-buta - 1"/⟩
- `sort` — w `apply-templates`/`call-template`

  ⟨xsl:sort order="descending" lang="pl"/⟩

Głównych elementów c.d.

- `if` (bez else!), `choose–when–otherwise`
- `variable`, `param`

  ⟨xsl:with-param name="rozmiar-buta"⟩9⟨/xsl:with-param⟩

  ↑ wołanie / odbiór → ⟨xsl:param name="rozmiar-buta"/⟩

  ⟨xsl:value-of select="$rozmiar-buta - 1"/⟩
- `sort` — w `apply-templates`/`call-template`

  ⟨xsl:sort order="descending" lang="pl"/⟩
- `output` — xml, html, text

Głównych elementów c.d.

- `if` (bez else!), `choose–when–otherwise`
- `variable`, `param`

  ⟨xsl:with-param name="rozmiar-buta"⟩9⟨/xsl:with-param⟩

  ↑ wołanie / odbiór → ⟨xsl:param name="rozmiar-buta"/⟩

  ⟨xsl:value-of select="$rozmiar-buta - 1"/⟩
- `sort` — w `apply-templates`/`call-template`

  ⟨xsl:sort order="descending" lang="pl"/⟩
- `output` — xml, html, text

- rozwijanie — w atrybutach elementów XSL zazwyczaj tak, w innych ręcznie:

  ⟨osobnik liczba-dzieci="{count(current()/dziecko)}"/⟩

Co wkurza

Co wkurza

- objętość kodu

Co wkurza

- objętość kodu
- eskejpowanie/brak

Co wkurza

- objętość kodu
- eskejpowanie/brak
- prymitywne funkcje na stringach

Co wkurza

- objętość kodu
- eskejpowanie/brak
- prymitywne funkcje na stringach
- łatwo się „spłaszcza”, ale „pogłębia” trudno

Co wkurza

- objętość kodu
- eskejpowanie/brak
- prymitywne funkcje na stringach
- łatwo się „spłaszcza”, ale „pogłębia” trudno
- trudne (bądź niemożliwe) domknięcie przechodnie

Co wkurza

- objętość kodu
- eskejpowanie/brak
- prymitywne funkcje na stringach
- łatwo się „spłaszcza”, ale „pogłębia” trudno
- trudne (bądź niemożliwe) domknięcie przechodnie
- brak pętli po liczbach

Co wkurza

- objętość kodu
- eskejpowanie/brak
- prymitywne funkcje na stringach
- łatwo się „spłaszcza”, ale „pogłębia” trudno
- trudne (bądź niemożliwe) domknięcie przechodnie
- brak pętli po liczbach
- brak „prawdziwych” zmiennych

Co wkurza

- objętość kodu
- eskejpowanie/brak
- prymitywne funkcje na stringach
- łatwo się „spłaszcza”, ale „pogłębia” trudno
- trudne (bądź niemożliwe) domknięcie przechodnie
- brak pętli po liczbach
- brak „prawdziwych” zmiennych
- brak obliczeń zmiennoprzecinkowych

Co jest fajne

Co jest fajne

- XPath — intuicyjność

Co jest fajne

- XPath — intuicyjność
- trudno coś zepsuć

Co jest fajne

- XPath — intuicyjność
- trudno coś zepsuć
- świetne automatyczne priorytety

Co jest fajne

- XPath — intuicyjność
- trudno coś zepsuć
- świetne automatyczne priorytety
- korzystanie z zewnętrznych funkcji (trudne)

Co jest fajne

- XPath — intuicyjność
- trudno coś zepsuć
- świetne automatyczne priorytety
- korzystanie z zewnętrznych funkcji (trudne)
- zazwyczaj wiadomo, co i jak robić

Arkusz zmieniający wybrany węzeł, resztę pozostawiający bez zmian

```
⟨xsl:stylesheet ...⟩

   ⟨xsl:template match="@*|node()"⟩
      ⟨xsl:copy⟩
         ⟨xsl:apply-templates select="@*|node()"/⟩
      ⟨/xsl:copy⟩
   ⟨/xsl:template⟩
```

Arkusz zmieniający wybrany węzeł, resztę pozostawiający bez zmian

```
⟨xsl:stylesheet ...⟩

   ⟨xsl:template match="@*|node()"⟩
      ⟨xsl:copy⟩
         ⟨xsl:apply-templates select="@*|node()"/⟩
      ⟨/xsl:copy⟩
   ⟨/xsl:template⟩

   ⟨xsl:template match="//rozdzial[5]/akapit[contains(., 'lub czasopisma')]"/⟩

⟨/xsl:stylesheet⟩
```

3. Ćwiczenia? Do domu / na zupkę?