Narzędzia informatyczne w językoznawstwie

Perl - Wprowadzenie do XML

Marcin Junczys-Dowmunt
junczys@amu.edu.pl

Zakład Logiki Stosowanej
http://www.logic.amu.edu.pl

16. kwietnia 2008

Plan dzisiejszego wykładu

- Definicja XML
- Zalety i wady XML
- Składnia XML
- Krótka powtórka z XHTML
- Walidacja XML, czyli DTD
- Przegląd XML-Schema, XPath, XSL
- Ogólne standardy XML
- Standardy XML w językoznawstwie - Osobny wykład

XML - Definicja

- XML (Extensible Markup Language - Rozszerzalny język znaczników) jest uniwersalnym językiem formalnym przeznaczonym do tworzenia dedykowanych języków znaczników.

XML - Definicja

- XML (Extensible Markup Language - Rozszerzalny język znaczników) jest uniwersalnym językiem formalnym przeznaczonym do tworzenia dedykowanych języków znaczników.
- Pozwala użytkownikom na tworzenie własnych elementów.

XML - Definicja

- XML (Extensible Markup Language - Rozszerzalny język znaczników) jest uniwersalnym językiem formalnym przeznaczonym do tworzenia dedykowanych języków znaczników.
- Pozwala użytkownikom na tworzenie własnych elementów.
- Umożliwia współdzielenie danych ustrukturyzowanych przez różne systemy informatyczne

XML - Definicja

- XML (Extensible Markup Language - Rozszerzalny język znaczników) jest uniwersalnym językiem formalnym przeznaczonym do tworzenia dedykowanych języków znaczników.
- Pozwala użytkownikom na tworzenie własnych elementów.
- Umożliwia współdzielenie danych ustrukturyzowanych przez różne systemy informatyczne
- Wykorzystywany do zapisu dokumentów i do serializacji danych (zapis złożonych struktur danych w postaci linearnej)

XML - Definicja

- XML (Extensible Markup Language - Rozszerzalny język znaczników) jest uniwersalnym językiem formalnym przeznaczonym do tworzenia dedykowanych języków znaczników.
- Pozwala użytkownikom na tworzenie własnych elementów.
- Umożliwia współdzielenie danych ustrukturyzowanych przez różne systemy informatyczne
- Wykorzystywany do zapisu dokumentów i do serializacji danych (zapis złożonych struktur danych w postaci linearnej)
- Czytelny podzbiór języka SGML (Standard Generalized Markup Language)

XML - Definicja

- XML (Extensible Markup Language - Rozszerzalny język znaczników) jest uniwersalnym językiem formalnym przeznaczonym do tworzenia dedykowanych języków znaczników.
- Pozwala użytkownikom na tworzenie własnych elementów.
- Umożliwia współdzielenie danych ustrukturyzowanych przez różne systemy informatyczne
- Wykorzystywany do zapisu dokumentów i do serializacji danych (zapis złożonych struktur danych w postaci linearnej)
- Czytelny podzbiór języka SGML (Standard Generalized Markup Language)
- Tworzony przez World Wide Web Consortium

Zalety XML

- Format plików XML to zwykły tekst

Zalety XML

- Format plików XML to zwykły tekst
- Korzysta standardowo z Unicode, najczęściej z kodowania UTF-8

Zalety XML

- Format plików XML to zwykły tekst
- Korzysta standardowo z Unicode, najczęściej z kodowania UTF-8
- Prosta reprezentacja podstawowych struktur danych: rekordów, list i drzew

Zalety XML

- Format plików XML to zwykły tekst
- Korzysta standardowo z Unicode, najczęściej z kodowania UTF-8
- Prosta reprezentacja podstawowych struktur danych: rekordów, list i drzew
- Nazwy elementów opisują dane w sposób samowyjaśniający

Zalety XML

- Format plików XML to zwykły tekst
- Korzysta standardowo z Unicode, najczęściej z kodowania UTF-8
- Prosta reprezentacja podstawowych struktur danych: rekordów, list i drzew
- Nazwy elementów opisują dane w sposób samowyjaśniający
- Rygorystyczna składnia ułatwia przetwarzanie

Zalety XML

- Format plików XML to zwykły tekst
- Korzysta standardowo z Unicode, najczęściej z kodowania UTF-8
- Prosta reprezentacja podstawowych struktur danych: rekordów, list i drzew
- Nazwy elementów opisują dane w sposób samowyjaśniający
- Rygorystyczna składnia ułatwia przetwarzanie
- Oparty na międzynarodowych standardach

Zalety XML

- Format plików XML to zwykły tekst
- Korzysta standardowo z Unicode, najczęściej z kodowania UTF-8
- Prosta reprezentacja podstawowych struktur danych: rekordów, list i drzew
- Nazwy elementów opisują dane w sposób samowyjaśniający
- Rygorystyczna składnia ułatwia przetwarzanie
- Oparty na międzynarodowych standardach
- Pozwala na walidację składniową języków dedykowanych

Zalety XML

- Format plików XML to zwykły tekst
- Korzysta standardowo z Unicode, najczęściej z kodowania UTF-8
- Prosta reprezentacja podstawowych struktur danych: rekordów, list i drzew
- Nazwy elementów opisują dane w sposób samowyjaśniający
- Rygorystyczna składnia ułatwia przetwarzanie
- Oparty na międzynarodowych standardach
- Pozwala na walidację składniową języków dedykowanych
- Struktura hierarchiczna jest wystarczająca dla wielu zagadnień

Zalety XML

- Format plików XML to zwykły tekst
- Korzysta standardowo z Unicode, najczęściej z kodowania UTF-8
- Prosta reprezentacja podstawowych struktur danych: rekordów, list i drzew
- Nazwy elementów opisują dane w sposób samowyjaśniający
- Rygorystyczna składnia ułatwia przetwarzanie
- Oparty na międzynarodowych standardach
- Pozwala na walidację składniową języków dedykowanych
- Struktura hierarchiczna jest wystarczająca dla wielu zagadnień
- Niezależny od platformy

Wady XML

- Składnia XML jest nadmierna dla prostych danych (np. dane tablice)

Wady XML

- Składnia XML jest nadmierna dla prostych danych (np. dane tablice)
- Nadmierność ma niekorzystny wpływ na wydajność programów poprzez większe wymagania pamięciowe i obliczeniowe

Wady XML

- Składnia XML jest nadmierna dla prostych danych (np. dane tablice)
- Nadmierność ma niekorzystny wpływ na wydajność programów poprzez większe wymagania pamięciowe i obliczeniowe
- Model hierarchiczny jest ograniczony w porównaniu z modelami grafowymi

Wady XML

- Składnia XML jest nadmierna dla prostych danych (np. dane tablice)
- Nadmierność ma niekorzystny wpływ na wydajność programów poprzez większe wymagania pamięciowe i obliczeniowe
- Model hierarchiczny jest ograniczony w porównaniu z modelami grafowymi
- Modelowanie związków typu "zachodzenie na siebie" węzłów jest trudne

Wady XML

- Składnia XML jest nadmierna dla prostych danych (np. dane tablice)
- Nadmierność ma niekorzystny wpływ na wydajność programów poprzez większe wymagania pamięciowe i obliczeniowe
- Model hierarchiczny jest ograniczony w porównaniu z modelami grafowymi
- Modelowanie związków typu "zachodzenie na siebie" węzłów jest trudne
- Rozróżnienie pomiędzy treścią a atrybutami wydaje się nienaturalne i utrudnia tworzenie struktur danych w XML

XHMTL - Powtórka

- XHMTL to eXtensible HyperText Markup Language

[1] Każdy standardowy parser XML poradzi sobie z XHTML ale niekoniecznie z HTML

[2] To akurat zależy niestety w znacznym stopniu od przeglądarki

XHMTL - Powtórka

- XHMTL to eXtensible HyperText Markup Language
- Jest rozwinięciem standardu HTML 4.01
- Można powiedzieć, że jest przecięciem HTML 4.01 i XML (oba języki są podzbiorami SGML)

[1] Każdy standardowy parser XML poradzi sobie z XHTML ale niekoniecznie z HTML

[2] To akurat zależy niestety w znacznym stopniu od przeglądarki

XHMTL - Powtórka

- XHMTL to eXtensible HyperText Markup Language
- Jest rozwinięciem standardu HTML 4.01
- Można powiedzieć, że jest przecięciem HTML 4.01 i XML (oba języki są podzbiorami SGML)
- XHTML jest lepszy pod względem automatycznego przetwarzania[1] (bardziej rygorystyczna składnia)

[1] Każdy standardowy parser XML poradzi sobie z XHTML ale niekoniecznie z HTML

[2] To akurat zależy niestety w znacznym stopniu od przeglądarki

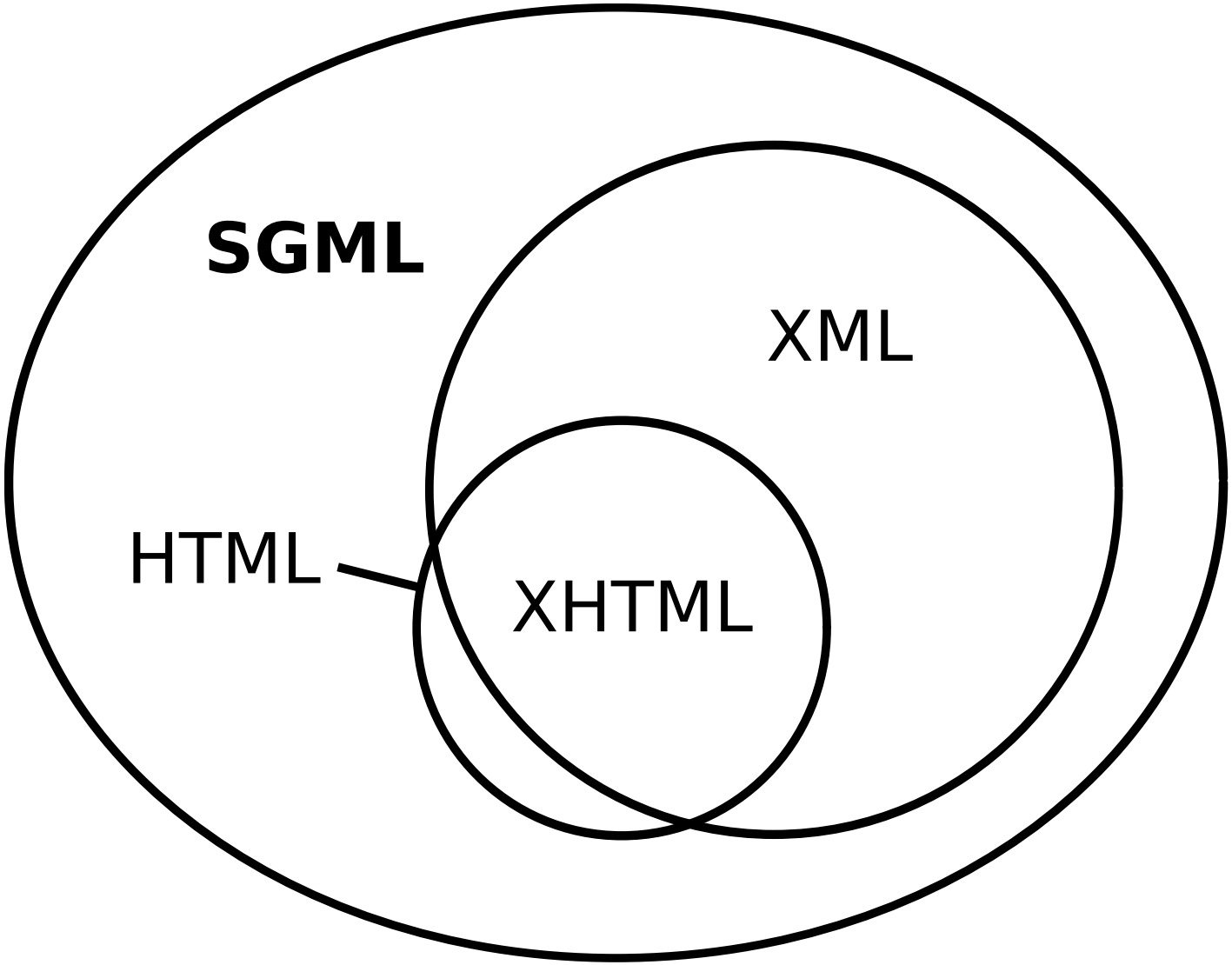
XHMTL - Powtórka

- XHMTL to eXtensible HyperText Markup Language
- Jest rozwinięciem standardu HTML 4.01
- Można powiedzieć, że jest przecięciem HTML 4.01 i XML (oba języki są podzbiorami SGML)
- XHTML jest lepszy pod względem automatycznego przetwarzania[1] (bardziej rygorystyczna składnia)
- Pozwala na korzystanie z różnych rozszerzeń XML, np. MathML, SVG itp. [2]

[1] Każdy standardowy parser XML poradzi sobie z XHTML ale niekoniecznie z HTML

[2] To akurat zależy niestety w znacznym stopniu od przeglądarki

HTML, XHTML i XML



XHTML w porównaniu do HTML (1)

- Każdemu znacznikowi otwierającemu odpowiada znacznik zamykający (np. `<li> ... </li>`)

XHTML w porównaniu do HTML (1)

- Każdemu znacznikowi otwierającemu odpowiada znacznik zamykający (np. `<li> ... </li>`)
- Puste elementy są także zamykane (np. zamiast `<br>` stosujemy `<br />`)

XHTML w porównaniu do HTML (1)

- Każdemu znacznikowi otwierającemu odpowiada znacznik zamykający (np. `<li> ... </li>`)
- Puste elementy są także zamykane (np. zamiast `<br>` stosujemy `<br />`)
- Poprawne zagnieżdżanie (np. zamiast `<p>tekst <em>wyróżniony</p></em>` - `<p>tekst <em>wyróżniony</em></p>`)

XHTML w porównaniu do HTML (1)

- Każdemu znacznikowi otwierającemu odpowiada znacznik zamykający (np. `<li> ... </li>`)
- Puste elementy są także zamykane (np. zamiast `<br>` stosujemy `<br />`)
- Poprawne zagnieżdżanie (np. zamiast `<p>tekst <em>wyróżniony</p></em>` - `<p>tekst <em>wyróżniony</em></p>`)
- Nazwy elementów i atrybutów pisane małymi literami

XHTML w porównaniu do HTML (1)

- Każdemu znacznikowi otwierającemu odpowiada znacznik zamykający (np. `<li> ... </li>`)
- Puste elementy są także zamykane (np. zamiast `<br>` stosujemy `<br />`)
- Poprawne zagnieżdżanie (np. zamiast `<p>tekst <em>wyróżniony</p></em>` - `<p>tekst <em>wyróżniony</em></p>`)
- Nazwy elementów i atrybutów pisane małymi literami
- Wartości artrybutów w cudzysłowie (np. `<td rowspan="3">`)

XHTML w porównaniu do HTML (1)

- Każdemu znacznikowi otwierającemu odpowiada znacznik zamykający (np. `<li> ... </li>`)
- Puste elementy są także zamykane (np. zamiast `<br>` stosujemy `<br />`)
- Poprawne zagnieżdżanie (np. zamiast `<p>tekst <em>wyróżniony</p></em>` - `<p>tekst <em>wyróżniony</em></p>`)
- Nazwy elementów i atrybutów pisane małymi literami
- Wartości artrybutów w cudzysłowie (np. `<td rowspan="3">`)
- Niedozwolona minimalizacja elementów (np. zamiast `<textarea readonly>` - `<textarea readonly="readonly">`)

XHTML w porównaniu do HTML (2)

- Główny element `hmtl` musi zawierać atrybut `xmlns` (np. `<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl">`)

XHTML w porównaniu do HTML (2)

- Główny element `hmtl` musi zawierać atrybut `xmlns` (np. `<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl">`)
- Dokument rozpoczyna się od (opcjonalnej) deklaracji XML, np. `<?xml version="1.0" encoding="UTF-8"?>`

XHTML w porównaniu do HTML (2)

- Główny element `hmtl` musi zawierać atrybut `xmlns` (np. `<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl">`)
- Dokument rozpoczyna się od (opcjonalnej) deklaracji XML, np. `<?xml version="1.0" encoding="UTF-8"?>`
- Należy zastosować odpowiednią definicję typu dokumentu (np. dla XHMTL 1.0 `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`)

Przykład dokumentu XHTML 1.0

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl">
  <head>
    <title>Przykład dokumentu zgodnego z XHTML 1.0 Strict</title>
    <link rel="stylesheet" type="text/css" href="style.css" />
  </head>
  <body>
    <p>To jest przykład.</p>
  </body>
</html>
```

Poprawność dokumentów XML

Każdy dokument XML musi spełniać dwa rodzaje poprawności:

Poprawność dokumentów XML

Każdy dokument XML musi spełniać dwa rodzaje poprawności:

- **Poprawność składniową** (*well-formedness*) - zgodność z podstawowymi regułami składni XML

Poprawność dokumentów XML

Każdy dokument XML musi spełniać dwa rodzaje poprawności:

- **Poprawność składniową** (*well-formedness*) - zgodność z podstawowymi regułami składni XML
- **Poprawność strukturalną/semantyczna** (*validity*) - zgodność z daną definicją typu dokumentu (DTD)

Poprawność dokumentów XML

Każdy dokument XML musi spełniać dwa rodzaje poprawności:

- **Poprawność składniową** (*well-formedness*) - zgodność z podstawowymi regułami składni XML
- **Poprawność strukturalną/semantyczna** (*validity*) - zgodność z daną definicją typu dokumentu (DTD)

Poprawność składniowa jest sprawdzana przez dowolny parser XML, w tym np. przeglądarka Firefox. Jeśli występuje błąd, to parser ma obowiązek nie wyświetlić dokumentu.

Poprawność dokumentów XML

Każdy dokument XML musi spełniać dwa rodzaje poprawności:

- **Poprawność składniową** (*well-formedness*) - zgodność z podstawowymi regułami składni XML
- **Poprawność strukturalną/semantyczna** (*validity*) - zgodność z daną definicją typu dokumentu (DTD)

Poprawność składniowa jest sprawdzana przez dowolny parser XML, w tym np. przeglądarka Firefox. Jeśli występuje błąd, to parser ma obowiązek nie wyświetlić dokumentu.

Poprawność strukturalna jest sprawdzana przez tzw. *walidatory*. Walidator porównuje dokument z podaną definicją typu dokumentu.

Poprawność składniowa XML

Kryteria (te same jak dla XHTML, tylko ogólniejsze):

- Istnieje dokładnie jeden element główny (np. `<html>...</html>`)

Poprawność składniowa XML

Kryteria (te same jak dla XHTML, tylko ogólniejsze):

- Istnieje dokładnie jeden element główny (np. `<html> ...</html>`)
- Główny element może by poprzedzony opcjonalną deklaracją XML `<?xml version="1.0" encoding="UTF-8"?>`

Poprawność składniowa XML

Kryteria (te same jak dla XHTML, tylko ogólniejsze):

- Istnieje dokładnie jeden element główny (np. `<html> ...</html>`)
- Główny element może by poprzedzony opcjonalną deklaracją XML `<?xml version="1.0" encoding="UTF-8"?>`
- Tekst objęty elementem głównym może zawierać dowolną ilość podelementów

Poprawność składniowa XML

Kryteria (te same jak dla XHTML, tylko ogólniejsze):

- Istnieje dokładnie jeden element główny (np. `<html> ...</html>`)
- Główny element może by poprzedzony opcjonalną deklaracją XML `<?xml version="1.0" encoding="UTF-8"?>`
- Tekst objęty elementem głównym może zawierać dowolną ilość podelementów
- Nazwy podelementów są dowolne (Uwaga: `<name>` != `<NAME>`)

Poprawność składniowa XML

Kryteria (te same jak dla XHTML, tylko ogólniejsze):

- Istnieje dokładnie jeden element główny (np. `<html> ...</html>`)
- Główny element może by poprzedzony opcjonalną deklaracją XML `<?xml version="1.0" encoding="UTF-8"?>`
- Tekst objęty elementem głównym może zawierać dowolną ilość podelementów
- Nazwy podelementów są dowolne (Uwaga: `<name>` != `<NAME>`)
- Każdy element musi zostać zamknięty (za pomocą znacznika zamykającego lub jako element samozamykający)

Poprawność składniowa XML

Kryteria (te same jak dla XHTML, tylko ogólniejsze):

- Istnieje dokładnie jeden element główny (np. `<html> ...</html>`)
- Główny element może by poprzedzony opcjonalną deklaracją XML `<?xml version="1.0" encoding="UTF-8"?>`
- Tekst objęty elementem głównym może zawierać dowolną ilość podelementów
- Nazwy podelementów są dowolne (Uwaga: `<name>` != `<NAME>`)
- Każdy element musi zostać zamknięty (za pomocą znacznika zamykającego lub jako element samozamykający)
- Elementy muszą być poprawnie zagnieżdżone

Poprawność składniowa XML

Kryteria (te same jak dla XHTML, tylko ogólniejsze):

- Istnieje dokładnie jeden element główny (np. `<html> ...</html>`)
- Główny element może by poprzedzony opcjonalną deklaracją XML `<?xml version="1.0" encoding="UTF-8"?>`
- Tekst objęty elementem głównym może zawierać dowolną ilość podelementów
- Nazwy podelementów są dowolne (Uwaga: `<name>` != `<NAME>`)
- Każdy element musi zostać zamknięty (za pomocą znacznika zamykającego lub jako element samozamykający)
- Elementy muszą być poprawnie zagnieżdżone
- Wartości atrybutów muszą być objęte cudzysłowem

Przykład dokumentu XML (leksykon.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<leksykon>
  <wpis nr="1">
     <lemat>analiza</lemat>
     <informacje>
        <część_mowy>rzeczownik</część_mowy>
        <rodzaj>żeński</rodzaj>
        <semantyka>process</semantyka>
     </informacje>
     <formy>
        <forma p="mian" l="1">analiza</forma>
        <forma p="dop"  l="1">analizy</forma>
        <forma p="cel"  l="1">analizie</forma>
        ...
     </formy>
  </wpis>
  ...
</leksykon>
```

Języki dedykowane z rodziny XML

- Języki określone z pomocą języka XML

Języki dedykowane z rodziny XML

- Języki określone z pomocą języka XML
- Spełniają wszystkie założenia dotyczące składni XML

Języki dedykowane z rodziny XML

- Języki określone z pomocą języka XML
- Spełniają wszystkie założenia dotyczące składni XML
- Najczęściej nazwy elementów i ich własności są ograniczane

Języki dedykowane z rodziny XML

- Języki określone z pomocą języka XML
- Spełniają wszystkie założenia dotyczące składni XML
- Najczęściej nazwy elementów i ich własności są ograniczane
- Tzn. mamy ograniczenia **semantyczne** w porównaniu z pełnym XML

Języki dedykowane z rodziny XML

- Języki określone z pomocą języka XML
- Spełniają wszystkie założenia dotyczące składni XML
- Najczęściej nazwy elementów i ich własności są ograniczane
- Tzn. mamy ograniczenia **semantyczne** w porównaniu z pełnym XML
- Przykłady: XHTML, MathML, Open Document, SVG ...

Języki dedykowane z rodziny XML

- Języki określone z pomocą języka XML
- Spełniają wszystkie założenia dotyczące składni XML
- Najczęściej nazwy elementów i ich własności są ograniczane
- Tzn. mamy ograniczenia **semantyczne** w porównaniu z pełnym XML
- Przykłady: XHTML, MathML, Open Document, SVG ...
- Np. element `<tr>...</tr>` nie ma **znaczenia** poza `<table>...</table>` w XHTML, składniowo nie byłoby to błędem.

Poprawność semantyczna - Walidacja

- XML nie określa dokładnych reguł składniowych dedykowanych języków z rodziny XML

Poprawność semantyczna - Walidacja

- XML nie określa dokładnych reguł składniowych dedykowanych języków z rodziny XML
- Wykorzystuje się w tym celu dokumenty DTD (document type definition), XML Schema lub inne języki schematów XML

Poprawność semantyczna - Walidacja

- XML nie określa dokładnych reguł składniowych dedykowanych języków z rodziny XML
- Wykorzystuje się w tym celu dokumenty DTD (document type definition), XML Schema lub inne języki schematów XML
- Można za ich pomocą określać np. dozwolone nazwy elementów, dozwolone sposoby zawierania się elementów, dozwolone atrybuty oraz typy danych

Poprawność semantyczna - Walidacja

- XML nie określa dokładnych reguł składniowych dedykowanych języków z rodziny XML
- Wykorzystuje się w tym celu dokumenty DTD (document type definition), XML Schema lub inne języki schematów XML
- Można za ich pomocą określać np. dozwolone nazwy elementów, dozwolone sposoby zawierania się elementów, dozwolone atrybuty oraz typy danych
- Dokument zgodny z określonymi regułami w DTD/schemacie jest poprawny strukturalnie/semantycznie (*valid*)

Poprawność semantyczna - Walidacja

- XML nie określa dokładnych reguł składniowych dedykowanych języków z rodziny XML
- Wykorzystuje się w tym celu dokumenty DTD (document type definition), XML Schema lub inne języki schematów XML
- Można za ich pomocą określać np. dozwolone nazwy elementów, dozwolone sposoby zawierania się elementów, dozwolone atrybuty oraz typy danych
- Dokument zgodny z określonymi regułami w DTD/schemacie jest poprawny strukturalnie/semantycznie (*valid*)
- Dzięki DTD/schematom można sprawdzać automatycznie, czy dany dokument XML jest zgodny z założeniami języka dedykowanego (parser walidujący)

Poprawność strukturalna - sposoby opisu

- DTD - najstarszy i na razie najczęściej stosowany język schematów. W miarę prosty i tym samym ograniczony. Nie jest językiem XML, tylko innym podzbiorem SGML.

Poprawność strukturalna - sposoby opisu

- DTD - najstarszy i na razie najczęściej stosowany język schematów. W miarę prosty i tym samym ograniczony. Nie jest językiem XML, tylko innym podzbiorem SGML.
- XML Schema - wypiera DTD. Język schematu XML, który sam jest opisany za pomocą XML. Duże możliwości, ale niestety trudny.

Poprawność strukturalna - sposoby opisu

- DTD - najstarszy i na razie najczęściej stosowany język schematów. W miarę prosty i tym samym ograniczony. Nie jest językiem XML, tylko innym podzbiorem SGML.
- XML Schema - wypiera DTD. Język schematu XML, który sam jest opisany za pomocą XML. Duże możliwości, ale niestety trudny.
- Relax NG - Opisany w XML, prostszy od XML Schema, rzadziej używany

Poprawność strukturalna - sposoby opisu

- DTD - najstarszy i na razie najczęściej stosowany język schematów. W miarę prosty i tym samym ograniczony. Nie jest językiem XML, tylko innym podzbiorem SGML.
- XML Schema - wypiera DTD. Język schematu XML, który sam jest opisany za pomocą XML. Duże możliwości, ale niestety trudny.
- Relax NG - Opisany w XML, prostszy od XML Schema, rzadziej używany
- ISO DSDL - Zbiór mniejszych języków schematów XML

Przykład dokumentu DTD (leksykon.dtd)

```
<!ELEMENT leksykon (wpis*)>
<!ELEMENT wpis (lemat, informacje, formy)>
<!ATTLIST wpis nr ID #REQUIRED>
<!ELEMENT lemat (#PCDATA)>
<!ELEMENT informacje (część_mowy, rodzaj?, semantyka?)>
<!ELEMENT część_mowy (#PCDATA)>
<!ELEMENT rodzaj (#PCDATA)>
<!ELEMENT semantyka (#PCDATA)>
<!ELEMENT formy (forma+)>
<!ELEMENT forma (#PCDATA)>
<!ATTLIST forma p CDATA #IMPLIED l CDATA #IMPLIED>
```

Przykład dokumentu XML (leksykon.xml) z DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE leksykon SYSTEM "leksykon.dtd">
<leksykon>
  <wpis nr="1">
     <lemat>analiza</lemat>
     <informacje>
        <część_mowy>rzeczownik</część_mowy>
        <rodzaj>żeński</rodzaj>
        <semantyka>process</semantyka>
     </informacje>
     <formy>
        <forma p="mian" l="1">analiza</forma>
        <forma p="dop"  l="1">analizy</forma>
        <forma p="cel"  l="1">analizie</forma>
        ...
     </formy>
  </wpis>
  ...
</leksykon>
```

Przykłady formatów XML - MathML

```
<math>
 <mrow>
  <msup>
   <mfenced>
    <mrow>
     <mi>x</mi>
     <mo>+</mo>
     <mi>y</mi>
    </mrow>
   </mfenced>
   <mn>4</mn>
  </msup>
 </mrow>
</math>
```

Przykłady formatów XML - MathML

```
<math>
 <mrow>
  <msup>
   <mfenced>
    <mrow>
     <mi>x</mi>
     <mo>+</mo>
     <mi>y</mi>
    </mrow>
   </mfenced>
   <mn>4</mn>
  </msup>
 </mrow>
</math>
```

oraz w zapisie czytelnym ale nie do przetworzenia: $(x + y)^4$

Przykłady formatów XML - SVG

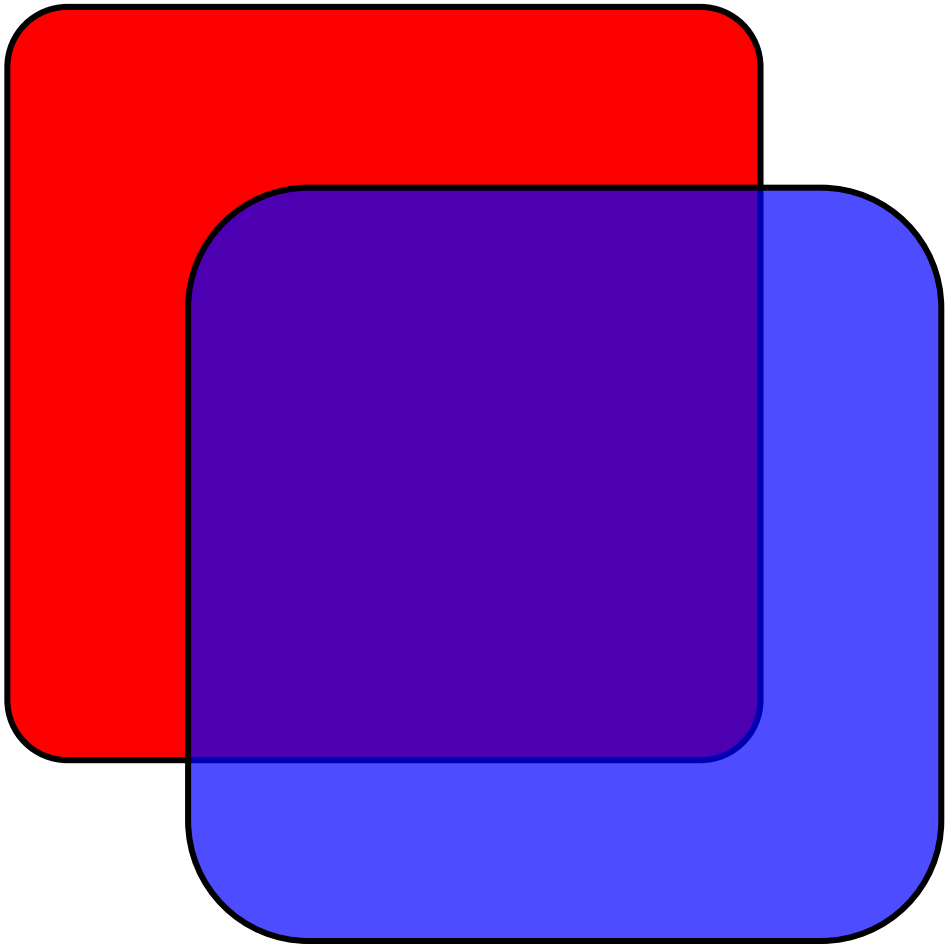
```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
    "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

<svg xmlns="http://www.w3.org/2000/svg" version="1.1"
     width="467" height="462">

  <!-- This is for the red square -->
  <rect x="80" y="60" width="250" height="250" rx="20" fill="red"
        stroke="black" stroke-width="2px" />
  <!-- This is for the blue square -->
  <rect x="140" y="120" width="250" height="250" rx="40" fill="blue"
        fill-opacity="0.7" stroke="black" stroke-width="2px" />

</svg>
```

Przykłady formatów XML - SVG (Wynik)



Rozszerzanie XML

- XPath - Podjęzyk umożliwiający dostęp do elementów dokumentu XML, porównywalny ze strukturą katalogów na dysku

Rozszerzanie XML

- XPath - Podjęzyk umożliwiający dostęp do elementów dokumentu XML, porównywalny ze strukturą katalogów na dysku
- XSL (EXtensible Stylesheet Language) - funkcyjny język programowania opisujący sposób prezentacji i przekształceń dokumentów zapisanych w XML

Rozszerzanie XML

- XPath - Podjęzyk umożliwiający dostęp do elementów dokumentu XML, porównywalny ze strukturą katalogów na dysku
- XSL (EXtensible Stylesheet Language) - funkcyjny język programowania opisujący sposób prezentacji i przekształceń dokumentów zapisanych w XML
- XQuery - Język zapytań do baz danych zapisanych w XML
- ...