

## **DTD – Document Type Definition**

### **Dokument poprawnie sformułowany (well-formed)**

Dokument poprawnie sformułowany jest zgodny z ogólnymi zasadami składniowymi:

1. Każdy znacznik musi posiadać zamknięcie (wyjątek stanowią znaczniki puste).
2. Wartości poszczególnych atrybutów znacznika muszą być objęte cudzysłowem.
3. Elementy muszą być poprawnie zagnieżdżone.
4. Dokument musi posiadać element najwyższego poziomu.

### **DTD**

Mechanizm kontroli poprawności sformułowania dokumentu nie daje nam gwarancji iż dany dokument będzie zawierał poprawnie zapisane informacje.

Sprawdzana jest tu wyłącznie część techniczna składni języka. Taka kontrola nie zawsze jest wystarczająca. Tworząc dokumenty XML powinniśmy czuwać także nad spójnością danych.

Jednym z rozwiązań wspomagających to zadanie jest DTD (Document Type Definition). Jest to mechanizm w jak na dzisiejsze wymagania niedoskonały, ale nadal bardzo często używany przez programistów.

Zasadniczo DTD stanowi coś w rodzaju arkusza opisu dokumentu XML.

**Wprowadzenie DTD**

DTD możemy umieścić w pliku XML [Przykład 1] lub wyłączyć go do oddzielnego pliku [Przykład 2].

[Przykład 1]

```
<?xml version="1.0" encoding="ISO-8859-2"?>
<!DOCTYPE biblioteka
[
  <!ELEMENT biblioteka (ksiazki)>
  <!ELEMENT ksiazki (ksiazka+)>
  <!ELEMENT ksiazka (id, tytul, autor)>
  <!ELEMENT id (#PCDATA)>
  <!ELEMENT tytul (#PCDATA)>
  <!ELEMENT autor (#PCDATA)>
]>
<biblioteka>
  <ksiazki>
    <ksiazka>
      <id>1</id>
      <tytul>Zaślubiny Patyków</tytul>
      <autor>J.Carroll</autor>
    </ksiazka>
    <ksiazka>
      <id>2</id>
      <tytul>Zbrodnia i Kara</tytul>
      <autor>F.Dostojewski</autor>
    </ksiazka>
  </ksiazki>
</biblioteka>
```

[plik xml z wewnętrznym DTD]

[Przykład 2]

```
<?xml version="1.0" encoding="ISO-8859-2"?>
<!DOCTYPE biblioteka SYSTEM "2.dtd">
<biblioteka>
  <ksiazki>
    <ksiazka>
      <id>1</id>
      <tytul>Zaślubiny Patyków</tytul>
      <autor>J.Caroll</autor>
    </ksiazka>
    <ksiazka>
      <id>2</id>
      <tytul>Zbrodnia i Kara</tytul>
      <autor>F.Dostojewski</autor>
    </ksiazka>
  </ksiazki>
</biblioteka>
```

[odwołanie do zewnętrznego arkusza DTD]

```
<!ELEMENT biblioteka (ksiazki)>
<!ELEMENT ksiazki (ksiazka+)>
<!ELEMENT ksiazka (id, tytul, autor)>
<!ELEMENT id (#PCDATA)>
<!ELEMENT tytul (#PCDATA)>
<!ELEMENT autor (#PCDATA)>
```

[zewnętrzny arkusz DTD (plik z rozszerzeniem dtd)]

## Budowa DTD

### Zasady:

- każdy element oraz atrybut występujący w pliku XML musi zostać zadeklarowany w DTD
- deklaracja taka polega na dodaniu specjalnej definicji
  - <!ELEMENT...> - dla całego elementu
  - <!ATTLIST...> - dla atrybutów danego elementu

### Składnia:

#### Elementy

<!ELEMENT **ksiazka** ...> - nazwa definiowanego elementu

<!ELEMENT **ksiazka** [**zawartość**]> - zawartość elementu:

Możliwe wartości:

<b>EMPTY</b>	<!ELEMENT <b>ksiazka</b> <b>EMPTY</b> >
Element pusty – może zawierać jedynie atrybuty	
<b>ANY</b>	<!ELEMENT <b>ksiazka</b> <b>ANY</b> >
Dopuszczalna jest każda zawartość	
<b>(#PCDATA)</b>	<!ELEMENT <b>ksiazka</b> <b>(#PCDATA)</b> >
Zakładamy występowanie dowolnych łańcuchów tekstowych	
<b>(ELEMENTY)</b>	<!ELEMENT <b>ksiazka</b> <b>(tytuł, autor)</b> >
Dopuszczamy występowanie elementów potomnych z wyspecyfikowanej listy (elementy wyłącznie z poziomu bezpośrednio niższego).	
(element_1, element_2) definiujemy kolejność występowania elementów	
(element_1   element_2) kolejność nie ma znaczenia	
(#PCDATA, element) element o zawartości mieszanej	
(element_1, (element_2a   element_2b)) opcjonalnie element 2a lub 2b	

Ilość wystąpień danego elementu:

<code>&lt;!ELEMENT książka (tytuł)&gt;</code>	ilość wystąpień: 1
<code>&lt;!ELEMENT książka (tytuł?)&gt;</code>	ilość wystąpień: 0 lub 1
<code>&lt;!ELEMENT książka (tytuł*)&gt;</code>	ilość wystąpień: od 0 do nieskończoności
<code>&lt;!ELEMENT książka (tytuł+)&gt;</code>	ilość wystąpień: od 1 do nieskończoności

Przykładowe kombinacje:

<code>&lt;!ELEMENT książka (tytuł+, autor+, zdjęcie?)&gt;</code>
<code>&lt;!ELEMENT książka (tytuł+, autor+, recenzje*)&gt;</code>
<code>&lt;!ELEMENT książka (#PCDATA   tytuł)*&gt;</code>

## Atrybuty

<!ATTLIST **ksiazka** ...> - nazwa elementu dla którego opisujemy atrybuty

<!ATTLIST **ksiazka** **ISBN**> - nazwa atrybutu

<!ATTLIST **ksiazka** **ISBN** **CDATA**> - typ atrybutu

<b>CDATA</b>	<!ATTLIST <b>ksiazka</b> <b>ISBN</b> <b>CDATA</b> >
Atrybut może przyjąć dowolną wartość tekstową	
<b>(LISTA OPCJI)</b>	<!ATTLIST <b>ksiazka</b> <b>oprawa</b> ( <b>twarda</b>   <b>miekka</b> )>
Definiujemy listę wartości jakie może przyjmować dany atrybut	
<b>NMTOKEN (S)</b>	<!ATTLIST <b>ksiazka</b> <b>rok_wydania</b> <b>NMTOKEN</b> >
Wartość atrybutu ma być nazwą XML (lub listą takich nazw – NMTOKENS oddzielonych przecinkami)	
<b>ID</b>	<!ATTLIST <b>ksiazka</b> <b>ISBN</b> <b>ID</b> >
Wartość atrybutu ma być unikalna w całym dokumencie	
<b>IDREF (S)</b>	<!ATTLIST <b>ksiazka</b> <b>numer</b> <b>IDREF</b> >
Odwołanie do nazwy zadeklarowanej przy pomocy atrybutu ID	

<!ATTLIST **ksiazka** **ISBN** **CDATA** **REQUIRED**> - wystąpienie atrybutu

<b>"WARTOŚĆ"</b>	<!ATTLIST <b>ksiazka</b> <b>jezyk</b> <b>CDATA</b> <b>"Polski"</b> >
Wartość w cudzysłowie staje się wartością domyślną – z możliwością zmiany w dokumencie XML	
<b>#FIXED "WARTOŚĆ"</b>	<!ATTLIST <b>ksiazka</b> <b>jezyk</b> <b>CDATA</b> <b>#FIXED</b> <b>"PL"</b> >
Wartość w cudzysłowie staje się wartością domyślną – brak możliwości zmiany w dokumencie XML	
<b>#REQUIRED</b>	<!ATTLIST <b>ksiazka</b> <b>jezyk</b> <b>CDATA</b> <b>#REQUIRED</b> >
Atrybut obowiązkowy (wymagany)	
<b>#IMPLIED</b>	<!ATTLIST <b>ksiazka</b> <b>jezyk</b> <b>CDATA</b> <b>#IMPLIED</b> >
Atrybut nieobowiązkowy (opcjonalny)	

<http://www.stg.brown.edu/service/xmlvalid/>