

XML

i nowoczesne metody zarządzania treścią

Wykład 1: Od zecera do XML-a

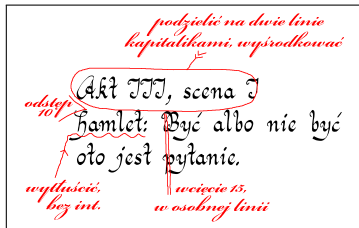
Maciej Ogrodniczuk

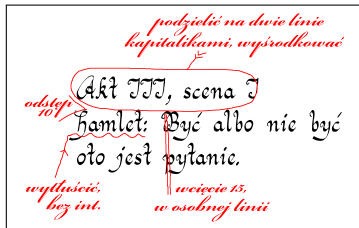
MIMUW, 1 października 2009

Znakowanie tekstu: prehistoria

*Akt III, scena I
Hamlet: Być albo nie być
oto jest pytanie.*

Znakowanie tekstu: prehistoria





AKT III

SCENA I

Hamlet

Być albo nie być —
oto jest pytanie.

[Włącz środkowanie]

[Włącz kapitaliki]

Akt III

[Nowa linia]

Scena I

[Nowa linia]

[Nowa linia]

[Wyłącz kapitaliki]

[Wyłącz środkowanie]

[Włącz wytłuszczenie]

Hamlet

[Wyłącz wytłuszczenie]

[Wcięcie 10 pt]

Być albo nie być —

oto jest pytanie.

→

AKT III

SCENA I

Hamlet

Być albo nie być —
oto jest pytanie.

Z czego składa się dokument?

[Oznaczenie
aktu]

Akt III

[Oznaczenie
sceny]

Scena I

[Postać]

Hamlet

[Kwestia]

Być albo
nie być —
oto jest
pytanie.

+

Oznaczenie aktu,
oznaczenie sceny:

[Tekst
wyśrodkowany]

[Kapitalikami]
[W osobnym
akapicie]

Postać:

[Tekst
wytłuszczony]
[W osobnym
akapicie]

Kwestia:

[Akapit wcięty
10 pt]

=

AKT III
SCENA I

Hamlet

Być albo nie być —
oto jest pytanie.

Inna wizualizacja tego samego dokumentu

[Oznaczenie
aktu]

Akt III

[Oznaczenie
sceny]

Scena I

[Postać]

Hamlet

[Kwestia]

Być albo
nie być —
oto jest
pytanie.

+

Oznaczenie aktu,
oznaczenie sceny:

[Wyśrodkowany,
biały na czar-
nym/brązowym
tle, w osobnym
akapicie]

Postać:

[Wytłuszczony,
wyśrodkowany,
w osobnym
akapicie]

Kwestia:

[Inicjał ozdobny]

=

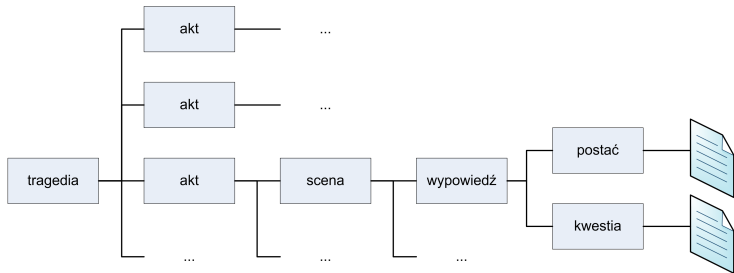
Akt III

Scena I

Hamlet

Być albo nie być —
oto jest pytanie.

Z czego tak naprawdę składa się dokument?



USTAWA

z dnia 2 lipca 2004 r.

o zmianie ustawy – Kodeks postępowania cywilnego oraz niektórych innych ustaw

Art. 1.

W ustawie z dnia 17 listopada 1964 r. – Kodeks postępowania cywilnego (Dz. U. Nr 43, poz. 296, z późn. zm.) wprowadza się następujące zmiany:

1) art. 5 otrzymuje brzmienie:

„Art. 5. W razie uzasadnionej potrzeby sąd może udzielić stronom i uczestnikom postępowania występującym w sprawie bez adwokata lub radcy prawnego niezbędnych pouczeń co do czynności procesowych.”;

...

USTAWA

z dnia 2 lipca 2004 r.

o zmianie ustawy – Kodeks postępowania cywilnego oraz niektórych innych ustaw

Art. 1.

W ustawie z dnia 17 listopada 1964 r. – Kodeks postępowania cywilnego (Dz. U. Nr 43, poz. 296, z późn. zm.) wprowadza się następujące zmiany:

1) art. 5 otrzymuje brzmienie:

„Art. 5. W razie uzasadnionej potrzeby sąd może udzielić stronom i uczestnikom postępowania występującym w sprawie bez adwokata lub radcy prawnego niezbędnych pouczeń co do czynności procesowych.”;

...

USTAWA

z dnia 2 lipca 2004 r.

o zmianie ustawy – Kodeks postępowania cywilnego oraz niektórych innych ustaw

Art. 1.

W ustawie z dnia 17 listopada 1964 r. – Kodeks postępowania cywilnego (Dz. U. Nr 43, poz. 296, z późn. zm.) wprowadza się następujące zmiany:

1) art. 5 otrzymuje brzmienie:

„**Art. 5.** W razie uzasadnionej potrzeby sąd może udzielić stronom i uczestnikom postępowania występującym w sprawie bez adwokata lub radcy prawnego niezbędnych pouczeń co do czynności procesowych.”;

...

Generalized Markup Language (Goldfarb, Mosher, Lorie):

- znaczniki opisują strukturę dokumentu,
- są zamieniane na polecenia formatujące,
- można definiować wiele „profilu” wizualizujących dokument,
- można rozszerzać początkowy zbiór znaczników.

```
:p.GML supported hierarchical containers, such as
:ol
:li.Ordered lists (like this one),
:li.Unordered lists, and
:li.Definition lists
:eol.
as well as simple structures.
:p.Markup minimization (later generalized and
formalized in SGML),
allowed the end-tags to be omitted for the "h1" and
"p" elements.
```

<http://www.sgmlsource.com/history/roots.htm>

Przykłady znakowania

Ventura	<code>Hamlet<D></code>
T_EX/L_AT_EX	<code>\textbf{Hamlet}</code>
QuarkXPress	<code>Hamlet</code>
RTF	<code>{\b\f5\cf1 Hamlet}</code>
HTML	<code>Hamlet</code>
PostScript	<code>/Times-BoldR 900 ff(Hamlet)W</code>
Word 2007	<code><w:r><w:rPr><w:b/></w:rPr> <w:t>Hamlet</w:t></w:r></code>

Problemy:

- prawie każda aplikacja wprowadza własny format zapisu i wymiany danych,
- nowe wersje aplikacji często wprowadzają zmiany do używanego formatu — występuje problem ze zgodnością wstecz (często: brak możliwości zapisu do formatu poprzednich wersji),
- aplikacje dostarczają konwerterów, ale:
 - tylko do najpopularniejszych formatów,
 - podczas konwersji możliwa jest utrata części danych.

Chcielibyśmy mieć format:

- na tyle elastyczny i uniwersalny, by mógł reprezentować wiele formatów dokumentów,
- na tyle jednorodny i prosty, by można było łatwo budować przetwarzające go narzędzia.

1986: SGML — Standard Generalized Markup Language

SGML — norma ISO 8879:1986.

1986: SGML — Standard Generalized Markup Language

SGML — norma ISO 8879:1986.

3 najważniejsze cechy SGML-a:

SGML — norma ISO 8879:1986.

3 najważniejsze cechy SGML-a:

- 1 deskryptywność opisu,

SGML — norma ISO 8879:1986.

3 najważniejsze cechy SGML-a:

- 1 deskryptywność opisu,
- 2 typizacja dokumentów,

SGML — norma ISO 8879:1986.

3 najważniejsze cechy SGML-a:

- 1 deskryptywność opisu,
- 2 typizacja dokumentów,
- 3 maksymalna przenośność.

SGML — norma ISO 8879:1986.

3 najważniejsze cechy SGML-a:

- 1 deskryptywność opisu,
- 2 typizacja dokumentów,
- 3 maksymalna przenośność.

1994: HTML 2.0 zdefiniowany jako zastosowanie SGML-a.

Niby niewiele wad, ale podstawowe...

- 1 Konieczność definiowania struktury.

Niby niewiele wad, ale podstawowe...

- 1 Konieczność definiowania struktury.
- 2 Duża złożoność, zbyt wiele opcji.

Niby niewiele wad, ale podstawowe...

- 1 Konieczność definiowania struktury.
- 2 Duża złożoność, zbyt wiele opcji.
- 3 Trudność tworzenia narzędzi (drogo!)

XML — standard W3C:

- **1.0:**
 - wydanie I z lutego 1998 r.,
 - **wydanie V z 2008 r.,**

XML — standard W3C:

- **1.0:**
 - wydanie I z lutego 1998 r.,
 - **wydanie V z 2008 r.,**
- 1.1 — od 2004 r.,

XML — standard W3C:

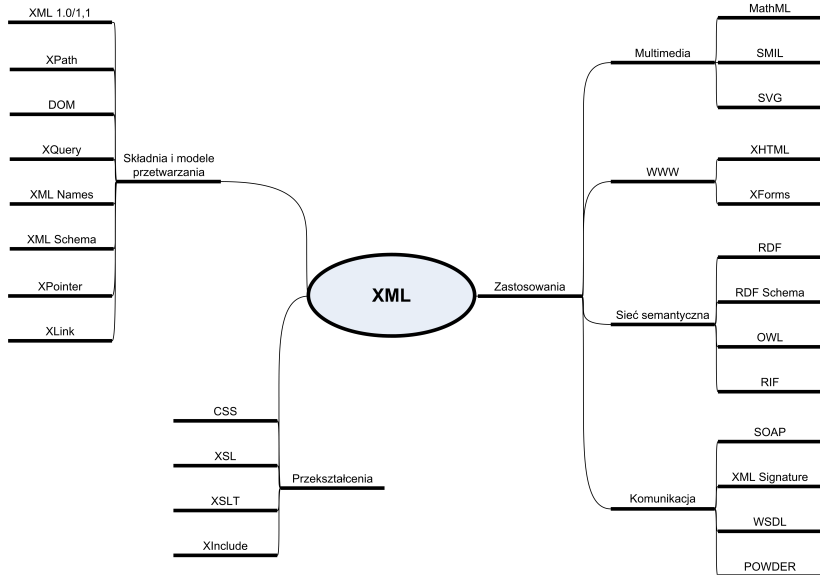
- **1.0:**
 - wydanie I z lutego 1998 r.,
 - **wydanie V z 2008 r.,**
- 1.1 — od 2004 r.,
- 2.0 — więcej pytań niż odpowiedzi.

XML — standard W3C:

- **1.0:**
 - wydanie I z lutego 1998 r.,
 - **wydanie V z 2008 r.,**
- 1.1 — od 2004 r.,
- 2.0 — więcej pytań niż odpowiedzi.

<http://www.w3.org/TR/REC-xml>

Rodzina standardów XML-owych





`www.w3.org`

W3C to międzynarodowe konsorcjum rozwijające standardy internetowe (nie tylko XML-owe, choć XML w sposób naturalny dominuje).

USTAWA
o realizacji przedpłat na samochody

1. Osoby, które w 1981 r. wniosły przedpłaty na zakup samochodów i dotychczas samochodu nie odebrały, mają prawo do otrzymania rekompensaty w wysokości:
 - 1) 5.930 zł, jeśli przedpłata została wniesiona na samochód marki Fiat 126p,
 - 2) 8.400 zł dla samochodu marki FSO 1500.

USTAWA

`<tytuł>o realizacji przedpłat na samochody</tytuł>`

1. Osoby, które w 1981 r. wniosły przedpłaty na zakup samochodów i dotychczas samochodu nie odebrały, mają prawo do otrzymania rekompensaty w wysokości:
 - 1) 5.930 zł, jeśli przedpłata została wniesiona na samochód marki Fiat 126p,
 - 2) 8.400 zł dla samochodu marki FSO 1500.

USTAWA

`<tytuł>o realizacji przedpłat na samochody</tytuł>`
`<ustęp>`

1. Osoby, które w 1981 r. wniosły przedpłaty na zakup samochodów i dotychczas samochodu nie odebrały, mają prawo do otrzymania rekompensaty w wysokości:

1) 5.930 zł, jeśli przedpłata została wniesiona na samochód marki Fiat 126p,

2) 8.400 zł dla samochodu marki FSO 1500.

`</ustęp>`

USTAWA

<tytuł>o realizacji przedpłat na samochody</tytuł>
<ustęp>

1. Osoby, które w 1981 r. wniosły przedpłaty na zakup samochodów i dotychczas samochodu nie odebrały, mają prawo do otrzymania rekompensaty w wysokości:
<punkt>

1) 5.930 zł, jeśli przedpłata została wniesiona na samochód marki Fiat 126p,</punkt>

2) 8.400 zł dla samochodu marki FSO 1500.

</ustęp>

USTAWA

<tytuł>o realizacji przedpłat na samochody</tytuł>
<ustęp>

1. Osoby, które w 1981 r. wniosły przedpłaty na zakup samochodów i dotychczas samochodu nie odebrały, mają prawo do otrzymania rekompensaty w wysokości:

<punkt>

1) 5.930 zł, jeśli przedpłata została wniesiona na samochód marki Fiat 126p,</punkt>

<punkt>

2) 8.400 zł dla samochodu marki FSO 1500.</punkt>

</ustęp>

Jak wygląda XML?

```
<ustawa>
  <tytuł>o realizacji przedpłat na samochody</tytuł>
  <ustęp>
    1. Osoby, które w 1981 r. wniosły przedpłaty na zakup
      samochodów i dotychczas samochodu nie odebrały,
      mają prawo do otrzymania rekompensaty w wysokości:
      <punkt>
        1) 5.930 zł, jeśli przedpłata została wniesiona
          na samochód marki Fiat 126p,</punkt>
      <punkt>
        2) 8.400 zł dla samochodu marki FSO 1500.</punkt>
      </ustęp>
  </ustawa>
```

Jak wygląda XML?

```
<ustawa>
  <tytuł>o realizacji przedpłat na samochody</tytuł>
  <ustęp nr="1">
    Osoby, które w 1981 r. wniosły przedpłaty na zakup
    samochodów i dotychczas samochodu nie odebrały,
    mają prawo do otrzymania rekompensaty w wysokości:
    <punkt nr="1">
      5.930 zł, jeśli przedpłata została wniesiona
      na samochód marki Fiat 126p,</punkt>
    <punkt nr="2">
      8.400 zł dla samochodu marki FSO 1500.</punkt>
    </ustęp>
  </ustawa>
```

Jak wygląda XML?

```
<?xml version="1.0" encoding="iso-8859-2"?>
```

```
<ustawa>
```

```
  <tytuł>o realizacji przedpłat na samochody</tytuł>
```

```
  <ustęp nr="1">
```

Osoby, które w 1981 r. wniosły przedpłaty na zakup samochodów i dotychczas samochodu nie odebrały, mają prawo do otrzymania rekompensaty w wysokości:

```
  <punkt nr="1">
```

5.930 zł, jeśli przedpłata została wniesiona na samochód marki Fiat 126p,</punkt>

```
  <punkt nr="2">
```

8.400 zł dla samochodu marki FSO 1500.</punkt>

```
</ustęp>
```

```
</ustawa>
```

Z czego składa się dokument XML-owy?

Deklaracja XML-owa:

```
<?xml version="1.0"  
      encoding="windows-1250"  
      standalone="yes"?>
```

- Zawsze na początku dokumentu (w pierwszej linii, od pierwszego znaku).
- Deklarację można pominąć, jeśli dokument jest w wersji 1.0, a jego kodowanie to UTF-8 lub UTF-16.
- Definicja ze specyfikacji: XMLDecl ::= '<?xml' VersionInfo EncodingDecl? SDDDecl? S? '?>'
- Sprawy kodowania → aneks F specyfikacji XML-owej.

Z czego składa się dokument XML-owy?

Elementy i atrybuty:

```
<element atrybut="wartość">  
  <pusty1></pusty1>  
  <pusty2/>  
</element>
```

- Dokument zawiera dokładnie jeden element główny.
- Elementy mogą zawierać podelementy i węzły tekstowe.
- Atrybuty muszą być ujęte w cudzysłowy lub apostrofy.
- Dany element nie może posiadać dwóch atrybutów o tej samej nazwie.
- Wartości atrybutów nie mogą zawierać znaku <.

Z czego składa się dokument XML-owy?

Komentarze:

```
<!-- komentarz  
...  
-->
```

- Mogą być wieloliniowe.
- Mogą zawierać dowolne znaki, także <, >, &.
- Nie mogą zawierać w treści dwóch minusów po sobie.
- Nie mogą być wstawiane w etykiety znaczników.
- Mogą wystąpić na zewnątrz elementu głównego.

Z czego składa się dokument XML-owy?

Instrukcje sterujące:

```
<?instrukcja i-jej treść?>
```

- Mogą wystąpić na zewnątrz elementu głównego.
- Nazwa instrukcji musi być różna od ciągu „XML” (i wariantów z różną wielkością liter).

Z czego składa się dokument XML-owy?

Sekcje CDATA:

```
<![CDATA[dowolny <tekst " nie przetwarzany \ przez  
[parser]]>
```

- Jedynym zabronionym ciągiem znaków jest `]]>` (zresztą nie tylko tu — w zawartości tekstowej też, więc w sekcji CDATA można zacytować dokument XML nie zawierający sekcji CDATA).
- Sekcji CDATA nie można zagnieżdżać.

Jeśli dokument jest zbudowany zgodnie z regułami XML-owej składni, to mówimy, że jest **poprawny składniowo** (ang. *well-formed*).

Jeśli dokument jest zbudowany zgodnie z regułami XML-owej składni, to mówimy, że jest **poprawny składniowo** (ang. *well-formed*).

W szczególności:

Jeśli dokument jest zbudowany zgodnie z regułami XML-owej składni, to mówimy, że jest **poprawny składniowo** (ang. *well-formed*).

W szczególności:

- znaczniki muszą być zamknięte i poprawnie zagnieżdżone,

Jeśli dokument jest zbudowany zgodnie z regułami XML-owej składni, to mówimy, że jest **poprawny składniowo** (ang. *well-formed*).

W szczególności:

- znaczniki muszą być zamknięte i poprawnie zagnieżdżone,
- atrybuty muszą być ujęte w apostrofy lub cudzysłowy,

Jeśli dokument jest zbudowany zgodnie z regułami XML-owej składni, to mówimy, że jest **poprawny składniowo** (ang. *well-formed*).

W szczególności:

- znaczniki muszą być zamknięte i poprawnie zagnieżdżone,
- atrybuty muszą być ujęte w apostrofy lub cudzysłowy,
- atrybuty nie mogą się powtarzać,

Jeśli dokument jest zbudowany zgodnie z regułami XML-owej składni, to mówimy, że jest **poprawny składniowo** (ang. *well-formed*).

W szczególności:

- znaczniki muszą być zamknięte i poprawnie zagnieżdżone,
- atrybuty muszą być ujęte w apostrofy lub cudzysłowy,
- atrybuty nie mogą się powtarzać,
- ... i 9 innych wymagań (→ specyfikacja).

Jeśli dokument jest zbudowany zgodnie z regułami XML-owej składni, to mówimy, że jest **poprawny składniowo** (ang. *well-formed*).

W szczególności:

- znaczniki muszą być zamknięte i poprawnie zagnieżdżone,
- atrybuty muszą być ujęte w apostrofy lub cudzysłowy,
- atrybuty nie mogą się powtarzać,
- ... i 9 innych wymagań (→ specyfikacja).

Analizę i weryfikację poprawności dokumentów XML-owych zapewniają **parseery XML-owe**.

Problem:

Jak połączyć dokumenty:

- 1 `<autor>`
 `<tytuł>prof.</tytuł>`
 `<imię>Jan</imię>`
 `<nazwisko>Ciekawski</nazwisko>`
 `</autor>`
- 2 `<rozdział nr="1">`
 `<tytuł>0 sepulkach</tytuł>`
 ...
 `</rozdział>`

Rozwiązanie:

Użycie *przestrzeni nazw* — powiązania między URI/IRI a wybraną „nazwą kontekstu”.

- 1 deklarujemy przestrzeń nazw dla poddrzewa atrybutem `xmlns:prefiks` z wartością URI/IRI — np.
`xmlns:autorzy="http://example.com/autorzy"`,
- 2 aby powiedzieć, że **nazwa** elementu `<imię>` pochodzi z przestrzeni nazw `http://example.com/autorzy`, używamy zapisu `<autorzy:imię>`,
- 3 brak prefiksu (`xmlns="http://example.com/autorzy"`) tworzy tzw. domyślną przestrzeń nazw.

Rozwiązanie:

Użycie *przestrzeni nazw* — powiązania między URI/IRI a wybraną „nazwą kontekstu”.

- 1 deklarujemy przestrzeń nazw dla poddrzewa atrybutem `xmlns:prefiks` z wartością URI/IRI — np.
`xmlns:autorzy="http://example.com/autorzy"`,
- 2 aby powiedzieć, że **nazwa** elementu `<imię>` pochodzi z przestrzeni nazw `http://example.com/autorzy`, używamy zapisu `<autorzy:imię>`,
- 3 brak prefiksu (`xmlns="http://example.com/autorzy"`) tworzy tzw. domyślną przestrzeń nazw.

Najlepszy dokument o przestrzeniach nazw:

<http://www.rpbouret.com/xml/NamespacesFAQ.htm>

Przykład zapisu:

```
<książka>
  <autorzy xmlns:a="http://example.com/authors">
    <a:autor>
      <a:tytuł>prof.</a:tytuł>
      <a:imię>Jan</a:imię>
      <a:nazwisko>Ciekawski</a:nazwisko>
    </a:autor>
  </autorzy>
  <treść>
    <rozdział nr="1">
      <tytuł>0 sepulkach</tytuł>
      ...
    </rozdział>
  </treść>
</książka>
```

Które nazwy należą do jakiej przestrzeni nazw?

Przykład:

```
<a xmlns:b="urn:b">
  <c xmlns="urn:x">
    <d:e xmlns="urn:y"
      xmlns:d="urn:d"
      b:f="g">
      <i xmlns:b="urn:b2">
        <j b:k="l"/>
      </i>
    </d:e>
    <m n="o"/>
  </c>
</p>
</a>
```

Które nazwy należą do jakiej przestrzeni nazw?

Przykład:

```
<a xmlns:b="urn:b">  
  <c xmlns="urn:x">  
    <d:e xmlns="urn:y"  
      xmlns:d="urn:d"  
      b:f="g">  
      <i xmlns:b="urn:b2">  
        <j b:k="l"/>  
      </i>  
    </d:e>  
    <m n="o"/>  
  </c>  
<p/>  
</a>
```

Odpowiedź:

a?

Które nazwy należą do jakiej przestrzeni nazw?

Przykład:

```
<a xmlns:b="urn:b">  
  <c xmlns="urn:x">  
    <d:e xmlns="urn:y"  
      xmlns:d="urn:d"  
      b:f="g">  
      <i xmlns:b="urn:b2">  
        <j b:k="l"/>  
      </i>  
    </d:e>  
    <m n="o"/>  
  </c>  
</p>  
</a>
```

Odpowiedź:

a nie należy do żadnej przestrzeni nazw

Które nazwy należą do jakiej przestrzeni nazw?

Przykład:

```
<a xmlns:b="urn:b">  
  <c xmlns="urn:x">  
    <d:e xmlns="urn:y"  
      xmlns:d="urn:d"  
      b:f="g">  
      <i xmlns:b="urn:b2">  
        <j b:k="l"/>  
      </i>  
    </d:e>  
    <m n="o"/>  
  </c>  
<p/>  
</a>
```

Odpowiedź:

a nie należy do żadnej
przestrzeni nazw
c?

Które nazwy należą do jakiej przestrzeni nazw?

Przykład:

```
<a xmlns:b="urn:b">  
  <c xmlns="urn:x">  
    <d:e xmlns="urn:y"  
      xmlns:d="urn:d"  
      b:f="g">  
      <i xmlns:b="urn:b2">  
        <j b:k="l"/>  
      </i>  
    </d:e>  
    <m n="o"/>  
  </c>  
</a>
```

Odpowiedź:

a nie należy do żadnej przestrzeni nazw
c należy do urn:x

Które nazwy należą do jakiej przestrzeni nazw?

Przykład:

```
<a xmlns:b="urn:b">  
  <c xmlns="urn:x">  
    <d:e xmlns="urn:y"  
      xmlns:d="urn:d"  
      b:f="g">  
      <i xmlns:b="urn:b2">  
        <j b:k="l"/>  
      </i>  
    </d:e>  
    <m n="o"/>  
  </c>  
</a>
```

Odpowiedź:

a nie należy do żadnej przestrzeni nazw
c należy do urn:x
e?

Które nazwy należą do jakiej przestrzeni nazw?

Przykład:

```
<a xmlns:b="urn:b">  
  <c xmlns="urn:x">  
    <d:e xmlns="urn:y"  
      xmlns:d="urn:d"  
      b:f="g">  
      <i xmlns:b="urn:b2">  
        <j b:k="l"/>  
      </i>  
    </d:e>  
    <m n="o"/>  
  </c>  
</a>
```

Odpowiedź:

a nie należy do żadnej przestrzeni nazw
c należy do urn:x
e należy do urn:d

Które nazwy należą do jakiej przestrzeni nazw?

Przykład:

```
<a xmlns:b="urn:b">  
  <c xmlns="urn:x">  
    <d:e xmlns="urn:y"  
      xmlns:d="urn:d"  
      b:f="g">  
      <i xmlns:b="urn:b2">  
        <j b:k="l"/>  
      </i>  
    </d:e>  
    <m n="o"/>  
  </c>  
<p/>  
</a>
```

Odpowiedź:

a nie należy do żadnej przestrzeni nazw
c należy do urn:x
e należy do urn:d
f?

Które nazwy należą do jakiej przestrzeni nazw?

Przykład:

```
<a xmlns:b="urn:b">  
  <c xmlns="urn:x">  
    <d:e xmlns="urn:y"  
      xmlns:d="urn:d"  
      b:f="g">  
      <i xmlns:b="urn:b2">  
        <j b:k="l"/>  
      </i>  
    </d:e>  
    <m n="o"/>  
  </c>  
</a>
```

Odpowiedź:

a nie należy do żadnej przestrzeni nazw
c należy do urn:x
e należy do urn:d
f należy do urn:b

Które nazwy należą do jakiej przestrzeni nazw?

Przykład:

```
<a xmlns:b="urn:b">  
  <c xmlns="urn:x">  
    <d:e xmlns="urn:y"  
      xmlns:d="urn:d"  
      b:f="g">  
      <i xmlns:b="urn:b2">  
        <j b:k="l"/>  
      </i>  
    </d:e>  
    <m n="o"/>  
  </c>  
</a>
```

Odpowiedź:

a nie należy do żadnej przestrzeni nazw
c należy do urn:x
e należy do urn:d
f należy do urn:b
i?

Które nazwy należą do jakiej przestrzeni nazw?

Przykład:

```
<a xmlns:b="urn:b">  
  <c xmlns="urn:x">  
    <d:e xmlns="urn:y"  
      xmlns:d="urn:d"  
      b:f="g">  
      <i xmlns:b="urn:b2">  
        <j b:k="l"/>  
      </i>  
    </d:e>  
    <m n="o"/>  
  </c>  
</a>
```

Odpowiedź:

a nie należy do żadnej przestrzeni nazw
c należy do urn:x
e należy do urn:d
f należy do urn:b
i należy do urn:y

Które nazwy należą do jakiej przestrzeni nazw?

Przykład:

```
<a xmlns:b="urn:b">  
  <c xmlns="urn:x">  
    <d:e xmlns="urn:y"  
      xmlns:d="urn:d"  
      b:f="g">  
      <i xmlns:b="urn:b2">  
        <j b:k="l"/>  
      </i>  
    </d:e>  
    <m n="o"/>  
  </c>  
</a>
```

Odpowiedź:

a nie należy do żadnej przestrzeni nazw
c należy do urn:x
e należy do urn:d
f należy do urn:b
i należy do urn:y
j?

Które nazwy należą do jakiej przestrzeni nazw?

Przykład:

```
<a xmlns:b="urn:b">  
  <c xmlns="urn:x">  
    <d:e xmlns="urn:y"  
      xmlns:d="urn:d"  
      b:f="g">  
      <i xmlns:b="urn:b2">  
        <j b:k="l"/>  
      </i>  
    </d:e>  
    <m n="o"/>  
  </c>  
</a>
```

Odpowiedź:

a nie należy do żadnej przestrzeni nazw
c należy do urn:x
e należy do urn:d
f należy do urn:b
i należy do urn:y
j należy do urn:y

Które nazwy należą do jakiej przestrzeni nazw?

Przykład:

```
<a xmlns:b="urn:b">
  <c xmlns="urn:x">
    <d:e xmlns="urn:y"
      xmlns:d="urn:d"
      b:f="g">
      <i xmlns:b="urn:b2">
        <j b:k="l"/>
      </i>
    </d:e>
    <m n="o"/>
  </c>
</p>
</a>
```

Odpowiedź:

a nie należy do żadnej przestrzeni nazw
c należy do urn:x
e należy do urn:d
f należy do urn:b
i należy do urn:y
j należy do urn:y
k?

Które nazwy należą do jakiej przestrzeni nazw?

Przykład:

```
<a xmlns:b="urn:b">  
  <c xmlns="urn:x">  
    <d:e xmlns="urn:y"  
      xmlns:d="urn:d"  
      b:f="g">  
      <i xmlns:b="urn:b2">  
        <j b:k="l"/>  
      </i>  
    </d:e>  
    <m n="o"/>  
  </c>  
</a>
```

Odpowiedź:

a nie należy do żadnej przestrzeni nazw
c należy do urn:x
e należy do urn:d
f należy do urn:b
i należy do urn:y
j należy do urn:y
k należy do urn:b2

Które nazwy należą do jakiej przestrzeni nazw?

Przykład:

```
<a xmlns:b="urn:b">
  <c xmlns="urn:x">
    <d:e xmlns="urn:y"
      xmlns:d="urn:d"
      b:f="g">
      <i xmlns:b="urn:b2">
        <j b:k="l"/>
      </i>
    </d:e>
    <m n="o"/>
  </c>
</a>
```

Odpowiedź:

a nie należy do żadnej przestrzeni nazw
c należy do urn:x
e należy do urn:d
f należy do urn:b
i należy do urn:y
j należy do urn:y
k należy do urn:b2
m?

Które nazwy należą do jakiej przestrzeni nazw?

Przykład:

```
<a xmlns:b="urn:b">
  <c xmlns="urn:x">
    <d:e xmlns="urn:y"
      xmlns:d="urn:d"
      b:f="g">
      <i xmlns:b="urn:b2">
        <j b:k="l"/>
      </i>
    </d:e>
    <m n="o"/>
  </c>
</a>
```

Odpowiedź:

a nie należy do żadnej przestrzeni nazw
c należy do urn:x
e należy do urn:d
f należy do urn:b
i należy do urn:y
j należy do urn:y
k należy do urn:b2
m należy do urn:x

Które nazwy należą do jakiej przestrzeni nazw?

Przykład:

```
<a xmlns:b="urn:b">  
  <c xmlns="urn:x">  
    <d:e xmlns="urn:y"  
      xmlns:d="urn:d"  
      b:f="g">  
      <i xmlns:b="urn:b2">  
        <j b:k="l"/>  
      </i>  
    </d:e>  
    <m n="o"/>  
  </c>  
</a>
```

Odpowiedź:

a nie należy do żadnej przestrzeni nazw
c należy do urn:x
e należy do urn:d
f należy do urn:b
i należy do urn:y
j należy do urn:y
k należy do urn:b2
m należy do urn:x
n?

Które nazwy należą do jakiej przestrzeni nazw?

Przykład:

```
<a xmlns:b="urn:b">
  <c xmlns="urn:x">
    <d:e xmlns="urn:y"
      xmlns:d="urn:d"
      b:f="g">
      <i xmlns:b="urn:b2">
        <j b:k="l"/>
      </i>
    </d:e>
    <m n="o"/>
  </c>
</a>
```

Odpowiedź:

a nie należy do żadnej przestrzeni nazw
c należy do urn:x
e należy do urn:d
f należy do urn:b
i należy do urn:y
j należy do urn:y
k należy do urn:b2
m należy do urn:x
n **nie należy do żadnej!**

Które nazwy należą do jakiej przestrzeni nazw?

Przykład:

```
<a xmlns:b="urn:b">  
  <c xmlns="urn:x">  
    <d:e xmlns="urn:y"  
      xmlns:d="urn:d"  
      b:f="g">  
      <i xmlns:b="urn:b2">  
        <j b:k="l"/>  
      </i>  
    </d:e>  
    <m n="o"/>  
  </c>  
</a>
```

Odpowiedź:

a nie należy do żadnej przestrzeni nazw
c należy do urn:x
e należy do urn:d
f należy do urn:b
i należy do urn:y
j należy do urn:y
k należy do urn:b2
m należy do urn:x
n **nie należy do żadnej!**
p?

Które nazwy należą do jakiej przestrzeni nazw?

Przykład:

```
<a xmlns:b="urn:b">  
  <c xmlns="urn:x">  
    <d:e xmlns="urn:y"  
      xmlns:d="urn:d"  
      b:f="g">  
      <i xmlns:b="urn:b2">  
        <j b:k="l"/>  
      </i>  
    </d:e>  
    <m n="o"/>  
  </c>  
</a>
```

Odpowiedź:

a nie należy do żadnej przestrzeni nazw
c należy do urn:x
e należy do urn:d
f należy do urn:b
i należy do urn:y
j należy do urn:y
k należy do urn:b2
m należy do urn:x
n **nie należy do żadnej!**
p nie należy do żadnej przestrzeni nazw

Domyślnym kodowaniem XML-owym jest UTF-8.

Użycie w dokumentach XML-owych:

- odpowiedni ciąg bajtów,
- `&#kod`; — kod dziesiętny znaku,
- `&#xkod`; — kod szesnastkowy znaku.

Można używać polskich znaków w nazwach atrybutów i elementów.

Unicode to standard opisujący zestaw znaków mający w zamierzeniu objąć wszystkie znaki używane na świecie.

Unicode numeruje znaki, natomiast nie definiuje sposobów ich zapisu (kodowania).

	010	011	012	013	014	015	016	017
0	Ā	Ð	Ġ	Ī	Ĺ	Ŏ	Š	Ů
	0100	0110	0120	0130	0140	0150	0160	0170
1	ā	ð	ġ	ī	ĺ	ó	š	ů
	0101	0111	0121	0131	0141	0151	0161	0171
2	Ǻ	Ē	Ģ	Ĳ	ł	Œ	Ť	Ů
	0102	0112	0122	0132	0142	0152	0162	0172
3	ǻ	ē	ģ	ĳ	ń	œ	ť	ů
	0103	0113	0123	0133	0143	0153	0163	0173
4	Ą	Ě	Ĥ	Ĵ	ń	Ŕ	Ť	Ű
	0104	0114	0124	0134	0144	0154	0164	0174
5	ą	ě	ĥ	ĵ	Ń	ŕ	ť	ű
	0105	0115	0125	0135	0145	0155	0165	0175

Unicode to standard opisujący zestaw znaków mający w zamierzeniu objąć wszystkie znaki używane na świecie.

Unicode numeruje znaki, natomiast nie definiuje sposobów ich zapisu (kodowania).

	010	011	012	013	014	015	016	017
0	Ā	Đ	Ġ	Ī	Ĺ	Ŏ	Š	Ů
	0100	0110	0120	0130	0140	0150	0160	0170
1	ā	đ	ġ	ī	ĺ	ő	š	ů
	0101	0111	0121	0131	0141	0151	0161	0171
2	Ǻ	Ē	Ģ	Ĵ	Į	Œ	Ŧ	Ū
	0102	0112	0122	0132	0142	0152	0162	0172
3	ǻ	ē	ģ	ĵ	į	œ	ŧ	ū
	0103	0113	0123	0133	0143	0153	0163	0173
4	Ą	Ě	Ĥ	Ĵ	ń	Ŕ	Ť	Ŵ
	0104	0114	0124	0134	0144	0154	0164	0174
5	ą	ě	ĥ	ĵ	ņ	ŕ	ť	ŵ
	0105	0115	0125	0135	0145	0155	0165	0175

Najpopularniejsze metody kodowania:

- UTF-8 (pierwsze 128 — ASCII, znak zajmuje od 1 do 6 bajtów),
- UTF-16 (każdy znak to jedno lub dwa słowa 16-bitowe),
- UTF-32/UCS-4 (każdy znak to jedno słowo 32-bitowe).

XML a znaczenie fragmentów dokumentu

```
<ustawa data-wydania="2009-09-10">  
  <tytuł>o zniesieniu podatków</tytuł>  
  <artykuł nr="1">Wszystkie podatki zostają zniesione  
    z dniem 1 stycznia 2010 r.</artykuł>  
</ustawa>
```

XML a znaczenie fragmentów dokumentu

```
<ustawa data-wydania="2009-09-10">  
  <tytuł>o zniesieniu podatków</tytuł>  
  <artykuł nr="1">Wszystkie podatki zostają zniesione  
    z dniem 1 stycznia 2010 r.</artykuł>  
</ustawa>
```

```
<u45 d="2009-09-10">  
  <t>o zniesieniu podatków</t>  
  <a n="1">Wszystkie podatki zostają zniesione  
    z dniem 1 stycznia 2010 r.</a>  
</u45>
```

```
<ustawa data-wydania="2009-09-10">  
  <tytuł>o zniesieniu podatków</tytuł>  
  <artykuł nr="1">Wszystkie podatki zostają zniesione  
    z dniem 1 stycznia 2010 r.</artykuł>  
</ustawa>
```

```
<u45 d="2009-09-10">  
  <t>o zniesieniu podatków</t>  
  <a n="1">Wszystkie podatki zostają zniesione  
    z dniem 1 stycznia 2010 r.</a>  
</u45>
```

Sam zapis dokumentu w XML-u
nie nadaje jego fragmentom żadnego znaczenia!

Podsumowanie: XML w 10 punktach (wg W3C)

- 1 XML służy do budowania struktur danych,

Podsumowanie: XML w 10 punktach (wg W3C)

- 1 XML służy do budowania struktur danych,
- 2 XML w zapisie przypomina HTML,

Podsumowanie: XML w 10 punktach (wg W3C)

- 1 XML służy do budowania struktur danych,
- 2 XML w zapisie przypomina HTML,
- 3 XML jest tekstem, ale nie należy go czytać,

Podsumowanie: XML w 10 punktach (wg W3C)

- 1 XML służy do budowania struktur danych,
- 2 XML w zapisie przypomina HTML,
- 3 XML jest tekstem, ale nie należy go czytać,
- 4 XML jest nadmiarowy,

Podsumowanie: XML w 10 punktach (wg W3C)

- 1 XML służy do budowania struktur danych,
- 2 XML w zapisie przypomina HTML,
- 3 XML jest tekstem, ale nie należy go czytać,
- 4 XML jest nadmiarowy,
- 5 XML to rodzina technologii,

Podsumowanie: XML w 10 punktach (wg W3C)

- 1 XML służy do budowania struktur danych,
- 2 XML w zapisie przypomina HTML,
- 3 XML jest tekstem, ale nie należy go czytać,
- 4 XML jest nadmiarowy,
- 5 XML to rodzina technologii,
- 6 XML to już uznany standard,

Podsumowanie: XML w 10 punktach (wg W3C)

- 1 XML służy do budowania struktur danych,
- 2 XML w zapisie przypomina HTML,
- 3 XML jest tekstem, ale nie należy go czytać,
- 4 XML jest nadmiarowy,
- 5 XML to rodzina technologii,
- 6 XML to już uznany standard,
- 7 XML sprawia, że HTML może stać się XHTML-em,

Podsumowanie: XML w 10 punktach (wg W3C)

- 1 XML służy do budowania struktur danych,
- 2 XML w zapisie przypomina HTML,
- 3 XML jest tekstem, ale nie należy go czytać,
- 4 XML jest nadmiarowy,
- 5 XML to rodzina technologii,
- 6 XML to już uznany standard,
- 7 XML sprawia, że HTML może stać się XHTML-em,
- 8 XML ma budowę modułową,

Podsumowanie: XML w 10 punktach (wg W3C)

- 1 XML służy do budowania struktur danych,
- 2 XML w zapisie przypomina HTML,
- 3 XML jest tekstem, ale nie należy go czytać,
- 4 XML jest nadmiarowy,
- 5 XML to rodzina technologii,
- 6 XML to już uznany standard,
- 7 XML sprawia, że HTML może stać się XHTML-em,
- 8 XML ma budowę modułową,
- 9 XML jest podstawą RDF,

Podsumowanie: XML w 10 punktach (wg W3C)

- 1 XML służy do budowania struktur danych,
- 2 XML w zapisie przypomina HTML,
- 3 XML jest tekstem, ale nie należy go czytać,
- 4 XML jest nadmiarowy,
- 5 XML to rodzina technologii,
- 6 XML to już uznany standard,
- 7 XML sprawia, że HTML może stać się XHTML-em,
- 8 XML ma budowę modułową,
- 9 XML jest podstawą RDF,
- 10 XML nie jest licencjonowany, jest niezależny od dostawców IT i ma szerokie wsparcie.

Podsumowanie: XML w 10 punktach (wg W3C)

- 1 XML służy do budowania struktur danych,
- 2 XML w zapisie przypomina HTML,
- 3 XML jest tekstem, ale nie należy go czytać,
- 4 XML jest nadmiarowy,
- 5 XML to rodzina technologii,
- 6 XML to już uznany standard,
- 7 XML sprawia, że HTML może stać się XHTML-em,
- 8 XML ma budowę modułową,
- 9 XML jest podstawą RDF,
- 10 XML nie jest licencjonowany, jest niezależny od dostawców IT i ma szerokie wsparcie.

<http://www.w3.org/XML/1999/XML-in-10-points.html>