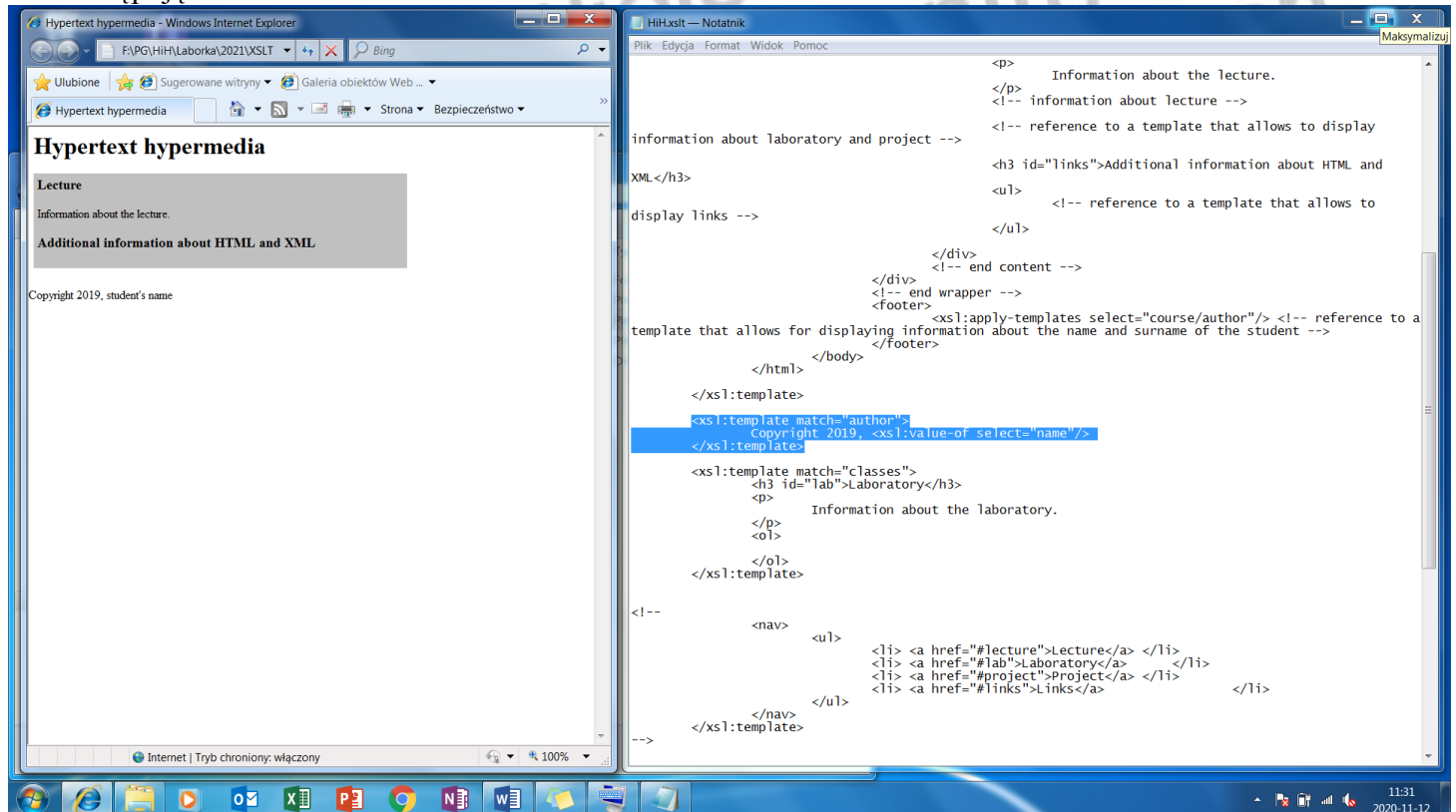


## Podstawy XSLT

Celem ćwiczenia jest zapoznanie ze standardem XSLT.

Do wykonania ćwiczenia potrzebny jest dowolny edytor plików tekstowych oraz przeglądarka internetowa.

- Na koniec zajęć swój katalog należy spakować do pliku **ZIP** o nazwie **Nazwisko\_Imie\_NrIndeksu\_XSLT.ZIP** (bez polskich liter) i przesłać na Moodlea.
- W przesyłanym katalogu ma się również znaleźć sprawozdanie z zajęć. Plik sprawozdania ma być nazwany **Nazwisko\_Imie\_NrIndeksu\_XSLT** i należy w nim umieszczać zrzuty ekranu zawierające jednocześnie widok kodu XSLT oraz widok przeglądarki. Na screen-ie ma być widoczna data i czas. Kod odpowiedzialny za dany etap powinien być zaznaczony. Przykładowy zrzut ekranu powinien wyglądać następująco:



1. Na dysku wskazanym przez prowadzącego stworzyć katalog nazwany własnym imieniem i nazwiskiem. Umieścić w nim pliki ściągnięte z Moodlea. Do pracy z plikami można użyć środowiska Visual lub zwykłego notatnika. Po zajęciach własny katalog należy **SKASOWAĆ**. Należy pamiętać o okresowym zachowywaniu wyników pracy.
2. Zapoznać się z plikami udostępnionymi na Moodleu. Przeanalizować plik HH.xml, zwrócić uwagę na strukturę dokumentu, wykorzystane znaczniki oraz powiązanie z dokumentem XSLT. Przeanalizować plik HH.xsl, zwrócić uwagę na szablon dla korzenia i sposób tworzenia dokumentu HTML. Na końcu pliku w komentarzach znajdują się przydatne fragmenty kodu HTML. **W czasie laboratorium należy używać konstrukcji <template> z atrybutem match. Konstrukcję <for-each> należy wykorzystać tylko wtedy, gdy zostanie to wyraźnie zaznaczone w instrukcji.**
3. (1pkt) W pliku XML dodać swoje imię i nazwisko w elemencie name i surname. W pliku XSLT pobrać swoje imię i nazwisko z dokumentu XML i wyświetlić je w stopce strony. Wykorzystać istniejący szablon z dopasowaniem do elementu author, uzupełnić go o odpowiednie składniki.
4. **Umieścić zrzut ekranu w pliku sprawozdania.**
5. (1pkt) Stworzyć szablon z nazwą. W szablonie umieścić menu do poruszania się w obrębie strony. Wywołać szablon w odpowiednim miejscu w szablonie dla korzenia.
6. **Umieścić zrzut ekranu w pliku sprawozdania.**

## Hypertext & hypermedia


7. (1,5 pkt) Wyświetlić obraz z Vannewarem Bushem. Stworzyć szablon z atrybutem **match** oraz wywołać go w odpowiednim miejscu. Dla obrazu wykorzystać **class="right"**, aby ustawić obraz po prawej stronie. Po najechaniu na obraz ma wyświetlać się tekst będący zawartością elementu `image` (tutaj: Vannevar Bush)
8. **Umieścić zrzut ekranu w pliku sprawozdania.**
9. (1,5pkt) Wyświetlić działające linki. Stworzyć szablon dla elementu `link`, w którym utworzyć element html-owy pozwalający na uzyskanie działającego linku. W odpowiednim miejscu wywołać szablon. Należy wyświetlić tylko dwa pierwsze linki z pliku XML (można wykorzystać predykat oraz funkcje: `position()` i `last()`).
10. **Umieścić zrzut ekranu w pliku sprawozdania.**
11. (2pkt) Wyświetlić informacje dotyczące tematów wykładu (element `theme`), posortować je rosnąco `<xsl:sort>` oraz ponumerować `<xsl:number>`. Aby numerowanie było prawidłowe należy wykorzystać atrybut **value** oraz funkcję `position()`. Aby przetworzyć wszystkie potrzebne węzły należy wykorzystać pętlę (`<for-each>`). Należy pamiętać, że predykaty w XPath pozwalają doprecyzować, o jaki zbiór węzłów chodzi (tutaj należy wybrać elementy `theme` dla których atrybut `kind` przyjmuje wartość `lecture`).
12. **Umieścić zrzut ekranu w pliku sprawozdania.**
13. (2 pkt) Wyświetlić informacje dotyczące laboratorium. Uzupełnić szablon dla elementu `classes`. W szablonie tym użyć instrukcji warunkowej (`<xsl:if>` lub `<xsl:choose>`), aby wykonać odpowiednie instrukcje tylko dla elementów dotyczących laboratorium (testować wartość atrybutu `kind`). Stworzyć odpowiednie szablony dla elementów `component` i `theme` i odpowiednio je wywoływać.
14. **Umieścić zrzut ekranu w pliku sprawozdania.**
15. (1 pkt) Wyświetlić informacje dotyczące projektu analogicznie jak zrobiono to dla informacji o laboratorium.
16. **Umieścić zrzut ekranu w pliku sprawozdania.**

### Hypertext hypermedia

#### Lecture

Information about the lecture.

1. DTD
2. HTML CSS
3. Hypertext & hypermedia
4. XML
5. XML Schema
6. XSLT



[Lecture](#)  
[Laboratory](#)  
[Project](#)  
[Links](#)

#### Laboratory

Information about the laboratory.

1. HTML + CSS
  - structure of the page
  - links
  - forms
  - css
2. XML + XML Schema
  - correctly formed XML file
  - creating a hierarchy
  - declaring elements, attributes
  - defining types
  - validating XML file
3. XSLT
  - XML -> HTML transformation
  - XML -> XML transformation

#### Project

Information about the project.

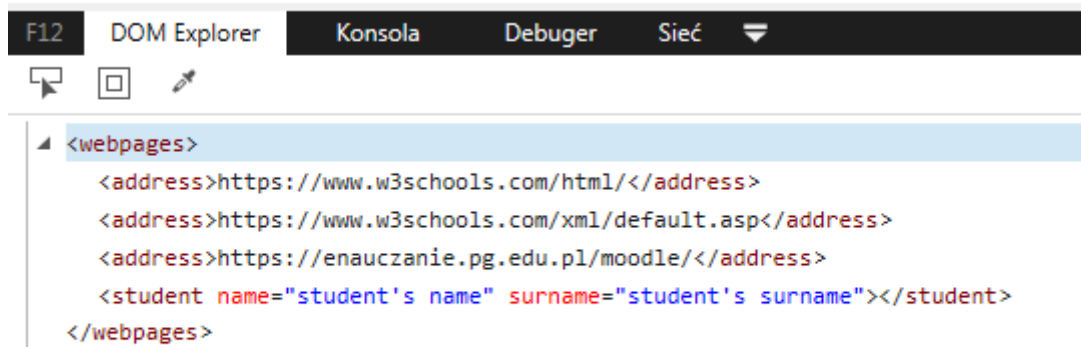
Title of the project	Score
HTML, XML, XSLT	20
XML, XML Schema, DTD	20

#### Additional information about HTML and XML

- [XML w3schools](#)
- [Moodle](#)

Copyright 2019, student's surname student's name

17. (1 pkt – zadanie dodatkowe, na dodatkowy punkt) Zapoznać się z plikiem HiH2.xslt. Plik pozwala na utworzenie nowego pliku XML, zawierającego tylko informacje o linkach. Zapoznać się ze sposobem tworzenia nowych elementów. Efekt działania pliku możemy obejrzeć np. wyświetlając plik HiH.xml w przeglądarce (w pliku HiH.xml należy wcześniej zmienić nazwę pliku xslt, wg którego będzie on przetwarzany). Strukturę nowego pliku XML możemy obejrzeć wybierając narzędzia deweloperskie F12. (Alternatywnie możemy przekształcić XML w Visual Studio). Należy dodać do pliku wynikowego element o nazwie student, dla którego należy dodać atrybuty name oraz surname. Wartości atrybutów należy pobrać z pliku XML.



18. Umieścić zrzut ekranu w pliku sprawozdania.

### XSLT krótka ściągą ☺

1. Szablony opisują, co ma być wykonane dla danego zakresu stosowania

```
<xsl:template match="zakres_stosowania" name="nazwa">
```

...

```
</xsl:template>
```

2. wywołanie szablonu zdefiniowanego z atrybutem match

```
<xsl:apply-templates select="zakres_stosowania"/>
```

3. wywołanie szablonu zdefiniowanego z atrybutem name

```
<xsl:call-template name="nazwa"/>
```

4. wybranie wartości

```
<xsl:value-of select="węzeł_drzewa_dokumentu"/>
```

testy węzłów

"/"	korzeń
"nazwa_elementu"	element
"@nazwa_atrybutu"	atrybut
"."	aktualny węzeł

5. instrukcja warunkowa

```
<xsl:if test="wyrażenie">
```

...

```
</xsl:if>
```

6. wielokrotny wybór

```
<xsl:choose>
```

```
<xsl:when test="wyrażenie1">
```

...

```
</xsl:when>
```

```
<xsl:when test="wyrażenie2">
```

```
...
</xsl:when>
...
<xsl:otherwise>
...
</xsl:otherwise >
</xsl:choose>
```

**7. pętla**

```
<xsl:for-each select="węzeł_drzewa_dokumentu">
...
</xsl:for-each>
```

**8. sortowanie**

```
<xsl:sort select="węzeł_drzewa_dokumentu" order="porządek_sort."/>
```

porządek sortowania

ascending	rosnąco
descending	malejąco

**9. numerowanie**

```
<xsl:number format="format_numerowania"/>
```

format\_numerowania to np. 1. a) A itp.

**10. tworzenie nowych elementów i atrybutów**

```
<xsl:element name="nazwa_elementu"></xsl:element>
<xsl:attribute name="nazwa_atrybutu"></xsl:attribute>
```