

## Podstawy HTML i arkusze stylów CSS (część 1)

- Przypomnienie podstaw języka HTML,
- Przypomnienie ogólnej struktury języka HTML, podstawowych znaczników, atrybutów,
- Sposób rozmieszczania elementów w przeglądarce,
- Różnice w wyglądzie dokumentów HTML dla różnych przeglądarek,
- Widok źródła dokumentu,
- Walidacja dokumentu HTML.

### Cel główny:

- Wykonanie prostego, walidowalnego dokumentu HTML (treść w postaci piosenki, wiersza, itp.).

### Cele cząstkowe:

- Przypomnienie podstawowej struktury języka HTML,
- Przypomnienie podstawowych znaczników HTML,
- Zaznajomienie się z różnicami w wyglądzie w różnych przeglądarkach,
- „Podejrzenie” źródła dowolnego dokumentu HTML,
- Walidacja własnego dokumentu HTML.

### Zadania do wykonania

- Przypomnieć sobie dowolny utwór wierszowany i zapisać go jako dokument HTML:
  - określić autora, tytuł, wyróżnić oba elementy,
  - wypisać 2 zwrotki,
  - użyć kilku różnych znaczników, do opisu struktury,
  - użyć kilku różnych znaczników, atrybutów do opisu wyglądu,
  - wstawić plik graficzny (np. zdjęcie autora lub zrobić tło).
- Zapoznać się z podstawową strukturą dokumentów HTML:
  - zwrócić uwagę na definicję DOCTYPE,
  - ustawić różne kodowania znaków (iso-8859-2, utf-8), i kodowanie pliku (ansi, utf-8), wybrać wszystkie możliwe kombinacje,
- Obejrzeć dokument w różnych przeglądarkach,
  - obejrzeć różnice w wyglądzie w różnych przeglądarkach,
  - obejrzeć i porównać źródła z oryginałem,
- Dokonać walidacji dokumentu HTML:
  - w przypadku błędów poprawić,
  - w przypadku ostrzeżeń, określić ich istotność i poprawić.

### Zawartość sprawozdania z ćwiczenia

- Protokół powinien zawierać:
  - opis różnic jakie pojawiły się w dostępnych przeglądarkach (przynajmniej w 3 różnych),
  - opis błędów i ostrzeżeń jakie pojawiły się podczas walidacji wraz ze sposobem ich poprawienia,

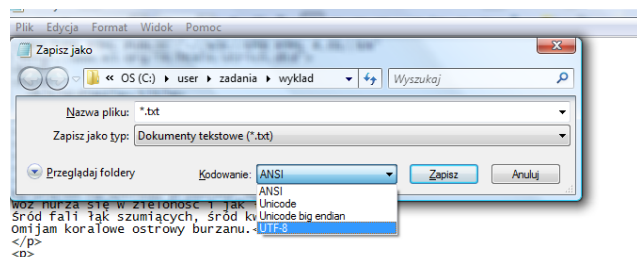
- opisać różnice w kodowaniach znaków i plików, i ich wpływ na wygląd w przeglądarce.
- opisać elementy służące do zmiany wyglądu stron www (kolory, wielkości, czcionki, itp.)
- UWAGA!!! Protokół należy oddać po całej serii ćwiczeń związanych z CSS (8 ćwiczeń).

## Źródła wiedzy

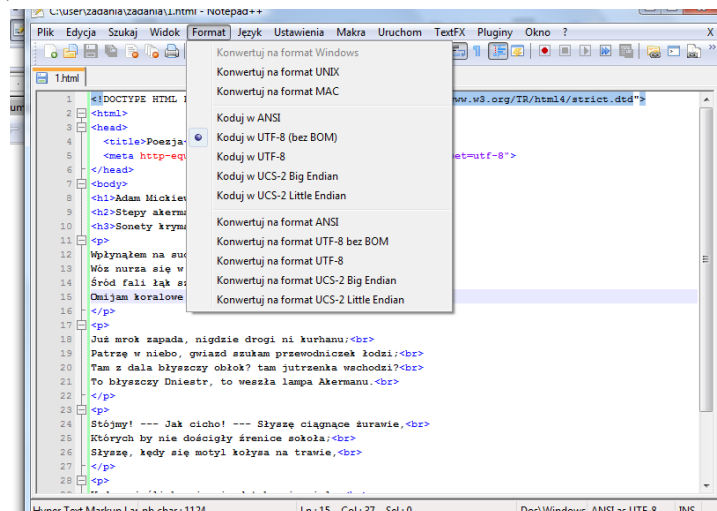
- R. Berezowski, Metajęzyki, wykład multimedialny,
- <http://validator.w3.org/>,
- <http://www.kurshtml.boo.pl/>
- <http://www.w3.org/>
- <http://www.w3schools.com/>

## Porady, wskazówki i podpowiedzi

- Definicja DOCTYPE:
  - `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">`
- Ustawienie kodowania znaków:
  - `<meta http-equiv="Content-Type" content="text/html; charset=utf-8">`
- Ustawienie kodowania plików:
  - Notatnik (od wersji windows XP), wybrać opcję zapisz jako i wybrać kodowanie:



- Notepad++



- Znaczniki do opisu struktury:
  - h1, h2, p, ul, ol, li,
- Znaczniki do opisu wyglądu:
  - font, b, i, u,
- Walidatory:
  - <http://validator.w3.org/>
  - <http://validator.aborla.net/index.php5?lang=pl>
  - <http://cgi.w3.org/cgi-bin/tidy>

## Podstawy HTML i arkusze stylów CSS (część 2)

- Elementy inline, bloki w HTML,
- Odnośniki HTML.
- Poznanie ogólnej struktury arkusza stylów CSS,
- Sposobu umieszczania stylów w HTML,
- Różnice w wyglądzie stron www dla różnych przeglądarek,
- Walidacja dokumentu HTML i arkusza stylów CSS.

### Cel główny:

- Wykonanie prostego, walidowalnego dokumentu HTML (treści w postaci kilku piosenek, wierszy, wykonanych w części 1 ćwiczenia),
- Zmiana wyglądu przy użyciu stylów inline i wewnętrznych .

### Cele cząstkowe:

- Przypomnienie sposobu deklaracji odnośników HTML,
- Określenie jakie właściwości obsługuje dana przeglądarka,
- Dołączanie stylów inline, wewnętrznych,
- Przesłanie właściwości CSS.

### Zadania do wykonania

- Zapisać kilka utworów w postaci dokumentu HTML:
  - określić autorów, tytuły, wyróżnić te elementy,
  - wypisać co najmniej po 2 zwrotki,
  - wstawić spis treści (w postaci listy) na początku strony z odnośnikami do poszczególnych utworów,
  - dołączyć definicję DOCTYPE,
  - ustawić kodowanie dla polskich znaków i kodowanie pliku (utf-8),
- Dołączyć style inline:
  - zmienić znaczniki font na odpowiadające im atrybuty style,
    - zamienić wszystkie możliwe atrybuty znacznika font,
  - wyrazy w spisie treści zamienić na „kapitałiki”,
- Dołączyć style wewnętrzne:
  - wszystkich autorów wypisać dużymi literami,
  - wszystkie tytuły podkreślić,
  - dołączyć tło w postaci obrazka,
- Wyszukać w dokumentacji jakie selektory CSS obsługuje używana przeglądarka internetowa,
- Dokonać walidacji dokumentu HTML oraz arkusza stylów CSS:
  - w przypadku błędów poprawić,
  - w przypadku ostrzeżeń, określić ich istotność i poprawić.

### Zawartość sprawozdania z ćwiczenia

- Protokół powinien zawierać:
  - opis różnic jakie pojawiły się w dostępnych przeglądarkach po zastosowaniu

- arkusza stylów (przynajmniej w 3 różnych),
- opis błędów i ostrzeżeń jakie pojawiły się podczas walidacji wraz ze sposobem ich poprawienia,
- właściwości, których użycie tworzy różne efekty w różnych przeglądarkach,
- które style (inline, wewnętrzne) są ważniejsze.

## **Źródła wiedzy**

- R. Berezowski, Metajęzyki, wykład multimedialny,
- <http://validator.w3.org/>,
- <http://www.kurshtml.boo.pl/>
- <http://www.w3.org/>
- <http://www.w3schools.com/>
- <http://jigsaw.w3.org/css-validator/>

## **Porady, wskazówki i podpowiedzi**

- Definicja DOCTYPE:
  - `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">`
- Ustawienie kodowania znaków:
  - `<meta http-equiv="Content-Type" content="text/html; charset=utf-8">`
- Przykład odnośnika do tego samego dokumentu:
  - odnośnik  
`<a href="#utwor1">Tytuł utworu</a>`
  - cel odnośnika  
`<a name="utwor1"/>`
- Dołączanie stylów inline – zamiana znacznika font:
  - `<h1><font color="red">`  
zamiana na `<h1 style="color: red">`
  - `<h2><font size="18px">`  
zamiana na `<h2 style="font-size: 18px">`
  - `<p><font face="arial">`  
zamiana na `<p style="font-family: arial">`
- Dołączanie stylów wewnętrznych:
  - `<head>`
  - `<style>`
  - `h1 {text-transform: uppercase;}`
  - `h2 {text-decoration: underline;}`
  - `ul {font-style: small-caps;}`
  - `</style>`
  - `</head>`
- Obsługa CSS przez przeglądarki:

- Firefox  
[https://developer.mozilla.org/pl/Firefox\\_3.5\\_dla\\_programist%C3%B3w](https://developer.mozilla.org/pl/Firefox_3.5_dla_programist%C3%B3w)
- Internet Explorer 8  
[http://www.microsoft.com/poland/centrumprasowe/prasa/09\\_01/07.msp](http://www.microsoft.com/poland/centrumprasowe/prasa/09_01/07.msp)
- Opera
  - <http://www.opera.com/docs/specs/opera95/>
- Walidatory HTML:
  - <http://validator.w3.org/>
  - <http://validator.aborla.net/index.php5?lang=pl>
  - <http://cgi.w3.org/cgi-bin/tidy>
- Walidatory CSS:
  - <http://jigsaw.w3.org/css-validator/>
  - <http://www.htmlhelp.com/tools/csscheck/>

## Podstawy HTML i arkusze stylów CSS (część 3)

- Tabele w HTML,
- Inline, wewnętrzne i zewnętrzne arkusze stylów CSS,
- Przyciski CSS,
- Walidacja dokumentu HTML i arkusz stylów CSS.

### Cel główny:

- Wykonanie prostej, walidowalnego strony www (kilka dokumentów o treści w postaci piosenek, wierszy, wykonanych w części 2 ćwiczenia),
- Zmiana wyglądu przy użyciu stylów wewnętrznych i zewnętrznych.

### Cele cząstkowe:

- Przypomnienie sposobu deklaracji tabel HTML,
- Dołączanie stylów wewnętrznych i zewnętrznych,
- Przesłanianie właściwości CSS,
- Utworzenie prostej pseudografiki przy użyciu arkusza stylów CSS (przycisk).

### Zadania do wykonania

- Zapisać kilka utworów w postaci kilku dokumentu HTML:
  - każdy dokument powinien zawierać autora, tytuł, wyróżnić te elementy stosując style wewnętrzne,
- Utworzyć dokument HTML zawierający spis utworów (w postaci tabeli) z odnośnikami do poszczególnych utworów - dokumentów,
- Dołączyć style zewnętrzne:
  - zmienić używane style wewnętrzne w dokumentach na jeden arkusz stylów umieszczony w pliku zewnętrznym,
  - dopisać w każdym pliku linie dołączającą zewnętrzny arkusz ze stylami,
- Dołączyć style wewnętrzne:
  - sformatować znaczniki table, tr, td, th występujące w spisie treści za pomocą stylów wewnętrznych,
- Na każdej podstronie dołączyć link powrotny do spisu treści,
- Zmienić wygląd odnośników by wyglądały jak przyciski,
- Dokonać walidacji strony www oraz arkusza stylów CSS:
  - w przypadku błędów poprawić,
  - w przypadku ostrzeżeń, określić ich istotność i poprawić.

### Zawartość sprawozdania z ćwiczenia

- Protokół powinien zawierać:
  - opis różnic jakie pojawiły się w dostępnych przeglądarkach po zastosowaniu arkusza stylów (przynajmniej w 3 różnych),
  - opis błędów i ostrzeżeń jakie pojawiły się podczas walidacji wraz ze sposobem ich poprawienia,
  - właściwości, których użycie tworzy różne efekty w różnych przeglądarkach,
  - ustalić, które style (inline, wewnętrzne, zewnętrzne) są ważniejsze.

- podać dwa różne sposoby utworzenia wyglądu odnośnika w postaci przycisku.

## Źródła wiedzy

- R. Berezowski, Metajęzyki, wykład multimedialny,
- <http://validator.w3.org/>,
- <http://www.kurshtml.boo.pl/>
- <http://www.w3.org/>
- <http://www.w3schools.com/>
- <http://jigsaw.w3.org/css-validator/>

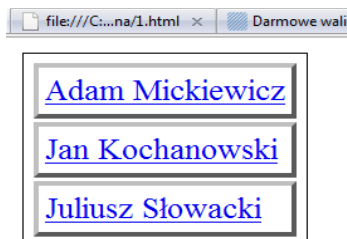
## Porady, wskazówki i podpowiedzi

- Definicja DOCTYPE:
  - `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">`
- Ustawienie kodowania znaków:
  - `<meta http-equiv="Content-Type" content="text/html; charset=utf-8">`
- Dołączanie stylów zewnętrznych:
  - `<link rel="stylesheet" href="plik.css" type="text/css">`
- Przykładowy wygląd pliku plik.css:
  - `h1 {text-transform: uppercase;}`
  - `h2 {text-decoration: underline;}`
  - `ul {font-style: small-caps;}`
- Przykładowy wygląd dokumentu i stylów wewnętrznych tworzących przycisk:

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/
2 <html>
3 <head>
4 <style>
5     table {padding: 3px; border: 1px solid black}
6     td {border: 3px outset gray; padding: 3px;}
7 </style>
8 <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
9 </head>
10 <body>
11 <table>
12 <tr><td><a href="am.html">Adam Mickiewicz</a></td></tr>
13 <tr><td><a href="jk.html">Jan Kochanowski</a></td></tr>
14 <tr><td><a href="js.html">Juliusz Słowacki</a></td></tr>
15 </table>
16 </body>
17 </html>
```

- Przykładowy wygląd „przycisków” w przeglądarce:





- Walidatory HTML:
  - <http://validator.w3.org/>
  - <http://validator.aborla.net/index.php5?lang=pl>
  - <http://cgi.w3.org/cgi-bin/tidy>
- Walidatory CSS:
  - <http://jigsaw.w3.org/css-validator/>
  - <http://www.htmlhelp.com/tools/csscheck/>

## Podstawy HTML i arkusze stylów CSS (część 4)

- Pseudoklasy hover, visited, active,
- Pseudoelementy first-line, first-letter,
- Inline, wewnętrzne i zewnętrzne arkusze stylów CSS,
- Przyciski CSS,
- Walidacja dokumentu HTML i arkusz stylów CSS.

### Cel główny:

- Wykonanie prostej, walidowalnego strony www (kilka dokumentów o treści w postaci piosenek, wierszy, wykonanych w części 3 ćwiczenia),
- Demonstracja możliwości pseudoklas i pseudoelementów.

### Cele cząstkowe:

- Definiowanie wbudowanych pseudoklas i pseudoelementów,
- Dołączanie stylów wewnętrznych i zewnętrznych,
- Przesłanie właściwości CSS,
- Utworzenie prostej ruchomej pseudografiki z odnośnikami przy użyciu arkusza stylów CSS (przycisk wciśnięty, wyciśnięty).

### Zadania do wykonania

- Wykorzystując dokument zbudowany w części trzeciej wykonać:
  - wykorzystując pseudoklasy hover, visited i active zrealizować linki umieszczone w tabeli jako przyciski tak by po trafieniu kursorem myszki przycisk zobaczył efekt „wciśnięcia” przycisku,
  - odpowiednie zapisy umieścić w stylach wewnętrznych,
  - odnośniki powrotu zrealizować w odwrotnym stylu (po trafieniu myszką zostają wyciśnięte) i umieścić w pliku zewnętrznym, nadać im przyjazną wartość „width”.
- Nadać specjalny format dla pierwszych liter w każdej zwrotce,
- Po każdym utworze dołączyć krótki opis, zawierający informację o autorze. Pierwszy wiersz w tej informacji powinien być pogrubiony,
- Dokonać walidacji strony www oraz arkusza stylów CSS:
  - w przypadku błędów poprawić,
  - w przypadku ostrzeżeń, określić ich istotność i poprawić.

### Zawartość sprawozdania z ćwiczenia

- Protokół powinien zawierać:
  - opis różnic jakie pojawiły się w dostępnych przeglądarkach po zastosowaniu arkusza stylów (przynajmniej w 3 różnych),
  - opis błędów i ostrzeżeń jakie pojawiły się podczas walidacji wraz ze sposobem ich poprawienia,
  - właściwości, których użycie tworzy różne efekty w różnych przeglądarkach,

## Źródła wiedzy

- R. Berezowski, Metajęzyki, wykład multimedialny,
- <http://validator.w3.org/>,
- <http://www.kurshtml.boo.pl/>
- <http://www.w3.org/>
- <http://www.w3schools.com/>
- <http://jigsaw.w3.org/css-validator/>

## Porady, wskazówki i podpowiedzi

- Definicja DOCTYPE:
  - `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">`
- Ustawienie kodowania znaków:
  - `<meta http-equiv="Content-Type" content="text/html; charset=utf-8">`
- Dołączanie stylów zewnętrznych:
  - `<link rel="stylesheet" href="plik.css" type="text/css">`
- Tworzenie odnośników – przycisków:
  - dodać do znacznika a właściwość `display: block`;
- Przykładowy wygląd dokumentu i stylów wewnętrznych tworzących „dynamiczny” przycisk:

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w
2 <html>
3 <head>
4 <style>
5 table {padding: 3px; border: 1px solid black}
6 a {border: 3px outset gray; padding: 1px; display: block}
7 a:hover {border-style: inset;}
8 </style>
9 <meta http-equiv="Content-Type" content="text/html; charset=u
10 </head>
11 <body>
12 <table>
13 <tr><td><a href="am.html">Adam Mickiewicz</a></td></tr>
14 <tr><td><a href="jk.html">Jan Kochanowski</a></td></tr>
15 <tr><td><a href="js.html">Juliusz Słowacki</a></td></tr>
16 </table>
17 </body>
18 </html>
```

- Przykładowy wygląd „przycisków” w przeglądarce:



- Przykładowy wygląd „przycisku” powrotu w przeglądarce:



- Pseudoklasa first-line:
  - `div:first-line {font-weight: bold;}`
- Pseudoklasa first-letter:
  - `p:first-letter {font-size: 1.5em; color: blue; text-transform: uppercase}`
- Walidatory HTML:
  - <http://validator.w3.org/>
  - <http://validator.aborla.net/index.php5?lang=pl>
  - <http://cgi.w3.org/cgi-bin/tidy>
- Walidatory CSS:
  - <http://jigsaw.w3.org/css-validator/>
  - <http://www.htmlhelp.com/tools/csscheck/>



## Podstawy HTML i arkusze stylów CSS (część 5)

- Selektory id i class
- Inline, wewnętrzne i zewnętrzne arkusze stylów CSS,
- Listy HTML,
- Walidacja dokumentu HTML i arkusz stylów CSS.

### Cel główny:

- Wykonanie prostej, walidowalnego strony www (kilka dokumentów o treści w postaci piosenek, wierszy, wykonanych w części 3 ćwiczenia),
- Demonstracja id i class na podstawie spisów treści o różnych formatach.

### Cele cząstkowe:

- Definiowanie stylów dla grup elementów,
- Definiowanie stylów dla wybranych elementów,
- Dołączanie stylów wewnętrznych i zewnętrznych,
- Przesłanie właściwości CSS,
- Tworzenie różnych formatów dla tych samych struktur.

### Zadania do wykonania

- Wykorzystując dokument zbudowany w części trzeciej wykonać:
  - dwa spisy treści w postaci list(odnośniki do utworów) umieszczone jeden obok drugiego,
  - kolejność utworów w spisie z lewej strony jest wg nazwisk autorów, a z prawej strony wg ich imion,
  - odnośniki w spisie z lewej strony mają być formatowane wg klasy naz,
  - odnośniki w spisie z prawej strony mają być formatowane wg klasy im,
  - strony z wierszami należy sformatować z użyciem klasy wiersz,
  - strony z piosenkami należy sformatować z użyciem klasy piosenka,
- Tło oraz kolory w dokumencie ze spisem treści ma być formatowane według identyfikatora spis,
- Dokonać walidacji strony www oraz arkusza stylów CSS:
  - w przypadku błędów poprawić,
  - w przypadku ostrzeżeń, określić ich istotność i poprawić.
- Zamienić klasy naz i im na identyfikatory o tych samych nazwach. Dokonać walidacji, oprawić ewentualne błędy

### Zawartość sprawozdania z ćwiczenia

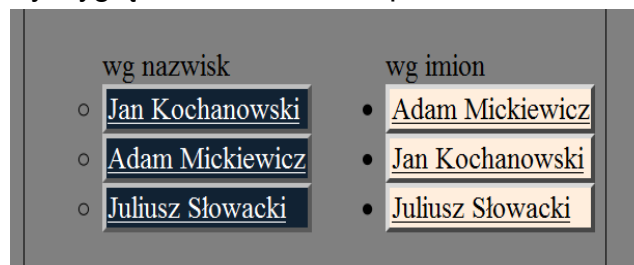
- Protokół powinien zawierać:
  - opis różnic jakie pojawiły się w dostępnych przeglądarkach po zastosowaniu arkusza stylów (przynajmniej w 3 różnych),
  - opis błędów i ostrzeżeń jakie pojawiły się podczas walidacji wraz ze sposobem ich poprawienia,
  - właściwości, których użycie tworzy różne efekty w różnych przeglądarkach,

## Źródła wiedzy

- R. Berezowski, Metajęzyki, wykład multimedialny,
- <http://validator.w3.org/>,
- <http://www.kurshtml.boo.pl/>
- <http://www.w3.org/>
- <http://www.w3schools.com/>
- <http://jigsaw.w3.org/css-validator/>

## Porady, wskazówki i podpowiedzi

- Definicja DOCTYPE:
  - `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">`
- Ustawienie kodowania znaków:
  - `<meta http-equiv="Content-Type" content="text/html; charset=utf-8">`
- Dołączanie stylów zewnętrznych:
  - `<link rel="stylesheet" href="plik.css" type="text/css">`
- Deklaracja klasy selektorów:
  - klasa naz     `ol.naz {list-style-type: circle}`
  - klasa im      `ol.im {list-style-type: disc}`
- Odwołanie do klasy:
  - klasa naz     `<ol class="naz">`
  - klasa im      `<ol class="im">`
- Aby spisy znajdowały się obok siebie można umieścić je w tabeli,
- Deklaracja identyfikatora spis:
  - `#spis {color:black; background-color: gray}`
- Odwołanie do identyfikatora spis:
  - `<body id="spis">`
- W przypadku użycia identyfikatora aby wskazać wszystkie odnośniki należy dołączyć go do elementu nadrzędnego zawierającego te odnośniki (np. `<ol>`),
- Przykładowy wygląd dokumentu ze spisem treści:



- Przykładowy wygląd arkusza stylów:

```
ol.naz {list-style-type: circle}
ol.im {list-style-type: disc}
ol.naz a {color: white; background-color: #112233}
ol.im a {color: black; background-color: #ffeedd}
#spis {color:black; background-color: gray}
```

- Walidatory HTML:
  - <http://validator.w3.org/>
  - <http://validator.aborla.net/index.php5?lang=pl>
  - <http://cgi.w3.org/cgi-bin/tidy>
- Walidatory CSS:
  - <http://jigsaw.w3.org/css-validator/>
  - <http://www.htmlhelp.com/tools/csscheck/>

## **Podstawy HTML i arkusze stylów CSS (część 6),**

- Ramki HTML
- Inline, wewnętrzne i zewnętrzne arkusze stylów CSS,
- Walidacja dokumentu HTML i arkusz stylów CSS.

### **Cel główny:**

- Wykonanie prostej, walidowalnej strony www (kilka dokumentów o treści w postaci piosenek, wierszy, wykonanych w części 3 ćwiczenia),
- Utworzenie strony www – szablonu zawierającego menu, nagłówek i treść.

### **Cele cząstkowe:**

- Logiczny podział na elementy stałe (spis utworów, nagłówek) i zmienne (utwory),
- Dołączanie stylów wewnętrznych i zewnętrznych,
- Dynamiczne ładowanie statycznych dokumentów do treści.

### **Zadania do wykonania**

- Wykorzystując dokument zbudowany w części trzeciej wykonać:
  - spis treści (menu) w postaci listy odnośników do utworów umieszczone jeden pod drugim,
  - odnośniki w menu mają być formatowane wg klasy menu,
  - strony z utworami należy sformatować z użyciem klasy utwor, tak by były samodzielnymi dokumentami,
- Utworzyć stronę z nagłówkiem (ogólne informacje o zawartości strony).
- Utworzyć stronę z podziałem na 3 ramki (frameset, frame),
  - menu – z lewej strony,
  - top – na górze,
  - main – pozostałe miejsce.
- Poprawić odnośniki w menu tak by otwierały się w main,
- Dokonać walidacji strony www oraz arkusza stylów CSS:
  - w przypadku błędów poprawić,
  - w przypadku ostrzeżeń, określić ich istotność i poprawić.

### **Zawartość sprawozdania z ćwiczenia**

- Protokół powinien zawierać:
  - opis różnic jakie pojawiły się w dostępnych przeglądarkach po zastosowaniu arkusza stylów (przynajmniej w 3 różnych),
  - opis błędów i ostrzeżeń jakie pojawiły się podczas walidacji wraz ze sposobem ich poprawienia,
  - właściwości, których użycie tworzy różne efekty w różnych przeglądarkach,

### **Źródła wiedzy**

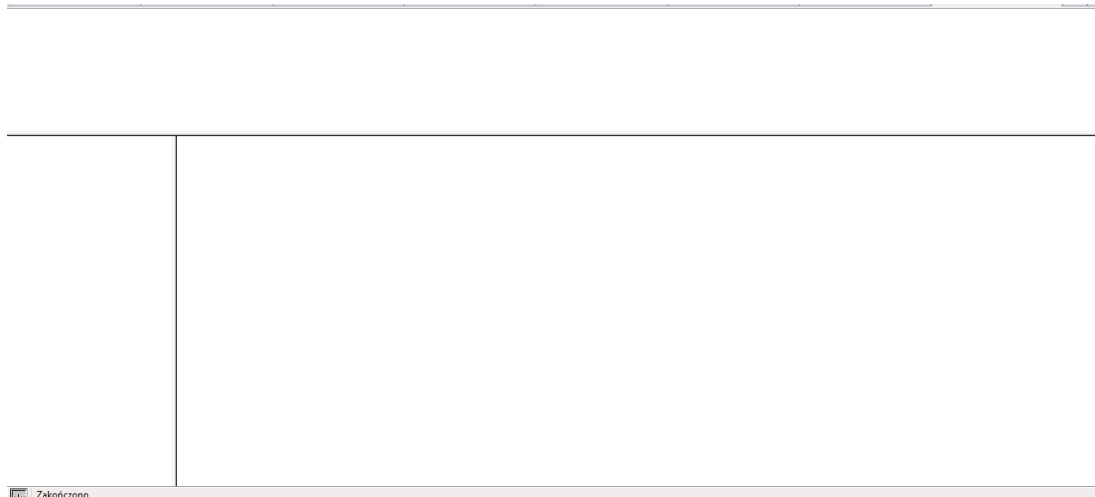
- R. Berezowski, Metajęzyki, wykład multimedialny,
- <http://validator.w3.org/>,
- <http://www.kurshtml.boo.pl/>



- <http://www.w3.org/>
- <http://www.w3schools.com/>
- <http://jigsaw.w3.org/css-validator/>

## Porady, wskazówki i podpowiedzi

- Definicja DOCTYPE:
  - `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">`
- Ustawienie kodowania znaków:
  - `<meta http-equiv="Content-Type" content="text/html; charset=utf-8">`
- Dołączanie stylów zewnętrznych:
  - `<link rel="stylesheet" href="plik.css" type="text/css">`
- Deklaracja klasy selektorów:
  - klasa naz    `ol.naz {list-style-type: circle}`
- Odwołanie do klasy:
  - klasa menu   `<ol class="menu">`
- Przykładowy wygląd szablonu:



- Przykładowy wygląd pliku index.html dla powyższego szablonu:

```
1 <frameset rows = "150,*">
2   <frame name = "top" scrolling = "no" src = "top.html"/>
3 </frameset>
4 <frameset cols = "150,*">
5   <frame name = "menu" scrolling = "no" src = "menu.html"/>
6   <frame name = "main" src = "main.html"/>
7 </frameset>
```

- Aby linki otwierały się w ramce o nazwie frame należy dodać w odnośniku atrybut target:
  - `<a href="jk.html" target="main">Jan Kochanowski</a>`
- Walidatory HTML:
  - <http://validator.w3.org/>

- <http://validator.aborla.net/index.php5?lang=pl>
- <http://cgi.w3.org/cgi-bin/tidy>
- Walidatory CSS:
  - <http://jigsaw.w3.org/css-validator/>
  - <http://www.htmlhelp.com/tools/csscheck/>

## Podstawy HTML i arkusze stylów CSS (część 7)

- Szablon stron HTML oparty o elementy div
- Inline, wewnętrzne i zewnętrzne arkusze stylów CSS,
- Walidacja dokumentu HTML i arkusz stylów CSS.

### Cel główny:

- Wykonanie prostej, walidowalnej strony www (kilka dokumentów o treści w postaci piosenek, wierszy, wykonanych w części 3 ćwiczenia),
- Utworzenie strony www – szablonu zawierającego menu, nagłówek i treść.

### Cele cząstkowe:

- Logiczny podział na elementy stałe (spis utworów, nagłówek) i zmienne (utwory),
- Dołączanie stylów wewnętrznych i zewnętrznych,
- Dynamiczne ładowanie statycznych dokumentów do treści.

### Zadania do wykonania

- Wykorzystując dokumenty stworzone w części 3 wykonać:
  - spis treści w postaci listy odnośników do utworów umieszczone jeden pod drugim, umieścić go w elemencie div o identyfikatorze menu,
  - odnośniki w menu mają być formatowane wg klasy menu,
  - dowolną stronę z utworem należy sformatować z użyciem klasy utwor i umieścić ją w elemencie div o identyfikatorze main,
- Utworzyć stronę z nagłówkiem (ogólne informacje o zawartości strony) i umieścić go w elemencie div o identyfikatorze top,
- Wykorzystując pozycjonowanie float rozmieścić tak elementy div by tworzyły następującą strukturę:
  - menu – z lewej strony,
  - top – na górze,
  - main – pozostałe miejsce.
- Dla każdego utworu utworzyć podobną strukturę, zawierającą elementy o identyfikatorach top, menu, main,
- Zmodyfikować adresy odnośników tak by działały one prawidłowo,
- Dokonać walidacji strony www oraz arkusza stylów CSS:
  - w przypadku błędów poprawić,
  - w przypadku ostrzeżeń, określić ich istotność i poprawić.
- Zmodyfikować arkusz stylów css tak by menu było z prawej strony.

### Zawartość sprawozdania z ćwiczenia

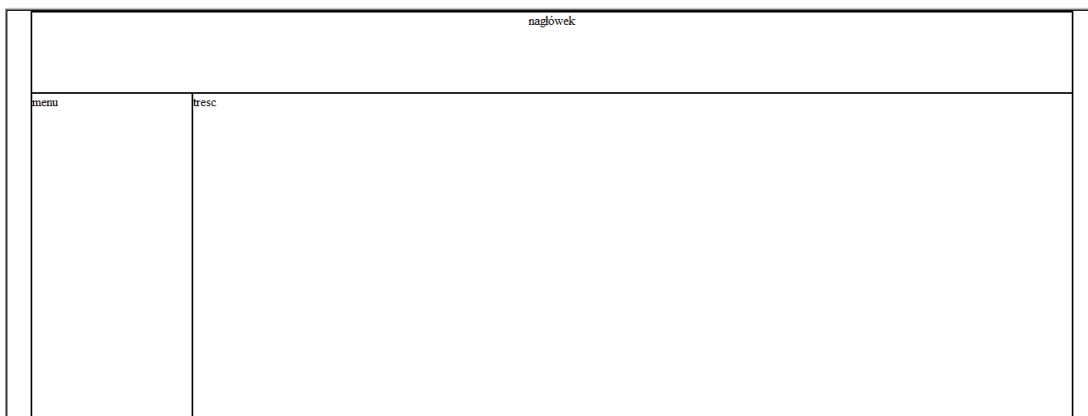
- Protokół powinien zawierać:
  - opis różnic jakie pojawiły się w dostępnych przeglądarkach po zastosowaniu arkusza stylów (przynajmniej w 3 różnych),
  - opis błędów i ostrzeżeń jakie pojawiły się podczas walidacji wraz ze sposobem ich poprawienia,
  - właściwości, których użycie tworzy różne efekty w różnych przeglądarkach,

## Źródła wiedzy

- R. Berezowski, Metajęzyki, wykład multimedialny,
- <http://validator.w3.org/>,
- <http://www.kurshtml.boo.pl/>
- <http://www.w3.org/>
- <http://www.w3schools.com/>
- <http://jigsaw.w3.org/css-validator/>

## Porady, wskazówki i podpowiedzi

- Definicja DOCTYPE:
  - `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">`
- Ustawienie kodowania znaków:
  - `<meta http-equiv="Content-Type" content="text/html; charset=utf-8">`
- Dołączanie stylów zewnętrznych:
  - `<link rel="stylesheet" href="plik.css" type="text/css">`
- Odwołanie do klasy:
  - klasa menu `<ol class="menu">`
- Przykładowy wygląd szablonu:



- Przykładowy wygląd arkusza stylów plik.css dla powyższego szablonu:

```
1  * {border: 1px solid black;  
2    padding: 0px;  
3    margin: 0px}  
4  body{ width: 1300px;  
5        margin: 0 auto;  
6        color: black;  
7        height: 500px}  
8  #top{ padding: 0;  
9        height: 97px;  
10       text-align: center;}  
11 #menu{float: left;  
12       width: 200px;  
13       height: 400px}  
14 #main{margin-left: 200px;  
15       height: 400px;}  
16
```

- Walidatory HTML:
  - <http://validator.w3.org/>
  - <http://validator.aborla.net/index.php5?lang=pl>
  - <http://cgi.w3.org/cgi-bin/tidy>
- Walidatory CSS:
  - <http://jigsaw.w3.org/css-validator/>
  - <http://www.htmlhelp.com/tools/csscheck/>

## Podstawy XML i arkusze stylów CSS

- Przypomnienie podstaw języka XML,
- Przypomnienie ogólnej struktury języka XML, instrukcje przetwarzania, zapis elementów, znaczników,
- Sposób rozmieszczania elementów w przeglądarce,
- Różnice w wyglądzie dokumentów XML dla różnych przeglądarek,
- Widok źródła dokumentu,
- Walidacja dokumentu XML.

### Cel główny:

- Wykonanie prostego, walidowalnego dokumentu XML (treść w postaci piosenki, wiersza, itp.).

### Cele cząstkowe:

- Przypomnienie podstawowej struktury języka XML,
- Walidacja własnego dokumentu HTML,
- Łączenie dokumentu XML i arkusza stylów CSS.

### Zadania do wykonania

- Przypomnieć sobie dowolny utwór wierszowany i zapisać go jako dokument HTML:
  - określić autora, tytuł, wyróżnić oba elementy,
  - wypisać 2 zwrotki,
  - użyć kilku różnych znaczników, do opisu struktury,
  - użyć arkusza stylów do opisu wyglądu,
  - ustawić kodowanie znaków i kodowanie pliku tak by można było używać polskich znaków,
- Na podstawie treści dokumentu HTML stworzyć dokument XML:
  - opracować strukturę dokumentu XML (rozmieszczenie i nazwy znaczników),
  - ustawić kodowanie znaków i kodowanie pliku tak by można było używać polskich znaków,
- Obejrzeć dokument XML w różnych przeglądarkach,
  - obejrzeć różnice w wyglądzie w różnych przeglądarkach,
- Dołączyć arkusz stylów CSS:
  - właściwości dobrać tak by dokument XML w przeglądarce wyglądał tak samo jak dokument HTML,
  - obejrzeć różnice w wyglądzie w różnych przeglądarkach,
- Dokonać walidacji dokumentu HTML i XML:
  - w przypadku błędów poprawić,
  - w przypadku ostrzeżeń, określić ich istotność i poprawić.

### Zawartość sprawozdania z ćwiczenia

- Protokół powinien zawierać:
  - opis różnic jakie pojawiły się w dostępnych przeglądarkach (przynajmniej w 3 różnych),

- opis błędów i ostrzeżeń jakie pojawiły się podczas walidacji wraz ze sposobem ich poprawienia,

## Źródła wiedzy

- R. Berezowski, Metajęzyki, wykład multimedialny,
- <http://validator.w3.org/>,
- <http://www.kurshtml.booo.pl/>
- <http://www.w3.org/>
- <http://www.w3schools.com/>

## Porady, wskazówki i podpowiedzi

- Definicja DOCTYPE (HTML):
  - `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">`
- Ustawienie kodowania znaków (HTML):
  - `<meta http-equiv="Content-Type" content="text/html; charset=utf-8">`
- Dołączenie zewnętrznego arkusza ze stylami (HTML):
  - `<link rel="stylesheet" href="plik.css" type="text/css">`
- Definicja XML i kodowanie znaków:
  - `<?xml version="1.0" encoding="utf-8" ?>`
- Dołączenie zewnętrznego arkusza ze stylami (XML):
  - `<link rel="stylesheet" href="plik.css" type="text/css">`
- Przykładowy wygląd dokumentu XML bez arkusza stylów CSS:

```
<?xml version="1.0" encoding="utf-8" ?>
<utwor>
  <autor>Adam Mickiewicz</autor>
  <tytul>A Judah</tytul>
  <zbiór>Sonety krymskie</zbiór>
</utwor>
<strofa>
  <wers>Lubię poglądać wsparty na Judahu skale,</wers>
  <wers>Jak spienione bałwany, to w czarne szeregi</wers>
  <wers>Ścisnąwszy się, buchają, to jak srebrne śniegi</wers>
  <wers>W milionowych tęczach kołują wspaniale.</wers>
</strofa>
<strofa>
  <wers>Trąca się o mieliznę, rozbija na fale,</wers>
  <wers>Jak wojsko wielorybów, zalegając brzegi,</wers>
  <wers>Zdobędą łód w tryumfie, i na powrót zbiegi,</wers>
  <wers>Mieczą za sobą muszle, perły i korale.</wers>
</strofa>
<strofa>
  <wers>Podobnie na twe serce, o poeto młody!</wers>
  <wers>Namiętność często groźne wzburza niepogody;</wers>
  <wers>Lecz gdy podniesiesz bardon, ona bez twój szkody</wers>
</strofa>
<strofa>
  <wers>Ucieka w zapomnienia pograć się toni,</wers>
  <wers>I nieśmiertelne pieśni za sobą uroni,</wers>
  <wers>Z których wieki uplotą ozdobę twych skroni.</wers>
</strofa>
</utwor>
```

- Przykładowy wygląd arkusza stylów plik-xml.css:

```
1 * {display: block;}
2 utwor {color: black; background-color: silver; font-size: 15px; width: 300px}
3 autor {color: silver; background-color: black}
4 tytul {text-transform: uppercase}
5 zbiór {font-style: italic}
6 strofa {margin-top: 10px;}
7 wers {margin-left: 10px}
8 wers: first-letter {font-size: 180%; color: blue}
```

- Przykładowy wygląd dokumentu XML z arkuszem stylów CSS:

Adam Mickiewicz  
AJUDAH  
*Sonetu krymskie*

Lubię poglądać wsparty na Judahu skale,  
Jak spienione bałwany, to w czarne szeregi  
Ścisnąwszy się, buchają, to jak srebrne śniegi  
W milionowych tęczach kołują wspaniale.

Trącą się o mieliznę, rozbijają na fale,  
Jak wojsko wielorybów, zalegając brzegi  
Zdobędą ład w tryumfie, i na powrót zbiegi  
Mieć za sobą muszle, perły i korale.

Podobnie na twe serce, o poeto młody!  
Namiętność często groźne wzburza niepogody;  
Lecz gdy podniesiesz bardon, ona bez twój  
szkody

Ucieka w zapomnienia pograć się toni,  
I nieśmiertelne pieśni za sobą uрони,  
Z których wieki uplotą ozdobę twych skroni.

- 
- Walidatory:
  - <http://validator.w3.org/>
  - <http://validator.aborla.net/index.php5?lang=pl>
  - <http://cgi.w3.org/cgi-bin/tidy>
- Walidatory CSS:
  - <http://jigsaw.w3.org/css-validator/>
  - <http://www.htmlhelp.com/tools/csscheck/>



## Arkusze stylów XSLT + XML

- Przekształcenie XML + XSLT → HTML + CSS.

### Cel główny:

- Wykonanie prostego, walidowalnego dokumentu XML (treść w postaci kilku piosenek, wierszy, itp.), dołączenie arkusza stylów XSLT, przekształcającego XML do HTML oraz dołączenie do niego arkusza stylów CSS.

### Cele częściowe:

- Rozszerzenie wiadomości o przekształcaniu XSLT oraz o przestrzeniach nazw,
- Walidacja dokumentów XML, XSLT, HTML.

### Zadania do wykonania

- Przypomnieć sobie dowolny utwór wierszowany i zapisać go jako dokument XML:
  - opracować strukturę dokumentu XML (rozmieszczenie i nazwy znaczników),
  - ustawić kodowanie znaków i kodowanie pliku tak by można było używać polskich znaków
- Wykorzystując strukturę utworu opisaną w powyższym punkcie umieścić kilka różnych utworów w jednym dokumencie,
- Dołączyć reguły XSLT (przekształcenia do HTML):
  - reguła dla elementu głównego:
    - stworzyć część HEAD,
      - ustawić kodowanie (utf-8)
      - ustawić znacznik title,
      - ustawić atrybuty części body,
  - reguła dla elementu utwor (dziecko elementu głównego):
    - ustawić formatowanie autora utworu,
    - ustawić formatowanie tytułu utworu
    - ustawić formatowanie treści utworu,
  - reguła dla pozostałych elementów:
    - ustawić odpowiednie formatowanie (wielkość czcionki, kolory, marginesy), tak by wygląd był zadowalający,
  - plik zapisać z rozszerzeniem \*.xsl,
- Dokonać walidacji dokumentów XML i XSLT:
  - w przypadku błędów poprawić,
  - w przypadku ostrzeżeń, określić ich istotność i poprawić,
- Wykonać konwersje do formatu HTML:
  - dokonać konwersji do formatu HTML (w przeglądarce),
  - obejrzeć wynikowy plik HTML, określić elementy formatu do poprawienia i poprawić,
- Dołączyć w pliku XSLT arkusz ze stylami CSS:
  - sformatować tak CSS aby widok w przeglądarce był zadowalający,

## Zawartość sprawozdania z ćwiczenia

- Protokół powinien zawierać:
  - opis błędów i ostrzeżeń jakie pojawiły się podczas walidacji wraz ze sposobem ich poprawienia,
  - opis błędów w powstałym pliku HTML, które wymagały poprawienia.

## Źródła wiedzy

- R. Berezowski, Metajęzyki, wykład multimedialny,
- <http://validator.w3.org/>,
- <http://www.kurshtml.boo.pl/>
- <http://www.w3.org/>
- <http://www.w3schools.com/>

## Porady, wskazówki i podpowiedzi

- Przykładowy kod XML:

```
<zbiór nazwa="Sonety krymskie">
  <utwor>
    <autor>Adam Mickiewicz</autor>
    <tytuł>Stepy akemańskie</tytuł>
    <strofa>
      <wers>Wpłynąłem na suchego przestwór oceanu,</wers>
      <wers>Wóz nurza się w zieloność i jak łódka brodzi,</wers>
      <wers>Śród fali łąk szumiących, śród kwiatów powodzi,</wers>
      <wers>Omijam koralowe ostrowy burzanu.</wers>
    </strofa>
    <strofa>
    </strofa>
    <strofa>
    </strofa>
  </utwor>
  <utwor>
    <autor>Adam Mickiewicz</autor>
    <tytuł>Ajudah</tytuł>
    <zbiór>Sonety krymskie</zbiór>
    <strofa>
      <wers>Lubię poglądać wsparty na Judahu skale,</wers>
      <wers>Jak spienione bałwany, to w czarne szeregi</wers>
      <wers>Ścisnąwszy się, buchają, to jak srebrne śniegi</wers>
      <wers>W milionowych tęczach kołują wspaniale.</wers>
    </strofa>
    <strofa>
    </strofa>
    <strofa>
    </strofa>
  </utwor>
</zbiór>
```

- reguła dla elementu głównego:

```
<xsl:template match="/zbiór">
  <html>
  <head>
    <link rel="stylesheet" href="plik.css" type="text/css"/>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
  </head>
  <body>
    <xsl:apply-templates/>
  </body>
</html>
</xsl:template>
```

- reguła dla elementu utwor (dziecko elementu głównego):

```
<xsl:template match="utwor">
  <h3><xsl:value-of select="../@nazwa"/></h3>
  <h1><xsl:value-of select="autor"/></h1>
  <h2><xsl:value-of select="tytuł"/></h2>
  <xsl:apply-templates select="strofa"/>
</xsl:template>
```

- reguła dla pozostałych elementów:

```
<xsl:template match="strofa">
  <p font-size="smaller">
    <xsl:apply-templates select="wers"/>
  </p>
</xsl:template>
<xsl:template match="wers">
  <xsl:value-of select="."/><br/>
</xsl:template>
</xsl:stylesheet>
```

- Plik HTML bez CSS:

```
Sonety krymskie

Adam Mickiewicz

Stepy akermańskie

Wpłynąłem na suchego przestwór oceanu,
Wóz murza się w zieloność i jak łódka brodzi,
Śród fali łak szumiących, śród kwiatów powodzi,
Omijam koralowe ostrowy burzanu.

Już mrok zapada, nigdzie drogi ni kurhanu;
Patrzę w niebo, gwiazd szukam przewodniczek łodzi;
Tam z dala błyszczą obłoki? tam jutrenka wschodzi?
To błyszczą Dniestr, to wesza lampa Akermanu.

Stójmy! --- Jak cicho! --- Słyszę ciągnące żurawie,
Których by nie dościgły żrenice sokoła,
Słyszę, kędy się motyl kołysa na trawie,

Kędy wąż śliską pierś dotyka się ziola.
W takiej ciszy --- tak ucho natężam ciekawie,
Że słyszałbym głos z Litwy. --- Jedźmy, nikt nie woła!
```

- Plik HTML z CSS:

### Sonety krymskie

## ADAM MICKIEWICZ

### Stępy akermańskie

Wpłynąłem na suchego przestwór oceanu,  
Wóz nurza się w zieloność i jak łódka brodzi,  
Śród fali łąk szumiących, śród kwiatów powodzi,  
Omijam koralowe ostrowy burzanu.

Już mrok zapada, nigdzie drogi ni kurhanu;  
Patrzę w niebo, gwiazd szukam przewodniczek łodzi;  
Tam z dala błyszczą obłoki? tam jutrenka wschodzi?  
To błyszczą Dniestr, to weszła lampa Akermanu.

Stójmy! --- Jak cicho! --- Słyszę ciągnące żurawie,  
Których by nie dościgły żrenice sokoła;  
Słyszę, kędy się motyl kołysa na trawie,

Kędy wąż śliską pierśią dotyka się ziola.  
W takiej ciszy --- tak ucho natężam ciekawie,  
Że słyszałbym głos z Litwy. --- Jedźmy, nikt nie woła!

### Sonety krymskie

## ADAM MICKIEWICZ

### Ajudah

Gotowe

- Plik CSS:

```
h1 {font-variant: small-caps}
h2 {font-size: larger}
h3 {color: blue}
p {margin-top: 10px}
```

## Podstawy arkuszy stylów XSL (część 1)

- Przypomnienie ogólnej struktury języka XML,
- Arkusze stylów CSS,
- Podstawy arkusza stylów XSL,
- Procesor FOP (generowanie dokumentów PDF),
- Walidacja dokumentów XML, XSL.

### Cel główny:

- Wykonanie prostego, walidowalnego dokumentu XSL (treść w postaci piosenki, wiersza, itp.) oraz przedstawienie go w postaci dokumentu PDF.

### Cele cząstkowe:

- Określenie podstawowych elementów, arkusza stylów XSL
- Walidacja stworzonego dokumentu XSL,
- Instalacja programu FOP,
- Uruchomienie programu FOP (utworzenie dokumentu PDF).

### Zadania do wykonania

- Przypomnieć sobie dowolny utwór wierszowany i zapisać go jako dokument XML:
  - opracować strukturę dokumentu XML (rozmieszczenie i nazwy znaczników),
  - ustawić kodowanie znaków i kodowanie pliku tak by można było używać polskich znaków,
  - dołączyć arkusz ze stylami CSS, tak by uzyskać estetyczny wygląd na stronie www,
- Na podstawie dokumentu XML utworzonego wcześniej stworzyć layout podstawowej strony XSL,
  - ustawić nazwę strony,
  - ustawić rozmiar strony (wysokość, szerokość),
  - ustawić marginesy regionu głównego (region-body),
- Wprowadzić treść strony:
  - w pierwszym bloku wprowadzić autora,
  - w drugim bloku wprowadzić tytuł,
  - w kolejnych blokach umieścić treść,
  - plik zapisać z rozszerzeniem \*.fo,
- Dokonać walidacji dokumentu XSL:
  - w przypadku błędów poprawić,
  - w przypadku ostrzeżeń, określić ich istotność i poprawić,
- Wykonać konwersję do formatu PDF:
  - ściągnąć i rozpakować program FOP,
  - dokonać konwersji utworzonego pliku \*.fo do formatu pdf,
  - obejrzeć wynikowy plik PDF, określić elementy formatu do poprawienia,
- Wprowadzić dodatkowe poprawki do pliku \*.fo tak aby wygląd był zadowalający:
  - wyróżnić autora i tytuł (różne style),
  - dodać margines (lewy i górny) do zwrotek,

- zwrotki zachowywały swoje formatowanie (entery i spacje zachowane).

## Zawartość sprawozdania z ćwiczenia

- Protokół powinien zawierać:
  - opis błędów i ostrzeżeń jakie pojawiły się podczas walidacji wraz ze sposobem ich poprawienia,
  - opis błędów w powstałym pliku PDF, które nie zostały naprawione.

## Źródła wiedzy

- R. Berezowski, Metajęzyki, wykład multimedialny,
- <http://validator.w3.org/>,
- <http://www.w3schools.com/xslfo/default.asp>
- <http://www.renderx.com/tutorial.html>
- <http://www.learn-xsl-fo-tutorial.com/>

## Porady, wskazówki i podpowiedzi

- Definicja XML i kodowanie znaków:
  - `<?xml version="1.0" encoding="utf-8"?>`
- Dołączenie zewnętrznego arkusza ze stylami (XML):
  - `<link rel="stylesheet" href="plik.css" type="text/css">`
- Przykładowy wygląd dokumentu XML bez arkusza stylów CSS:

```
<?xml version="1.0" encoding="utf-8"?>
<utwor>
  <autor>Adam Mickiewicz</autor>
  <tytuł>Ajadah</tytuł>
  <zbiór>Sonety krymskie</zbiór>
</utwor>
<strofa>
  <wers>Łubię poglądać wsparty na Judahu skale,</wers>
  <wers>Jak spienione bałwany, to w czarne szeregi</wers>
  <wers>Ścisnąwszy się, buchają, to jak srebrne śniegi</wers>
  <wers>W milionowych tęczach kołują wspaniale.</wers>
</strofa>
<strofa>
  <wers>Trąca się o mieliznę, rozbijają na fale,</wers>
  <wers>Jak wojsko wielorybów, zalegając brzegi,</wers>
  <wers>Zdobędą łód w tryumfie, i na powrót zbiegi,</wers>
  <wers>Mieć za sobą muszle, perły i korale.</wers>
</strofa>
<strofa>
  <wers>Podobnie na twe serce, o poeto młody!</wers>
  <wers>Namiętność często groźne wzburza niepogody;</wers>
  <wers>Lecz gdy podniesiesz bardon, ona bez twój szkody</wers>
</strofa>
<strofa>
  <wers>Ucieka w zapomnienia pogrążyć się toni,</wers>
  <wers>I nieśmiertelne pieśni za sobą uroni,</wers>
  <wers>Z których wieki uplotą ozdobę twych skroni.</wers>
</strofa>
</utwor>
```

- Przykładowy wygląd arkusza stylów plik-xml.css:

```
1 * {display: block;}
2 utwor {color: black; background-color: silver; font-size: 15px; width: 300px}
3 autor {color: silver; background-color: black}
4 tytuł {text-transform: uppercase}
5 zbiór {font-style: italic}
6 strofa {margin-top: 10px;}
7 wers {margin-left: 10px}
8 wers:first-letter {font-size: 180%; color: blue}
```

- Przykładowy wygląd dokumentu XML z arkuszem stylów CSS:

Adam Mickiewicz  
AJUDAH  
*Sonety krymskie*

Lubię poglądać wsparty na Judahu skale,  
Jak spienione bałwany, to w czarne szeregi  
Ścisnąwszy się, buchają, to jak srebrne śniegi  
W milionowych tęczach kołują wspaniale.

Trącą się o mieliznę, rozbijają na fale,  
Jak wojsko wielorybów, zalegając brzegi  
Zdobędą łód w tryumfie, i na powrót zbiegi,  
Mieczą za sobą muszle, perły i korale.

Podobnie na twe serce, o poeto młody!  
Namiętność często groźne wzburza niepogody;  
Lecz gdy podniesiesz bardon, ona bez twój  
szkody

Ucieka w zapomnienia pogrążyć się toni,  
I nieśmiertelne pieśni za sobą uрони,  
Z których wieki uplotą ozdobę twych skroni.

- Walidatory:
  - <http://validator.w3.org/>
  - <http://validator.aborla.net/index.php5?lang=pl>
  - <http://cgi.w3.org/cgi-bin/tidy>
- Walidatory CSS:
  - <http://jigsaw.w3.org/css-validator/>
  - <http://www.htmlhelp.com/tools/csscheck/>
- Przykładowy kod XSL:

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
3  <fo:layout-master-set>
4  <fo:simple-page-master master-name="A4" height="150px" width="200px"
5  <fo:region-body margin="10px" column-count="3" />
6  </fo:simple-page-master>
7  </fo:layout-master-set>
8  <fo:page-sequence master-reference="A4">
9  <fo:flow flow-name="xsl-region-body" white-space="pre">
10 <fo:block>Adam Mickiewicz</fo:block>
11 <fo:block>Ajudah</fo:block>
12 <fo:block width="50px">
13 Lubię poglądać wsparty na Judahu skale,
14 Jak spienione bałwany, to w czarne szeregi
15 Ścisnąwszy się, buchają, to jak srebrne śniegi
16 W milionowych tęczach kołują wspaniale.
17 </fo:block>
18 <fo:block width="50px">
```

- Walidacja XSL:
  - na stronie <http://www.renderx.com/tools/validators.html> znajduje się plik fo.dtd zawierający strukturę dokumentu XSL. Następnie dowolnym walidatorem można sprawdzić plik \*.fo,
- Program FOP:
  - <http://archive.apache.org/dist/xmlgraphics/fop/binaries/> i rozpakować:
  - przykładowa instrukcja tworząca pdf:
    - fop plik.fo plik.pdf,
  - do obejrzenia pliku PDF potrzebny jest Acrobat Reader,
- Plik PDF:

```
Adam Mickiewicz
Ajudah
Lubi# pogl#da# wsparty na Judahu skale, Jak spienione ba#wany, to w czarne szeregi #cisl#wszy si#,
buchaj#, to jak srebrne #niegi W milionowych t#czach ko#uj# wspaniale.
Tr#c# si# o mielizn#, rozbij# na fale, Jak wojsko wielorybów, zalegaj#c brzegi, Zdob#d# l#d w tryumfie,
i na powrót zbiegi, Miec# za sob# muszle, per#y i korale.
Podobnie na twe serce, o poeto m#ody! Nami#tno## cz#sto gro#ne wzburza niepogody; Lecz gdy
podniesiesz bardon, ona bez twej szkody
Ucieka w zapomnienia pogr##y# si# toni, I nie#miertelne pie#ni za sob# uroni, Z których wieki uplot#
ozdob# twych skroni.
```

- Program FOP:
  - wyróżnić autora i tytuł (różne style):
    - <fo:block font-size="larger" color="blue">
    - <fo:block text-decoration="underline" font-style="oblique">
  - dodać margines (lewy i górny) do zwrotek:
    - margin="10px 0 0 10px"
    - margin-top="10px" margin-left="10px"
  - zwrotki zachowywały swoje formatowanie (entery i spacje zachowane):
    - white-space="pre"
- Plik PDF po poprawkach:

```
Adam Mickiewicz
Ajudah

Lubi# pogl#da# wsparty na Judahu skale,
Jak spienione ba#wany, to w czarne szeregi
#cisl#wszy si#, buchaj#, to jak srebrne #niegi
W milionowych t#czach ko#uj# wspaniale.

Tr#c# si# o mielizn#, rozbij# na fale,
Jak wojsko wielorybów, zalegaj#c brzegi,
Zdob#d# l#d w tryumfie, i na powrót zbiegi,
Miec# za sob# muszle, per#y i korale.

Podobnie na twe serce, o poeto m#ody!
Nami#tno## cz#sto gro#ne wzburza niepogody;
Lecz gdy podniesiesz bardon, ona bez twej szkody
```



## Podstawy arkuszy stylów XSL (część 2)

- Procesor FOP (generowanie dokumentów PDF)
  - polskie znaki,
- Regiony XSL.

### Cel główny:

- Wykonanie prostego, walidowalnego dokumentu XSL (treść w postaci piosenki, wiersza, itp.) oraz przedstawienie go w postaci dokumentu PDF.

### Cele cząstkowe:

- Określenie podstawowych elementów, arkusza stylów XSL,
- Walidacja stworzonego dokumentu XSL,
- Konfiguracja programu FOP – polskie znaki,
- Poznanie regionów w XSL.

### Zadania do wykonania

- Przypomnieć sobie dowolny utwór wierszowany i zapisać go jako dokument XSL:
  - stworzyć layout podstawowej strony XSL,
  - ustawić nazwę strony,
  - ustawić rozmiar strony (wysokość, szerokość),
  - ustawić marginesy regionu głównego (region-body), oraz wielkości pozostałych regionów (region-before, region-after),
- Wprowadzić treść strony:
  - w region-body wprowadzić autora, tytuł i treść utworu,
  - w region-before wstawić nazwisko autora,
  - w region-after wstawić nazwisko numer strony,
  - odpowiednio sformatować treści tak aby wygląd był zadowalający,
  - plik zapisać z rozszerzeniem \*.fo,
- Dokonać walidacji dokumentu XSL:
  - w przypadku błędów poprawić,
  - w przypadku ostrzeżeń, określić ich istotność i poprawić,
- Dołączyć polskie znaki:
  - wygenerować plik metryczny czcionki przy pomocy klasy TTFFReader,
  - zarejestrować font w pliku konfiguracyjnym,
  - dołączyć font do pliku \*.fo,
- Wykonać konwersję do formatu PDF:
  - dokonać konwersji utworzonego pliku \*.fo do formatu pdf,
  - obejrzeć wynikowy plik PDF, określić elementy formatu do poprawienia i poprawić,

### Zawartość sprawozdania z ćwiczenia

- Protokół powinien zawierać:
  - opis błędów i ostrzeżeń jakie pojawiły się podczas walidacji wraz ze sposobem ich poprawienia,

- opis błędów w powstałym pliku PDF, które nie zostały naprawione.

## Źródła wiedzy

- R. Berezowski, Metajęzyki, wykład multimedialny,
- <http://validator.w3.org/>,
- <http://www.w3schools.com/xslfo/default.asp>
- <http://www.renderx.com/tutorial.html>
- <http://www.learn-xsl-fo-tutorial.com/>

## Porady, wskazówki i podpowiedzi

- Przykładowy kod XSL:

```
<?xml version="1.0" encoding="utf-8"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format" font-family="Verdana">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="A4a" height="14cm" width="10cm" page-height="14cm"
page-width="20cm">
      <fo:region-body margin="1cm"/>
      <fo:region-before extent="1cm"/>
      <fo:region-after extent="1cm"/>
    </fo:simple-page-master>
  </fo:layout-master-set>
  <fo:page-sequence master-reference="A4a">
    <fo:static-content flow-name="xsl-region-before" text-align="center">
      <fo:block font-size="smaller">Adam Mickiewicz</fo:block>
    </fo:static-content>
    <fo:static-content flow-name="xsl-region-after" text-align="right">
      <fo:block font-size="smaller"><fo:page-number/></fo:block>
    </fo:static-content>
    <fo:flow flow-name="xsl-region-body">
      <fo:block font-size="larger" color="blue">Adam Mickiewicz</fo:block>
      <fo:block text-decoration="underline" font-style="italic">Ajudah</fo:block>
      <fo:block width="50px" margin="10px 0 0 10px">
        Lubię poglądać wsparty na Judahu skale,
```

- Walidacja XSL:
  - na stronie <http://www.renderx.com/tools/validators.html> znajduje się plik fo.dtd zawierający strukturę dokumentu XSL. Następnie dowolnym walidatorem można sprawdzić plik \*.fo,
- Konfiguracja programu FOP:
  - generowanie pliku metrycznego:  
java -cp \$FOPLIB org.apache.fop.fonts.apps.TTFReader verdana.ttf verdana.xml
  - Rejestracja fontów w pliku konfiguracyjnym (fop.xconf):  
<font metrics-url="verdana.xml" kerning="yes" embed-url="verdana.ttf">  
 <font-triplet name="Verdana" style="normal" weight="normal"/>  
</font>
  - dopisanie fontów do pliku \*.fo:  
<fo:root font-family="Verdana"...>
- Uruchomienie programu FOP z plikiem konfiguracyjnym:

```
fop -d -c conf\fop.xconf plik.fo plik.pdf
```

- Plik PDF:

Adam Mickiewicz

[Adam Mickiewicz](#)

Ajudah

Lubię poglądać wsparty na Judahu skale, Jak spienione bałwany, to w czarne szeregi Ścisnąwszy się, buchają, to jak srebrne śniegi W milionowych tęczach kołują wspaniale.

Trącą się o mieliznę, rozbijają na fale, Jak wojsko wielorybów, zalegając brzegi, Zdobędą ład w tryumfie, i na powrót zbiegi, Mieczą za sobą muszle, perły i korale.

Podobnie na twe serce, o poeto młody! Namiętność często groźne wzburza niepogody; Lecz gdy podniesiesz bardon, ona bez twej szkody

Ucieka w zapomnienia pogrążyć się toni, I nieśmiertelne pieśni za sobą uroni, Z których wieki uplotą ozdobę twych skroni.

## Podstawy arkuszy stylów XSL (część 3)

- Procesor FOP (generowanie dokumentów PDF)
  - dzielenie wyrazów,
- regiony XSL.

### Cel główny:

- Wykonanie prostego, walidowalnego dokumentu XSL (treść w postaci piosenki, wiersza, itp.) oraz przedstawienie go w postaci dokumentu PDF, w którym wykorzystane jest dzielenie wyrazów.

### Cele cząstkowe:

- Określenie podstawowych elementów, arkusza stylów XSL,
- Walidacja stworzonego dokumentu XSL,
- Konfiguracja programu FOP – dzielenie wyrazów,
- Poznanie regionów w XSL.

### Zadania do wykonania

- Przypomnieć sobie dowolny utwór wierszowany i zapisać go jako dokument XSL:
  - stworzyć layout podstawowej strony XSL,
  - ustawić nazwę strony,
  - ustawić rozmiar strony (wysokość, szerokość),
  - ustawić marginesy regionu głównego (region-body), oraz wielkości pozostałych regionów (region-before, region-after, region-start),
- Wprowadzić treść strony:
  - w region-body wprowadzić autora, tytuł i treść utworu,
  - w region-before wstawić nazwisko autora,
  - w region-after wstawić nazwisko numer strony oraz liczbę stron dokumentu,
  - w region-start wstawić na środku napis „na oprawę”,
  - odpowiednio sformatować treści tak aby wygląd był zadowalający,
  - plik zapisać z rozszerzeniem \*.fo,
- Dokonać walidacji dokumentu XSL:
  - w przypadku błędów poprawić,
  - w przypadku ostrzeżeń, określić ich istotność i poprawić,
- Wykonać konwersję do formatu PDF:
  - dokonać konwersji utworzonego pliku \*.fo do formatu pdf,
  - obejrzeć wynikowy plik PDF, określić elementy formatu do poprawienia i poprawić,
- Wprowadzić dodatkowe poprawki:
  - zmniejszyć długość pola region-start tak by napis nie mieścił się,
  - dołączyć dzielenie wyrazów,
  - „obrócić” tekst w region-start o 90 stopni.

### Zawartość sprawozdania z ćwiczenia

- Protokół powinien zawierać:

- opis błędów i ostrzeżeń jakie pojawiły się podczas walidacji wraz ze sposobem ich poprawienia,
- opis błędów w powstałym pliku PDF, które nie zostały naprawione.

## Źródła wiedzy

- R. Berezowski, Metajęzyki, wykład multimedialny,
- <http://validator.w3.org/>,
- <http://www.w3schools.com/xslfo/default.asp>
- <http://www.renderx.com/tutorial.html>
- <http://www.learn-xsl-fo-tutorial.com/>

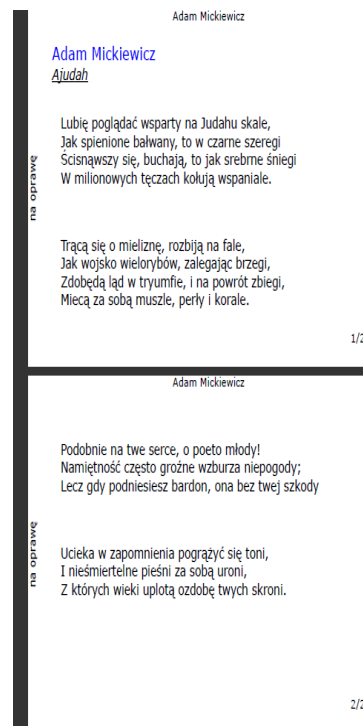
## Porady, wskazówki i podpowiedzi

- Przykładowy kod XSL:

```
<?xml version="1.0" encoding="utf-8"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format" font-family="Verdana" hyphenate="true" language="pl">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="A4a" height="10cm" width="14cm" page-height="10cm" page-width="14cm">
      <fo:region-body margin="1cm"/>
      <fo:region-before extent="1cm"/>
      <fo:region-after extent="1cm"/>
      <fo:region-start extent="1cm" overflow="hidden" reference-orientation="90"/>
    </fo:simple-page-master>
  </fo:layout-master-set>
  <fo:page-sequence master-reference="A4a">
    <fo:static-content flow-name="xsl-region-before" text-align="center">
      <fo:block font-size="smaller">Adam Mickiewicz</fo:block>
    </fo:static-content>
    <fo:static-content flow-name="xsl-region-after" text-align="right">
      <fo:block font-size="smaller"><fo:page-number/></fo:page-number-citation ref-id="koniec"/></fo:block>
    </fo:static-content>
    <fo:static-content flow-name="xsl-region-start">
      <fo:block text-align="center" font-size="smaller" width="1cm" overflow="hidden">na oprawę</fo:block>
    </fo:static-content>
    <fo:flow flow-name="xsl-region-body" white-space="pre">
      <fo:block font-size="larger" color="blue">Adam Mickiewicz</fo:block>
      <fo:block text-decoration="underline" font-style="italic">Ajudah</fo:block>
      <fo:block width="50px" margin="10px 0 0 10px">
        Kubie poglądać wsparty na Judahu skale,
        Jak spienione bałwany, to w czarne szeregi
        Icisnąwszy się, buchają, to jak srebrne śniegi
        / milionowych tęczach kołują wspaniale.
      </fo:block>
      <fo:block width="50px" margin-top="10px" margin-left="10px">
        <fo:block width="50px" margin="10px 0 0 10px">
          <fo:block width="50px" margin="10px 0 0 10px">
            <fo:block id="koniec"/>
          </fo:block>
        </fo:block>
      </fo:block>
    </fo:flow>
  </fo:page-sequence>
</fo:root>
```

- Walidacja XSL:
  - na stronie <http://www.renderx.com/tools/validators.html> znajduje się plik fo.dtd zawierający strukturę dokumentu XSL. Następnie dowolnym walidatorem można sprawdzić plik \*.fo,
- Konfiguracja programu FOP:
  - umieszczenie pliku (pl.xml lub pl\_PL.xml) ze wzorcami w katalogi FOP:
  - dopisanie kodu do dzielenia wyrazów:
    - jeżeli nazwa pliku ze wzorcami to pl.xml:  
<fo:root hyphenate="true" language="pl"...>

- jeżeli nazwa pliku ze wzorcami to pl\_PL.xml:  
`<fo:root hyphenate="true" language="pl" country="PL"...>`
- „obrót” tekstu w regin-start o 90 stopni:  
`<fo:region-start extent="1cm" overflow="hidden" reference-orientation="90"/>`
- wstawienie numeru strony i numeru ostatniej strony:
  - określenie końca dokumentu (ostatni blok elementu fo:flow):  
`<fo:block id="koniec"/>`
  - wstawienie numeru strony:  
`<fo:page-number/>`
  - wstawienie numeru strony elementu o id="koniec":  
`<fo:page-number-citation ref-id="koniec"/>`
- Plik PDF:



## Podstawy arkuszy stylów XSL (część 4)

- Procesor FOP (generowanie dokumentów PDF)
  - wstawianie grafiki – pliki graficzne,
  - wstawianie grafiki – pliki SVG.

### Cel główny:

- Wykonanie prostego, walidowalnego dokumentu XSL (treść – atlas państw) oraz przedstawienie go w postaci dokumentu PDF.

### Cele cząstkowe:

- Określenie podstawowych elementów, arkusza stylów XSL,
- Walidacja stworzonego dokumentu XSL,
- Wstawianie gotowych plików graficznych,
- Wstawianie plików graficznych w formacie SVG,
- Odnośniki,
- Generowanie spisu treści.

### Zadania do wykonania

- Utworzyć atlas geograficzny z krótkim opisem kilku państw, zapisać go jako dokument XSL:
  - stworzyć layout podstawowej strony XSL,
  - ustawić nazwę strony,
  - ustawić rozmiar strony (wysokość, szerokość),
  - ustawić marginesy regionu głównego (region-body), oraz wielkości pozostałych regionów (np. region-before, region-after),
- Wprowadzić treść strony:
  - w region-body wprowadzić nazwę państwa, krótki opis oraz flagę (w postaci pliku jpg/gif),
  - w region-before wstawić nazwisko autora treści strony,
  - w region-after wstawić nazwisko numer strony oraz liczbę stron dokumentu,
  - odpowiednio sformatować treści tak aby wygląd był zadowalający,
  - plik zapisać z rozszerzeniem \*.fo,
- Dokonać walidacji dokumentu XSL:
  - w przypadku błędów poprawić,
  - w przypadku ostrzeżeń, określić ich istotność i poprawić,
- Wykonać konwersję do formatu PDF:
  - dokonać konwersji utworzonego pliku \*.fo do formatu pdf,
  - obejrzeć wynikowy plik PDF, określić elementy formatu do poprawienia i poprawić,
- Wprowadzić dodatkowe poprawki:
  - utworzyć w postaci listy spis treści na pierwszej stronie,
  - dołączyć numery stron do spisu treści,
  - dołączyć elementy wykopkowania tak by numery stron w spisie treści były „równo ułożone”.

- Dołączyć na pierwszej stronie flagę Polski w formacie SVG.

## Zawartość sprawozdania z ćwiczenia

- Protokół powinien zawierać:
  - opis błędów i ostrzeżeń jakie pojawiły się podczas walidacji wraz ze sposobem ich poprawienia,
  - opis błędów w powstałym pliku PDF, które nie zostały naprawione.

## Źródła wiedzy

- R. Berezowski, Metajęzyki, wykład multimedialny,
- <http://validator.w3.org/>,
- <http://www.w3schools.com/xslfo/default.asp>
- <http://www.renderx.com/tutorial.html>
- <http://www.learn-xsl-fo-tutorial.com/>

## Porady, wskazówki i podpowiedzi

- Przykładowy kod XSL:

```
<fo:static-content flow-name="xsl-region-after">
<fo:block font-size="8px" border-top-style="solid" border-top-width="0.5px">
<fo:page-number/> / <fo:page-number-citation ref-id="koniec"/>
</fo:block>
</fo:static-content>
<fo:static-content flow-name="xsl-region-before">
<fo:block font-size="6px" border-bottom-style="solid" border-bottom-width="0.5px">Atlas Świata</fo:block>
</fo:static-content>
<fo:flow flow-name="xsl-region-body">
<fo:block text-align="center" font-size="14px" id="tytuł" break-after="page">
<fo:external-graphic src="url('1.svg')" width="30mm" height="20mm"/>
</fo:block>
<fo:instream-foreign-object xmlns:svg="http://www.w3.org/2000/svg">
<svg:svg width="50px" height="50px" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 16 10">
<svg:rect width="16" height="5" fill="white" style="stroke: black; stroke-width: 1px"/>
<svg:rect width="16" height="5" fill="#dcd43c" y="5" style="stroke: black; stroke-width: 1px"/>
</svg:svg>
</fo:instream-foreign-object>
</fo:block>
<fo:block font-size="larger" break-after="page">
<fo:page-citation>
<fo:basic-link color="blue" text-decoration="underline" internal-destination="polska"><fo:page-number-citation ref-id="polska"/></fo:basic-link>
</fo:block>
<fo:block text-align-last="justify">Europa<fo:leader leader-pattern="dots"/>
<fo:basic-link color="blue" text-decoration="underline" internal-destination="europa"><fo:page-number-citation ref-id="europa"/></fo:basic-link>
</fo:block>
<fo:block text-align-last="justify">USA<fo:leader leader-pattern="dots"/>
<fo:basic-link color="blue" text-decoration="underline" internal-destination="usa"><fo:page-number-citation ref-id="usa"/></fo:basic-link>
</fo:block>
<fo:block break-after="page" id="polska">Polska <fo:external-graphic src="url('pl.jpg')" content-width="30px" content-height="30px" margin="5px"/></fo:block>
<fo:block break-after="page" id="europa">Europa <fo:external-graphic src="url('ue.jpg')" content-width="30px" content-height="30px" margin="5px"/></fo:block>
<fo:block id="usa">USA <fo:external-graphic src="url('usa.jpg')" content-width="30px" content-height="30px" margin="5px"/></fo:block>
<fo:block id="koniec"/>
</fo:flow>
```

- Walidacja XSL:
  - na stronie <http://www.renderx.com/tools/validators.html> znajduje się plik fo.dtd zawierający strukturę dokumentu XSL. Następnie dowolnym walidatorem można sprawdzić plik \*.fo,
- Wstawienie zewnętrznego pliku z grafiką:



- ```
<fo:external-graphic src="url('pl.jpg')" content-  
width="30px" content-height="30px" margin="5px"/>
```
- ```
<fo:external-graphic src="url('1.svg')" width="30mm"  
height="20mm"/>
```
- **Wstawienie obiektu svg wewnątrz dokumentu XSL (flaga polski):**

```
<fo:instream-foreign-object  
xmlns:svg="http://www.w3.org/2000/svg">  
<svg:svg width="50px" height="50px"  
xmlns="http://www.w3.org/2000/svg" viewBox="0 0 16 10">  
<svg:rect width="16" height="5" fill="#fff" style="stroke:  
black; stroke-width: 1px"/>  
<svg:rect width="16" height="5" fill="#dc143c" y="5"  
style="stroke: black; stroke-width: 1px"/>  
</svg:svg>
```
- **wstawienie numeru strony i numeru ostatniej strony:**
  - **określenie końca dokumentu (ostatni blok elementu fo:flow):**

```
<fo:block id="koniec"/>
```
  - **wstawienie numeru strony:**

```
<fo:page-number/>
```
  - **wstawienie numeru strony elementu o id="koniec":**

```
<fo:page-number-citation ref-id="koniec"/>
```
- **Wykonanie spisu treści:**
  - **rozciągnięcie spisu na szerokość strony:**

```
<fo:block text-align-last="justify" margin-  
top="5px">Polska
```
  - **wykropkowanie:**

```
<fo:leader leader-pattern="dots"/>
```
  - **wstawienie odnośnika do elementu o id="polska" z napisem „numer strony”:**

```
<fo:basic-link color="blue" text-decoration="underline"  
internal-destination="polska">  
<fo:page-number-citation ref-id="polska"/>  
</fo:basic-link>
```
- **Plik PDF:**



## Podstawy arkuszy stylów XSL (część 5)

- Procesor FOP (generowanie dokumentów PDF)
  - wstawianie wyrażeń matematycznych.

### Cel główny:

- Wykonanie prostego, walidowalnego dokumentu XSL (treść – równanie kwadratowe) oraz przedstawienie go w postaci dokumentu PDF.

### Cele cząstkowe:

- Określenie podstawowych elementów, arkusza stylów XSL,
- Walidacja stworzonego dokumentu XSL,
- Wstawianie wyrażeń matematycznych w formacie mathml,
- Tworzenie tabel.

### Zadania do wykonania

- Utworzyć dokumentację na temat równanie kwadratowe, zapisać go jako dokument XSL:
  - stworzyć layout podstawowej strony XSL,
  - ustawić nazwę strony,
  - ustawić rozmiar strony (wysokość, szerokość),
  - ustawić marginesy regionu głównego (region-body), oraz wielkości pozostałych regionów (np. region-before, region-after),
- Wprowadzić treść strony:
  - w region-body wprowadzić wyrażenie opisujące równanie kwadratowe, a następnie w tabeli umieścić trzy różne, ogólne rozwiązania,
  - w region-before wstawić nazwisko autora treści strony,
  - w region-after wstawić nazwisko numer strony oraz liczbę stron dokumentu,
  - odpowiednio sformatować treści tak aby wygląd był zadowalający,
  - plik zapisać z rozszerzeniem \*.fo,
- Dokonać walidacji dokumentu XSL:
  - w przypadku błędów poprawić,
  - w przypadku ostrzeżeń, określić ich istotność i poprawić,
- Wykonać konwersję do formatu PDF:
  - dokonać konwersji utworzonego pliku \*.fo do formatu pdf,
  - obejrzeć wynikowy plik PDF, określić elementy formatu do poprawienia i poprawić.

### Zawartość sprawozdania z ćwiczenia

- Protokół powinien zawierać:
  - opis błędów i ostrzeżeń jakie pojawiły się podczas walidacji wraz ze sposobem ich poprawienia,
  - opis błędów w powstałym pliku PDF, które nie zostały naprawione.

## Źródła wiedzy

- R. Berezowski, Metajęzyki, wykład multimedialny,
- <http://validator.w3.org/>,
- <http://www.w3schools.com/xslfo/default.asp>
- <http://www.renderx.com/tutorial.html>
- <http://www.learn-xsl-fo-tutorial.com/>

## Porady, wskazówki i podpowiedzi

- Przykładowy kod XSL:

```
<fo:static-content flow-name="xsl-region-before">
<fo:block text-align="center">MathML</fo:block>
</fo:static-content>
<fo:flow flow-name="xsl-region-body">
<fo:block>
    równanie kwadratowe
<fo:instream-foreign-object>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
<mrow><mn>y</mn><mo>=</mo>
<mrow><mi>A</mi><msup><mi>x</mi><mn>2</mn></msup></mrow>
<mo>+</mo>
<mrow><mn>B</mn><mi>x</mi></mrow>
<mrow><mo>+</mo><mn>C</mn></mrow>
<mo>=</mo><mn>0</mn>
</mrow>
</math>
</fo:instream-foreign-object>
</fo:block>
<fo:block>
<fo:table border="solid 0.5px black">
<fo:table-body>
<fo:table-row height="auto">
<fo:table-cell border="solid 0.5px black">
<fo:block><fo:instream-foreign-object>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
<mrow><mi>x</mi><mo>=</mo>
<mfrac>
<mrow>
<mrow><mo>-</mo><mi>b</mi></mrow><mo>±</mo>
<msqrt>
<mrow><msup><mi>b</mi><mn>2</mn></msup><mo>-</mo>
<mrow><mn>4</mn><mi>a</mi><mi>c</mi></mrow>
</mrow>
</msqrt>
</mrow>
<mrow><mn>2</mn><mi>a</mi></mrow>
</math>
```

- Walidacja XSL:
  - na stronie <http://www.renderx.com/tools/validators.html> znajduje się plik fo.dtd zawierający strukturę dokumentu XSL. Następnie dowolnym walidatorem można sprawdzić plik \*.fo,
- Wstawienie wewnętrznego pliku z mml:

```
<fo:instream-foreign-object>
<math xmlns="http://www.w3.org/1998/Math/MathML">
<mrow><mn>y</mn><mo>=</mo>
<mrow><mi>A</mi><msup><mi>x</mi><mn>2</mn></msup></mrow>
<mo>+</mo><mrow><mn>B</mn><mi>x</mi></mrow>
<mrow><mo>+</mo><mn>C</mn></mrow>
<mo>=</mo><mn>0</mn>
</mrow>
```

```
</math>  
</fo:instream-foreign-object>
```

- Konfiguracja mathml:
  - <http://jeuclid.sourceforge.net/jeuclid-fop/index.html>
  - ściągnąć plik:
    - Plik jeuclid-3.1.8-distribution.zip
  - Skopiować do katalogu \$FOP/lib i włączyć do %classpath% następujące pliki:
    - -jeuclid-core-x.x.x.jar
    - -jeuclid-fop-x.x.x.jar
- wstawienie numeru strony i numeru ostatniej strony:
  - określenie końca dokumentu (ostatni blok elementu fo:flow):  
`<fo:block id="koniec"/>`
  - wstawienie numeru strony:  
`<fo:page-number/>`
  - wstawienie numeru strony elementu o id="koniec":  
`<fo:page-number-citation ref-id="koniec"/>`
- Plik PDF:

#### MathML

Równanie kwadratowe  $y = Ax^2 + Bx + C = 0$

$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$	$x = \frac{-b}{2a}$	brak rozwiązań
--	---------------------	----------------

## Podstawy arkuszy stylów XSL (część 6)

- Procesor FOP (generowanie dokumentów PDF)
- Przekształcenie XML + XSLT → XSL → PDF.

### Cel główny:

- Wykonanie prostego, walidowalnego dokumentu XML (treść w postaci kilku piosenek, wierszy, itp.), dołączenie arkusza stylów XSLT, przekształcającego XML do XSL oraz przedstawienie go w postaci dokumentu PDF.

### Cele cząstkowe:

- Rozszerzenie wiadomości o przekształceniu XSLT oraz o przestrzeniach nazw,
- Walidacja dokumentów XML, XSLT, XSL.

### Zadania do wykonania

- Przypomnieć sobie dowolny utwór wierszowany i zapisać go jako dokument XML:
  - opracować strukturę dokumentu XML (rozmieszczenie i nazwy znaczników),
  - ustawić kodowanie znaków i kodowanie pliku tak by można było używać polskich znaków
- Wykorzystując strukturę utworu opisaną w powyższym punkcie umieścić kilka różnych utworów w jednym dokumencie,
- Dołączyć reguły XSLT (przekształcenia do XSL):
  - reguła dla elementu głównego:
    - stworzyć layout podstawowej strony XSL,
    - ustawić nazwę strony,
    - ustawić rozmiar strony (wysokość, szerokość),
    - ustawić marginesy regionu głównego (region-body), oraz wielkości pozostałych regionów (region-before, region-after),
  - reguła dla elementu utworu (dziecko elementu głównego):
    - ustawić element <fo:static-content...>,
      - autor utworu w nagłówku,
      - numer strony / ilość stron w stopce,
    - ustawić element <fo:flow...> – treść utworu,
  - reguła dla pozostałych elementów:
    - ustawić odpowiednie formatowanie (wielkość czcionki, kolory, marginesy), tak by wygląd był zadowalający,
- plik zapisać z rozszerzeniem \*.xsl,
- Dokonać walidacji dokumentów XML i XSLT:
  - w przypadku błędów poprawić,
  - w przypadku ostrzeżeń, określić ich istotność i poprawić,
- Wykonać konwersję do formatu PDF:
  - dokonać konwersji utworzonego pliku \*.fo do formatu pdf,
  - obejrzeć wynikowy plik PDF, określić elementy formatu do poprawienia i poprawić,

## Zawartość sprawozdania z ćwiczenia

- Protokół powinien zawierać:
  - opis błędów i ostrzeżeń jakie pojawiły się podczas walidacji wraz ze sposobem ich poprawienia,
  - opis błędów w powstałym pliku PDF, które nie zostały naprawione.

## Źródła wiedzy

- R. Berezowski, Metajęzyki, wykład multimedialny,
- <http://validator.w3.org/>,
- <http://www.w3schools.com/xslfo/default.asp>
- <http://www.renderx.com/tutorial.html>
- <http://www.learn-xsl-fo-tutorial.com/>

## Porady, wskazówki i podpowiedzi

- Przykładowy kod XML:

```
<zbiór nazwa="Sonety krymskie">
  <utwór>
    <autor>Adam Mickiewicz</autor>
    <tytuł>Stepy akemańskie</tytuł>
    <strofa>
      <wers>Wpłynąłem na suchego przestwór oceanu,</wers>
      <wers>Wóz nurza się w zieloność i jak łódka brodzi,</wers>
      <wers>Śród fali łąk szumiących, śród kwiatów powodzi,</wers>
      <wers>Omijam koralowe ostrowy burzanu.</wers>
    </strofa>
  </utwór>
  <utwór>
    <autor>Adam Mickiewicz</autor>
    <tytuł>Ajudah</tytuł>
    <zbiór>Sonety krymskie</zbiór>
    <strofa>
      <wers>Lubię poglądać wsparty na Judahu skale,</wers>
      <wers>Jak spienione bałwany, to w czarne szeregi</wers>
      <wers>Ścisnąwszy się, buchają, to jak srebrne śniegi</wers>
      <wers>W milionowych tęczach kołują wspaniale.</wers>
    </strofa>
  </utwór>
</zbiór>
```

- reguła dla elementu głównego:

```
<xsl:template match="/zbiór">
  <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format" font-family="Verdana" page-width="13cm" page-height="21cm" width="13cm" height="21cm" language="pl" hyphenate="true">
    <fo:layout-master-set width="13cm" height="21cm" page-height="21cm" page-width="13cm">
      <fo:simple-page-master master-name="główna" page-height="21cm" page-width="13cm" margin="1cm" width="13cm" height="21cm">
        <fo:region-body extent="1cm"/> <fo:region-before extent="1cm"/> <fo:region-after extent="1cm"/>
      </fo:simple-page-master>
    </fo:layout-master-set>
    <xsl:apply-templates/>
  </fo:root>
</xsl:template>
```

- reguła dla elementu utwor (dziecko elementu głównego):

```
<xsl:template match="utwor">
  <fo:page-sequence master-reference="główna">
    <fo:static-content flow-name="xsl-region-before" text-align="last" justify">
      <fo:block font-size="smaller" border-bottom="thin solid black">
        <fo:leader leader-pattern="dots" alignment-baseline="hanging"/><xsl:value-of select=".."&nazwa"/><fo:leader leader-pattern="dots" alignment-baseline="hanging"/>
      </fo:block>
    </fo:static-content>
    <fo:static-content flow-name="xsl-region-after">
      <fo:block text-align="right" border-top="thin solid black">
        <fo:page-number/> / <fo:page-number-citation ref-id="koniec"/>
      </fo:block>
    </fo:static-content>
    <fo:flow flow-name="xsl-region-body">
      <xsl:apply-templates select="autor"/> <xsl:apply-templates select="tytuł"/> <xsl:apply-templates select="strofa"/>
      <xsl:if test=" (./tytuł) = ..utwor[count(//utwor)]/tytuł">
        <fo:block id="koniec"/>
      </xsl:if>
    </fo:flow>
  </fo:page-sequence>
</xsl:template>
```

- reguła dla pozostałych elementów:

```
<xsl:template match="autor">
  <fo:block color="green">
    <xsl:value-of select=".." />
  </fo:block>
</xsl:template>
<xsl:template match="tytuł">
  <fo:block font-style="italic">
    <xsl:value-of select=".." />
  </fo:block>
</xsl:template>
<xsl:template match="strofa">
  <fo:block margin-top="5mm" margin-left="5mm" font-size="smaller">
    <xsl:apply-templates select="wers" />
  </fo:block>
</xsl:template>
<xsl:template match="wers">
  <fo:block>
    <xsl:value-of select=".." />
  </fo:block>
</xsl:template>
```

- Plik PDF:

