

Prezentacja dokumentów XML

Patryk Czarnik

Instytut Informatyki UW

XML i nowoczesne technologie zarządzania treścią –
2007/08

Arkusze stylu

Rozdzielenie treści od wyglądu
Przypisanie stylu do dokumentu

CSS

Idea i zastosowania
Struktura arkusza
Wizualizacja

XSL

Wizualizacja dzięki przekształceniu
XSL-FO

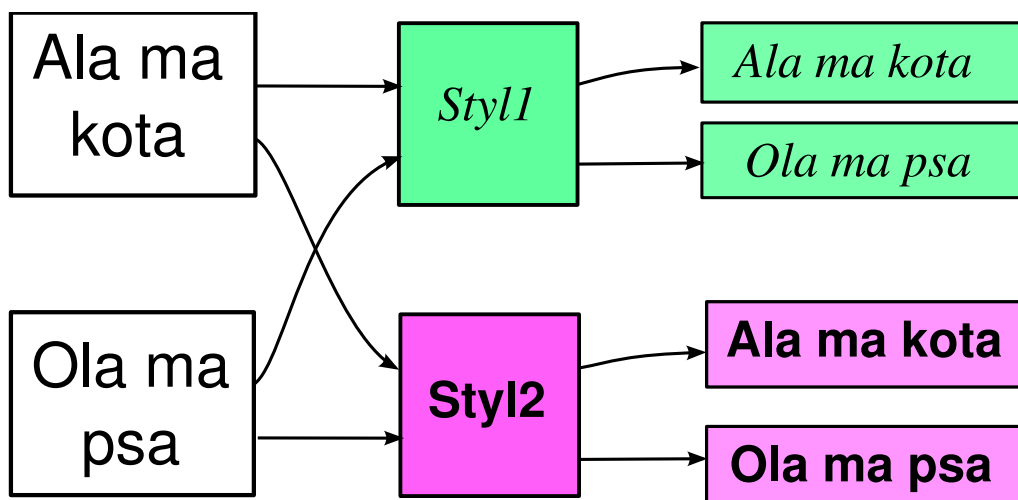
HTML

Idea i historia
XHTML

Rozdzielenie treści od wyglądu

- ▶ Dokumenty źródłowe:
 - ▶ tylko treść (dane),
 - ▶ znaczniki dla struktury, semantyki, nie wyglądu.
- ▶ Wygląd:
 - ▶ interpretacja znanych typów dokumentów,
 - ▶ zewnętrzne arkusze stylu (dowolne typy dokumentów).

Idea arkusza stylu



Zalety rozdzielenia treści od wyglądu

- ▶ Łatwiejsza analiza danych źródłowych.
- ▶ Możliwość prezentacji tych samych danych źródłowych na wiele sposobów.
- ▶ Zmiana wyglądu dokonywana poza dokumentem źródłowym, raz dla całej klasy dokumentów.

Przypisanie stylu do dokumentu

- ▶ Za pomocą instrukcji przetwarzania `xml-stylesheet`.
- ▶ Opisane w rekomendacji W3C *Associating Style Sheets with XML documents*.

Przykład z rekomendacji

```
<?xml-stylesheet href="common.css" type="text/css"?>  
  
<?xml-stylesheet alternate="yes" title="compact"  
    href="small-base.css" type="text/css"?>  
  
<?xml-stylesheet alternate="yes" title="compact"  
    href="small-extras.css" type="text/css"?>  
  
<?xml-stylesheet alternate="yes" title="big print"  
    href="bigprint.css" type="text/css"?>
```

Cascading Style Sheets

- ▶ Początki idei arkuszy stylu: lata 1970.
- ▶ Początki CSS: 1994.
- ▶ Rekomendacja CSS Level 1: grudzień 1996.
- ▶ Rekomendacja CSS Level 2: maj 1998
 - ▶ mniej więcej zaimplementowana we współczesnych przeglądarkach internetowych.
- ▶ CSS Level 3: ciągle rozwijane
 - ▶ modularyzacja (niektóre moduły już zatwierdzone),
 - ▶ nowe możliwości.
- ▶ CSS 2.1: rekomendacja kandydująca
 - ▶ ograniczenie i zmiany specyfikacji CSS 2,
 - ▶ dostosowanie specyfikacji do powszechnie stosowanych praktyk,
 - ▶ rezygnacja z rzeczy, które się nie sprawdziły.

Zastosowania CSS

- ▶ Pierwsze i główne zastosowanie: styl dla stron WWW.
- ▶ Rozdzielenie treści od prezentacji dla HTML.
- ▶ Od CSS 2 bardzo ważne:
 - ▶ wsparcie dla alternatywnych metod prezentacji treści (np. czytanie na głos, tabulatory Braila),
 - ▶ umożliwienie czytelnikom podania własnego stylu (np. większej czcionki i innych kolorów dla słabo widzących).

Przykład arkusza

```
pracownik {
  display: block;
  margin: 3pt 6pt;
  font-family: 'Times New Roman', 'Nimbus Roman', serif;
  text-align: left;
}

imię, nazwisko {
  display: inline;
  color: #004488;
}

pracownik[stopień='kierownik'] nazwisko {
  font-weight: bold;
}
```

Selektory

- ▶ `imię` – nazwa elementu,
- ▶ `imię, nazwisko` – oba elementy,
- ▶ `pracownik imię` – potomek,
- ▶ `pracownik > imię` – dziecko,
- ▶ `A + B` – następnik,
- ▶ `imię:first-child` – pierwsze dziecko,
- ▶ `pracownik[stopień]` – test istnienia atrybutu,
- ▶ `pracownik[stopień='kierownik']` – test wartości atrybutu,
- ▶ `pracownik[stopnie~='kierownik']` – wartość występuje na liście,
- ▶ `pracownik#k12` – wartość atrybutu zadeklarowanego jako ID,
- ▶ `ol.pracownicy` – równoważne `ol[class~='pracownicy']` (tylko HTML).

Media types

Styl zależny od rodzaju medium.

Przykład

```
@media print {  
  body {  
    margin: 15mm;  
  }  
}  
  
@media screen {  
  body {  
    margin: 0;  
  }  
}
```

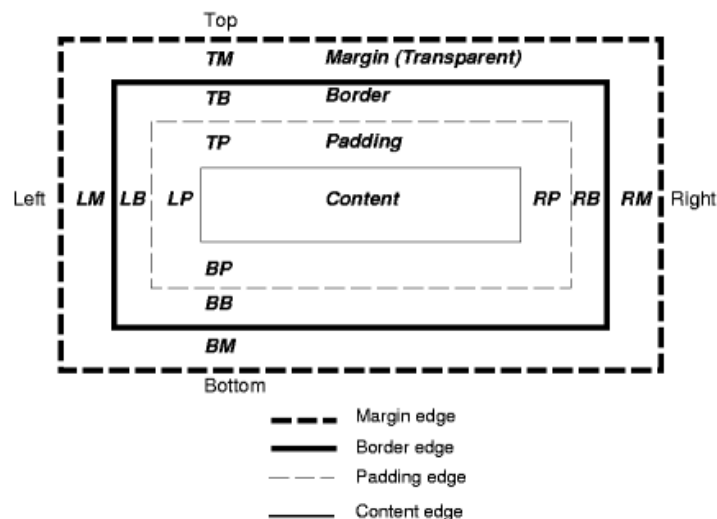
Własność `display`

- ▶ Rodzaj obiektu wizualnego reprezentującego element.
- ▶ **Możliwe wartości:** `inline`, `block`, `list-item`, `run-in`, `inline-block`, `table`, `table-cell`, ..., `none`,
- ▶ Podstawowa własność w przypadku wizualizacji XML.
- ▶ Nie stosowane zwykle dla HTML (poza `none`).

Pudełka i wyrównanie

- ▶ Zagnieżdżenie bloków odpowiada zagnieżdżeniu elementów w dokumencie.
- ▶ Możliwe ręczne pozycjonowanie.
- ▶ `margin`, `padding` – margines zewnętrzny i wewnętrzny,
- ▶ `border-style`, `border-color`, `border-width` – obramowanie,
- ▶ `position` (`static`, `relative`, `absolute`, `fixed`) – sposób pozycjonowania,
- ▶ `left`, `right`, `top`, `bottom` – pozycja,
- ▶ `width`, `height`, `min-width`, `max-height`, ... – rozmiar.

Pudełka



Tekst i czcionka

- ▶ Zagnieżdżenie bloków odpowiada zagnieżdżeniu elementów w dokumencie.
- ▶ Możliwe ręczne pozycjonowanie.
- ▶ `color`, `background-color`, `background-image` – kolor i tło,
- ▶ `font-family` – nazwa czcionki oraz `serif`, `sans-serif`, `monospace` ...
- ▶ `font-size`,
- ▶ `font-style`, `font-weight`
- ▶ `text-decoration`,
- ▶ `text-align`.

Wizualizacja dzięki przekształceniu

- ▶ Format A – czyste dane.
- ▶ Format B – wiemy jak prezentować.
- ▶ Sposób prezentacji A = sposób przekształcenia A do B.
- ▶ Prezentowalne formaty XML:
 - ▶ XHTML,
 - ▶ XSL Formatting Objects.

Extensible Stylesheet Language (XSL)

- ▶ Zalecany sposób prezentacji XML.
- ▶ Zdefiniowany w rekomendacjach (wersje 1.0 w 1999 i 2001):
 - ▶ XSL (ogólne ramy, język XSL-FO),
 - ▶ XSLT (przekształcenia dokumentów XML, niekoniecznie w celu wizualizacji),
 - ▶ XPath – język wyrażeń zawierający adresowanie fragmentów dokumentu.
- ▶ Arkusz stylu mówi jak przekształcać dany typ dokumentu do dokumentu XSL-FO.
- ▶ W praktyce przekształcenia także do innych formatów, często (X)HTML.

XSL Formatting Objects

- ▶ Zastosowanie XML służące do prezentacji.
- ▶ Używany w profesjonalnych zastosowaniach publikacyjnych.
- ▶ Nie pisze się dokumentów w XSL-FO, tylko przekształcenia do XSL-FO.

Przykład dokumentu XSL-FO

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="moja-strona">
      <fo:region-body />
    </fo:simple-page-master>
  </fo:layout-master-set>

  <fo:page-sequence master-reference="moja-strona">
    <fo:flow flow-name="xsl-region-body">
      <fo:block>Hello World!</fo:block>
    </fo:flow>
  </fo:page-sequence>

</fo:root>
```

Hierarchia „pudełek”

- ▶ strona
- ▶ obszar (*region*) – pięć predefiniowanych:
 - ▶ *region-body* – główna zawartość strony,
 - ▶ *region-before* – nad,
 - ▶ *region-after* – pod,
 - ▶ *region-start* – po lewej,
 - ▶ *region-end* – po prawej.
- ▶ blok
- ▶ wiersz (*line*)
- ▶ wewnątrz wiersza (*inline*)

Szablon strony (*page-master*)

- ▶ Określają układ pojedynczej strony.
- ▶ Dokument może zostać podzielony na wiele takich stron (gdy zawartość się nie mieści)

Przykład z W3 Schools

```
<fo:simple-page-master master-name="A4"
  page-width="297mm" page-height="210mm"
  margin-top="1cm"   margin-bottom="1cm"
  margin-left="1cm"  margin-right="1cm">
  <fo:region-body    margin="3cm"/>
  <fo:region-before  extent="2cm"/>
  <fo:region-after   extent="2cm"/>
  <fo:region-start   extent="2cm"/>
  <fo:region-end     extent="2cm"/>
</fo:simple-page-master>
```

Zawartość stron

- ▶ `page-sequence` – pewna liczba stron dokumentu.
- ▶ Zawartość elementu `flow` przepływa na kolejne strony.
- ▶ Atrybut `flow-name` – do którego obszaru strony ma trafić zawartość.

Przykład z W3 Schools

```
<fo:page-sequence master-reference="A4">
  <fo:flow flow-name="xsl-region-body">
    <fo:block>Hello W3Schools</fo:block>
  </fo:flow>
</fo:page-sequence>
```

Wygląd fragmentów dokumentu

- ▶ Wygląd i własności bloków (różnych poziomów) określone w atrybutach:
 - ▶ margin, padding, border-style ...
 - ▶ background-color, background-image ...
 - ▶ font-family, font-weight, font-style, font-size ...
 - ▶ text-align, text-align-last, text-indent, start-indent, end-indent, wrap-option, break-before ...
- ▶ Bardzo podobne do własności CSS.

Listy

Przykład z W3 Schools

```
<fo:list-block>
  <fo:list-item>
    <fo:list-item-label>
      <fo:block>*</fo:block>
    </fo:list-item-label>
    <fo:list-item-body>
      <fo:block>Volvo</fo:block>
    </fo:list-item-body>
  </fo:list-item>

  <fo:list-item>
    <fo:list-item-label>
      <fo:block>*</fo:block>
    </fo:list-item-label>
    <fo:list-item-body>
      <fo:block>Saab</fo:block>
    </fo:list-item-body>
  </fo:list-item>
</fo:list-block>
```

Tabele – Przykład z W3 Schools

```
<fo:table-and-caption>
  <fo:table>
    <fo:table-column column-width="25mm"/>
    <fo:table-column column-width="25mm"/>

    <fo:table-header>
      <fo:table-row>
        <fo:table-cell>
          <fo:block font-weight="bold">Car</fo:block>
        </fo:table-cell>
        <fo:table-cell> <!-- ... --> </fo:table-cell>
      </fo:table-row>
    </fo:table-header>

    <fo:table-body>
      <fo:table-row> <!-- ... --> </fo:table-row>
      <fo:table-row> <!-- ... --> </fo:table-row>
    </fo:table-body>
  </fo:table>
</fo:table-and-caption>
```

Historia HTML

- ▶ Początki HTML:
 - ▶ Tim Berners-Lee, CERN, początek lat 1980-tych,
 - ▶ ENQUIRE – język dla dokumentów technicznych.
- ▶ Od 1989 – zastosowanie HTML w Internecie.
- ▶ 1993 – HTML zdefiniowany jako zastosowanie SGML.
- ▶ HTML 4 – 1999:
 - ▶ ograniczony zestaw znaczników,
 - ▶ nacisk na strukturę i semantykę,
 - ▶ CSS zamiast znaczników i atrybutów formatujących.

Znaczniki HTML

- ▶ **Strukturalne:**
 - ▶ h1 – h6, p,
 - ▶ ol, ul, li, dl, dt, dd,
 - ▶ table, tr, td, th ...
- ▶ **Semantyczne:**
 - ▶ q, dfn, code ...
- ▶ **Formatujące:**
 - ▶ b, i, font, center

XHTML – główne idee

- ▶ XML zamiast SGML (łatwiejsze parsowanie, mniej wygodne pisanie).
- ▶ Modularyzacja (podział znaczników na grupy, stosowane niekoniecznie wszystkie na raz).
- ▶ Wykorzystanie przestrzeni nazw:
 - ▶ wykorzystanie istniejących standardów (XForms, SVG, MathML, ...),
 - ▶ dodawanie własnych znaczników.

Status HTML

- ▶ Obecnie zalecane:
 - ▶ HTML 4.01,
 - ▶ XHTML 1.0.
 - ▶ XHTML 1.1 (rekomendacja, podobno problemy z serwerami – niezgodny typ MIME).
- ▶ Obecnie opracowywane:
 - ▶ HTML 5 – nowe zastosowania (`video`, `canvas`), wzbogacenie struktury (`article`, `aside`),
 - ▶ XHTML 2.0 – duża zmiana zestawu znaczników, odwołania do zewnętrznych standardów (np. XForms zamiast HTML Forms).