

Hipertekst i Hipermedia

Projekt

Temat:

MOJE HOBBY

Etap:

Etap	Punktacja [pkt]
HTML, Dokument XML, XML Schema, DTD	25
XSLT	15

Etap 1: HTML (11pkt), Dokument XML(1pkt), XML Schema (11pkt), DTD (2pkt)

HTML: (11 pkt)

Wymagania:

- zawartość strony zgodna z tematem projektu
- HTML5,
- strona responsywna (minimum to obsługa dwóch różnych wielkości ekranu) **(1,6pkt)**. Minimalne wymagania na responsywność:
 - viewport,
 - media queries (@media)
 - zmiana położenia menu
 - wykorzystanie viewport-width property
- walidacja HTML **(1pkt)** (<https://validator.w3.org/>)
- walidacja css **(0,5pkt)** (<http://jigsaw.w3.org/css-validator/>)
- układ strony:
 - podział strony na kilka elementów (nagłówek, menu, stopka, pole z treścią), **(1pkt)**
 - wykorzystanie znaczników semantycznych (header, footer, nav, figure ...) **(0,6pkt)**
- rozdzielenie treści na kilka plików (przynajmniej trzy) **(0,6pkt)**
- menu zawierające przynajmniej trzy opcje, a jedna z nich z dodatkowymi opcjami podrzędnymi; zaznaczanie wybranej opcji **(0,6pkt)**
- umieszczenie na stronie multimediiów:
 - grafika
 - galeria zdjęć (grafika rastrowa) (przynajmniej 5) ma być zorganizowana w postaci miniaturek, które można obejrzyć powiększone **(0,6pkt)**
 - prosta grafika wektorowa SVG, umieszczona w pliku HTML w znacznikach `<svg></svg>` **(0,5 pkt)**
 - animacja wykorzystująca mechanizm klatek kluczowych CSS3 (@keyframes) oraz efekt przejścia (transition) **(0,6pkt)**
- umieszczenie na stronie:
 - tabeli **(0,2pkt)**
 - odsyłaczy do innych stron internetowych **(0,3pkt)**
 - odsyłaczy (min 2) do wybranego miejsca w tekście lub do początku strony (wyświetlony tekst powinien być odpowiednio długi, aby była możliwość zademonstrowania tej opcji) **(0,4pkt)**
- style należy zdefiniować w oddzielnym arkuszu stylów, wykorzystać mechanizm CSS
 - różne style dla przynajmniej 4 selektorów (grup selektorów) **(0,6 pkt)**
 - klasy (przynajmniej 3) **(0,6pkt)**
 - identyfikator (przynajmniej 1) **(0,2pkt)**
 - wykorzystanie min 2 pseudoklas **(0,2pkt)**

- wykorzystanie pseudoelementu **(0,1pkt)**
- stworzenie prostej ankiety-formularza **(0,8pkt)**
 - przynajmniej 7 pól do wprowadzania danych (w tym lista rozwijana oraz radio button)
 - przynajmniej 5 różnych rodzajów pól umożliwiających wprowadzanie danych (w tym lista rozwijana oraz radio button),
 - wykorzystanie znacznika <label>
 - przyciski do czyszczenia zawartości formularza oraz wysyłania danych (w atrybucie *action* formularza nie używać mailto)
- dbałość o estetyczny wygląd strony

XML: (1pkt)

Wymagania:

- utworzyć plik w formacie XML zawierający **dane** związane z tematem projektu. Dane zawarte w dokumencie XML powinny odpowiadać danym prezentowanym na stronie HTML. Błędem jest tworzenie dokumentu XML zawierającego znaczniki odpowiedzialne za prezentację danych – odzwierciedlające rozmieszczenie danych na stworzonej w poprzednim punkcie stronie HTML. Plik XML należy tworzyć zwracając uwagę na umieszczane w nim dane, relacje pomiędzy nimi a nie na to jak były umieszczone na stronie HTML.

W pliku XML:

- nazwy znaczników, hierarchia mają być informacją o przechowywanych danych
- muszą istnieć co najmniej 4 poziomy zagłębienia nie licząc korzenia **(0,3pkt)**
- należy wykorzystać przynajmniej 6 różnych atrybutów **(0,3pkt)**
- dokument XML-owy ma zawierać przynajmniej 12 różnych nazw elementów **(0,2pkt)**
- trzeba umieścić dane dla przynajmniej trzech podelementów korzenia
- muszą znajdować się zdjęcia (przynajmniej 3) **(0,1pkt)**
- muszą znajdować się linki (przynajmniej 3) **(0,1pkt)**

XML SCHEMA: (11pkt)

Wymagania:

- Dla pliku XML, aby wymusić jego odpowiednią składnię, należy zaprojektować i utworzyć plik XML Schema.
- Plik XML musi być poprawny składniowo i semantycznie. Struktura pliku XML musi być zgodna z podaną w XML Schema. Do sprawdzenia poprawności należy użyć walidatora
- Należy również zwrócić uwagę na postać dokumentu, czyli sposób zapisu, stosowanie wcięć obrazujących strukturę danych, odpowiednie (adekwatne do zawartej w nich treści) nazywanie znaczników, atrybutów.

Wymagania szczegółowe:

W pliku XML Schema należy zadeklarować i wykorzystać:

- co najmniej 6 definicji globalnych typów złożonych **(1,6pkt)**
- przynajmniej 6 definicji globalnych typów prostych **(1,6pkt)**
- co najmniej 4 definicje lokalnych typów złożonych **(0,8pkt)**
- przynajmniej 4 definicje lokalnych typów prostych **(0,8pkt)**
- stosowanie różnych modeli wyboru (*sequence*, *choice*, *all*) i/lub mieszanego typu zawartości (*mixed*) **(0,3pkt)**
- przynajmniej jedna definicja grupy (elementów lub atrybutów) **(0,2pkt)**
- istnienie przynajmniej 4 poziomów zagłębienia w strukturze dokumentu xml **(0,4pkt)**
- deklaracja przynajmniej 6 atrybutów z czego przynajmniej 1 zdefiniowany globalnie i użyty przynajmniej 2 razy **(1,2pkt)**
- różnorodne deklaracje przynajmniej 12 różnych elementów (0,6pkt), w tym przynajmniej 5 z nich powinno zawierać podelementy(1pkt) **(1,6pkt)**
- stosowanie aspektów (ograniczeń na elementy i atrybuty)

- *length, minLength, maxLength, maxInclusive, minInclusive, maxExclusive, minExclusive, ...* (wybrane min 4 różne) **(0,4pkt)**
- *pattern* i *enumeration* **(0,6pkt)**
- wprowadzanie typów **(0,3pkt)**
 - *extension* - rozszerzenie istniejącego typu o dodatkowe **elementy**
- przynajmniej 3 odnośniki (ref) do elementów i/lub atrybutów (ma być odniesienie i do atrybutu i do elementu) **(0,6pkt)**
- użycie listy (*list*) (0,05pkt), należy określić długość listy (liczbę elementów listy) ((0,05pkt) oraz ograniczyć wartości, jakie mogą wystąpić na liście (0,05pkt), lista musi być również wykorzystana w pliku XML (0,05pkt) **(0,2pkt)**
- wykorzystanie kombinacji (*union*) oraz użycie elementu tego typu w pliku XML **(0,2pkt)**
- wykorzystanie przynajmniej 4 różnych typów wbudowanych **(0,2pkt)**
- walidowanie pliku
- w pliku XML przynajmniej 3 wypełnione podelementy korzenia

XML, DTD: (2pkt)

Wymagania:

- Dla pliku XML (wykorzystanego w zadaniu z XML Schema), aby wymusić jego odpowiednią składnię, należy zaprojektować i utworzyć plik DTD.
- Plik XML musi być poprawny składniowo i semantycznie. Struktura pliku XML musi być zgodna z podaną w DTD. Do sprawdzenia poprawności należy użyć walidatora umieszczonego na stronie enauczanie.

Wymagania szczegółowe:

W pliku DTD należy wykorzystać:

- deklaracje elementów, w deklaracji elementów wykorzystać zarówno model sekwencji jak i wyboru **(0,5pkt)**
- deklaracje przynajmniej 6 atrybutów **(0,5pkt)**
- w deklaracjach atrybutów wykorzystać:
 - typ wyliczeniowy **(0,3pkt)**
 - nadanie wartości domyślnej **(0,3pkt)**
 - zapewnienie wymagania wystąpienia atrybutu **(0,2pkt)**
- encję parametryczną przynajmniej 2 razy **(0,2 pkt)**
- niedozwolone jest używanie konstrukcji ANY

Wybrane przykładowe błędy występujące w schematach:

- tworzenie dokumentu XML zawierającego znaczniki odzwierciedlające sposób prezentacji danych na stronie np. <podstrona> <akapit></akapit></podstrona> a nie same dane **(-2pkt)**
- błędy walidacji XML(plik się nie waliduje) **(do -10pkt)**
- trywialna definicja typu prostego (np. typ prosty, który jest zwykłym typem string) (do **-2pkt)**
- powtarzanie definicji typów (wielokrotne definiowanie takich samych typów) (do **-2pkt)**
- wykorzystanie anyType **(do -10pkt)**
- nieznacznie przerobiony, wygenerowany plik xsd **(do -10pkt)**
- niepoprawne definiowanie elementów, atrybutów, struktury **(do -6pkt)**
 - np. zamiast używać maxOccurs=4, czterokrotne deklaracje takiego samego elementu
- brak znaczników przechowujących zdjęcia **(-0,5pkt)**
- brak znaczników przechowujących linki **(-0,5pkt)**
- brak w pliku XML przynajmniej 3 wypełnionych podelementów korzenia **(-2pkt)**
- wykorzystanie ANY w DTD **(-1pkt)**

Uwaga

- Ostateczna liczba punktów za projekt jest uzależniona od odpowiedzi udzielanych podczas oddawania projektu, orientacji w projekcie i obowiązujących zagadnieniach.
- Podczas oddawania projektu **kod** ma być **pozbawiony** wszelkich **komentarzy**

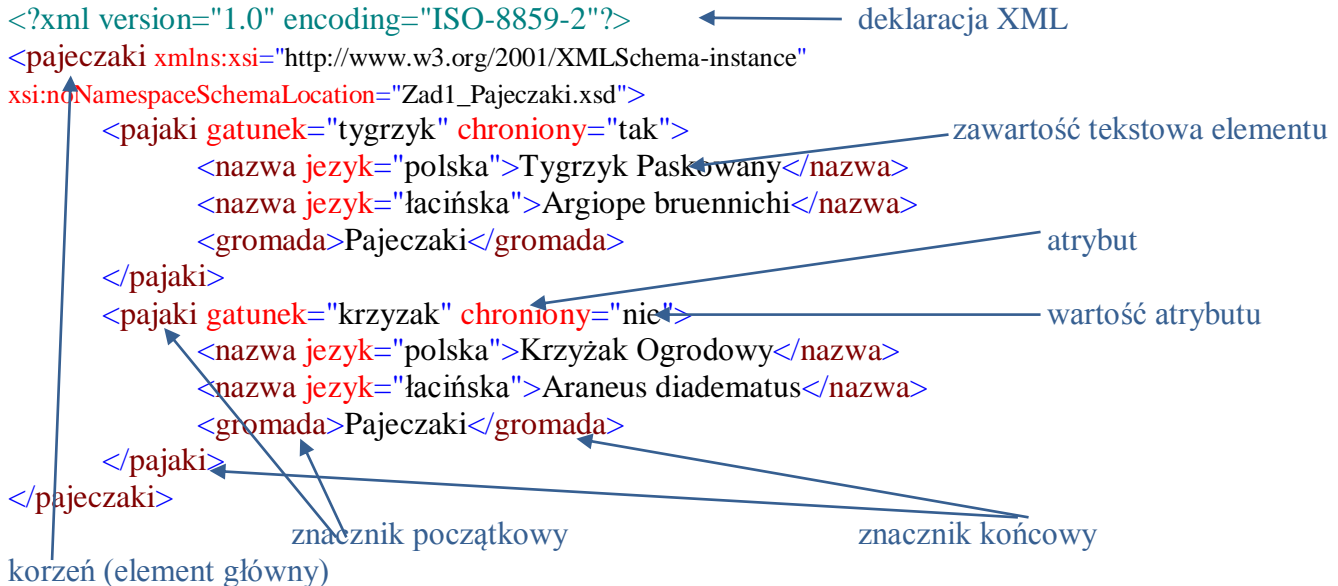
Oddawanie projektów

- każda osoba ma wyznaczony na odbiór projektu termin: dzień, godzinę, salę oraz prowadzącego. Odbiór projektu odbywa się tylko w wyznaczonym terminie.
- nie ma możliwości poprawiania oddanych projektów

- XML, DTD i XML Schema - krótka ściągą ☺

XML

- wszystkie niepuste elementy muszą mieć znacznik początkowy i końcowy
- elementy mogą być zagnieżdżone, nie mogą na siebie zachodzić
- rozróżnianie dużych i małych liter

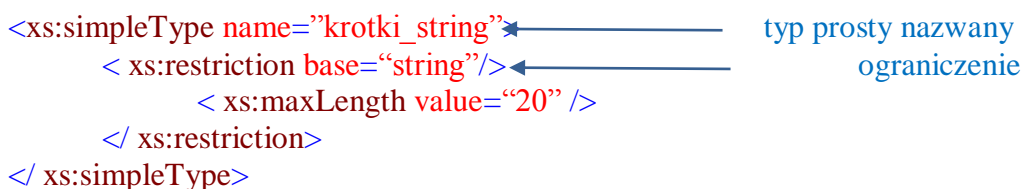


XML Schema

Jeśli chcemy stworzyć:

- tylko element z zawartością tekstową
 - typ prosty
- element z podelementami
 - typ złożony
- element z podelementami i atrybutami
 - typ złożony
- element z zawartością mieszaną (podelementy i tekst)
 - typ złożony z atrybutem `mixed=true`
- element z atrybutami
 - typ złożony
- element z atrybutami i zawartością prostą
 - typ złożony z `simpleContent`

1) Definicja typu prostego nazwanego (globalnego)



2) Definicja elementu

```
<xs:element name="pajaki" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="nazwa" maxOccurs="unbounded">
        <xs:complexType>
          <xs:attribute name="jezyk" type="xs:string" />
        </xs:complexType>
      </xs:element>
      <xs:element name="gromada" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="gatunek" type="xs:string" />
    <xs:attribute name="chroniony" type="xs:string" />
  </xs:complexType>
</xs:element>
```

3) Wyliczenia - lista predefiniowanych wartości

```
<xs:simpleType name="nazwa_typu">
  <xs:restriction base="string">
    <xs:enumeration value="wartosc1" />
    <xs:enumeration value="wartosc2" />
    <xs:enumeration value="wartosc3" />
  </xs:restriction>
</xs:simpleType>
```

4) SimpleContent

Gdy stworzymy pochodny typ złożony na podstawie typu prostego lub innego typu złożonego o zawartości prostej. Można w ten sposób dodać atrybuty do typu bazowego.

```
<xs:complexType name="nazwa_typu">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="nazwa_atrybutu" type="xs:string"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

5) Odniesienia do elementu

```
<xs:element name="data" type="xs:date"/>
```

globalna definicja elementu

```
<xs:element ref="data" minOccurs="0"/>
```

odniesienie do elementu zdefiniowanego globalnie

DTD

6) Deklaracja elementu

```
<!ELEMENT nazwa_elementu typ_zawartosci_elementu>
```

7) Wskaźniki liczby wystąpień

- ? 0 lub 1 raz
- + 1 lub dowolnie wiele razy
- * 0 lub dowolnie wiele razy

8) Deklaracja atrybutu

<!ATTLIST nazwa-elementu

nazwa-atrybutu1 typ-zawartości-atrybutu1 opis1

nazwa-atrybutu2 typ-zawartości-atrybutu2 opis2

...>

Typ zawartości atrybutu

CDATA dowolny tekst,

ID nazwa unikalna w całym dokumencie, dla danego typu elementu tylko jeden atrybut może być zadeklarowany jako ID,

IDREF nazwa występująca gdzieś w dokumencie jako wartość typu ID

Opis: określa czy atrybut jest wymagany i jaką ma wartość domyślną

#REQUIRED atrybut wymagany, #IMPLIED atrybut opcjonalny,

"wartość-domyślna" wartość domyślna atrybutu #FIXED "wartość" wartość ustalona.

9) Deklaracja encji

- wewnętrznych

<!ENTITY nazwa_encji 'reprezentowany tekst'>

- zewnętrznych

<!ENTITY nazwa_encji SYSTEM 'nazwa.pliku'>

- parametrów

<!ENTITY % nazwa_encji 'reprezentowany tekst'>