

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="UTF-8"?>

<!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.Caroll</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="UTF-8"?>

<!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:

EMPTY

<!ELEMENT **ksiazka** **EMPTY**>

Element pusty – może zawierać jedynie atrybuty

ANY

<!ELEMENT **ksiazka** **ANY**>

Dopuszczalna jest każda zawartość

(#PCDATA)

<!ELEMENT **ksiazka** (**#PCDATA**)>

Zakładamy występowanie dowolnych łańcuchów tekstowych

(ELEMENTY)

<!ELEMENT **ksiazka** (**tytuł, autor**)>

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) może wystąpić jeden z elementów

(#PCDATA | element)* element o zawartości mieszanej

(element_1, (element_2a | element_2b)) opcjonalnie element 2a lub 2b

Ilość wystąpień danego elementu:

<!ELEMENT książka (tytuł)>	ilość wystąpień: 1
<!ELEMENT książka (tytuł?)>	ilość wystąpień: 0 lub 1
<!ELEMENT książka (tytuł*)>	ilość wystąpień: od 0 do nieskończoności
<!ELEMENT książka (tytuł+)>	ilość wystąpień: od 1 do nieskończoności

Przykładowe kombinacje:

<!ELEMENT książka (tytuł+, autor+, zdjęcie?)>
<!ELEMENT książka (tytuł+, autor+, recenzje*)>
<!ELEMENT książka (#PCDATA tytuł)*>

Atrybuty

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

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

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

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

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

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

Walidacja XML i DTD

Netbeans

VS Code + XML (Red Hat)

Notepad++ i plugin XML Tools (zewnętrzny DTD)

<https://www.xmlvalidation.com>