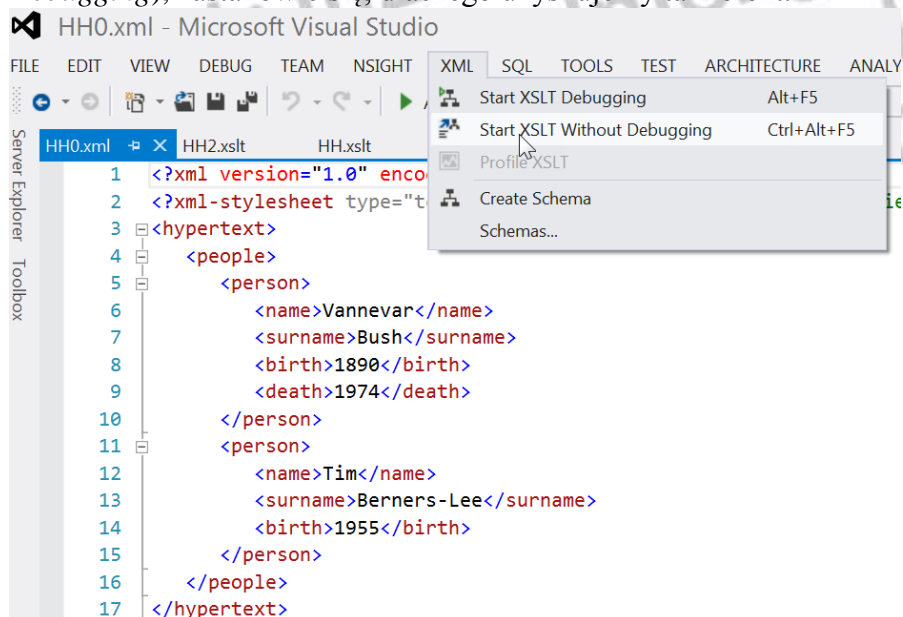


## Podstawy XSLT

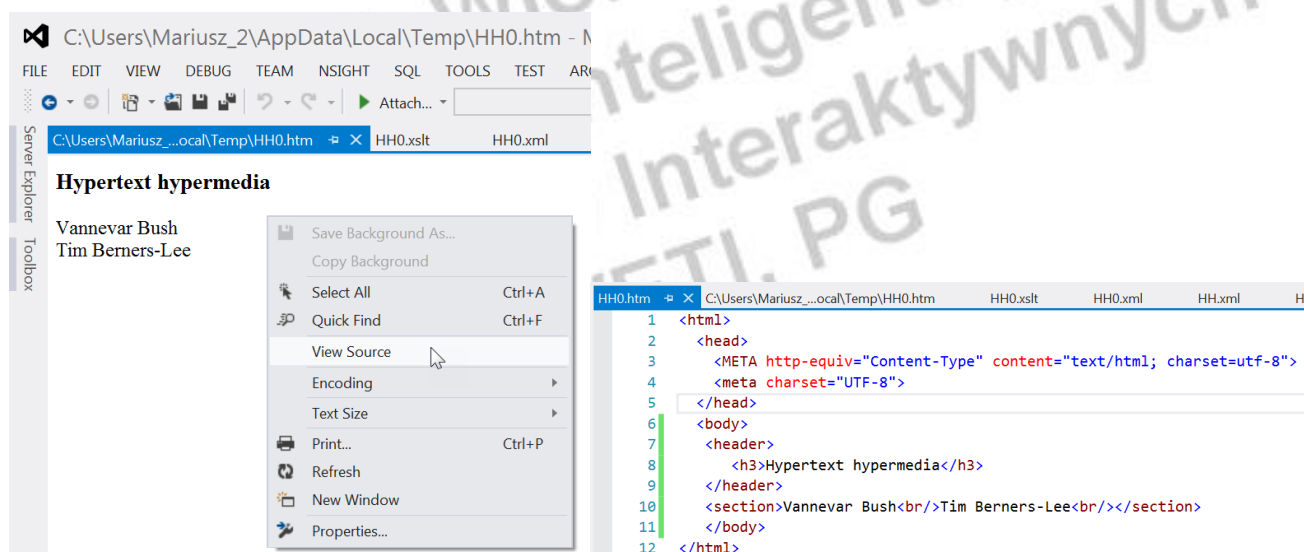
Celem ćwiczenia jest zapoznanie ze standardem XSLT.

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

1. Na dysku wskazanym przez prowadzącego stworzyć katalog nazwany własnym imieniem i nazwiskiem. Umieścić w nim pliki ściągnięte z Moodle'a. Do pracy z plikami można użyć środowiska Visual lub zwykłego notatnika. Po zajęciach własny katalog należy **SKASOWAĆ**. Należy pamiętać o okresowym zachowywaniu wyników pracy. Aby obejrzeć przetworzony zgodnie z regułami XSLT plik XML można wykorzystać procesor XSLT z Visual-a.
2. (1pkt) Zapoznać się z plikami udostępnionymi na Moodle'u. Przeanalizować plik HH0.xml oraz plik HH0.xslt, zwrócić uwagę na szablon dla korzenia (`<xsl:template match="/">`) i sposób tworzenia wynikowego dokumentu HTML. Przetworzyć plik xml w Visual Studio (XML->Start XSLT Without Debugging), zastanowić się, dlaczego uzyskujemy taki efekt.



Aby obejrzeć dokument, który powstał w wyniku transformacji należy z menu kontekstowego wybrać *View Source*. Kod będzie dostępny w nowym oknie, plik można zapisać (jako plik HTML) i obejrzeć w przeglądarce. Należy pamiętać, że przeglądarki mają wbudowany procesor XSLT, więc otwarcie pliku XML w przeglądarce również powinno pozwolić na przekształcenie XML do HTML.



Uzupełnić kod pliku xslt, aby uzyskać efekt jak na powyższym rysunku. W tym celu stworzyć szablon z dopasowaniem do odpowiedniego elementu (`<template>` z atrybutem `match`) i w nim umieścić instrukcje

pozwalające na wydobywanie wybranych informacji **<value-of>**).

**W czasie laboratorium należy używać szablonów **<template>** z atrybutem **match**.**

**Element **<for-each>** należy wykorzystać tylko wtedy, gdy zostanie to wyraźnie zaznaczone w instrukcji.**

3. **Poprosić prowadzącego o sprawdzenie pracy**
4. 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.
5. **(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.
6. **(0,5pkt)** Stworzyć szablon z nazwą (**<template>**) z atrybutem **name**). W szablonie umieścić menu do poruszania się w obrębie strony. Wywołać szablon w odpowiednim miejscu w szablonie dla korzenia.
7. **Poprosić prowadzącego o sprawdzenie pracy**
8. **(1pkt)** Wypisać informacje o systemach pobrane z dokumentu XML. W tym celu stworzyć szablon z dopasowaniem do elementu `system` (**<template>**) z atrybutem **match**). W szablonie wypisać nazwę systemu (`name`), opis systemu (`description`)
9. **(1pkt)** Wyświetlić działające linki związane z danym systemem. W tym celu stworzyć nowy szablon dla elementu `link` i wywołać go. W szablonie dla elementu `link` należy utworzyć element html-owy pozwalający na uzyskanie działającego linku (`<a>`).
10. **Poprosić prowadzącego o sprawdzenie pracy**
11. **(1pkt)** Wypisać informacje o osobach. Stworzyć odpowiedni szablon i wywołać go. Należy wypisać imię i nazwisko osoby oraz podać datę urodzin i śmierci. Tekst: *Date of death* ma się pojawić tylko, jeśli jest danej osoby wartość atrybutu `alive='no'`. W celu sprawdzenia wartości atrybutu wykorzystać instrukcję warunkową `if`.
12. **(1,5 pkt)** Wyświetlić obrazy związane z poszczególnymi osobami. Należy stworzyć szablon **<template>** z atrybutem **match** oraz wywołać go w odpowiednim miejscu. W szablonie utworzyć znacznik `<img>` z odpowiednimi parametrami. Nazwę obrazu należy pobrać z pliku XML z atrybutu `source` elementu `image`. Dla obrazu wykorzystać **class="right"**, aby ustawić obraz po prawej stronie. Po najechaniu na obraz ma wyświetlać się tekst będący zawartością elementu `image`.
13. **(0,5pkt)** Wyświetlić działające linki o osobach. Wykorzystać szablon stworzony w punkcie 9.
14. **Poprosić prowadzącego o sprawdzenie pracy**
15. **(1pkt)** Wyświetlić informacje dotyczące osiągnięć osoby (element `achievement`), posortować je rosnąco wg elementu `title` **<xsl:sort>** oraz ponumerować **<xsl:number>**. Aby numerowanie było prawidłowe należy wykorzystać dla znacznika `number` atrybut **value** oraz funkcję `position()`. Aby przetworzyć wszystkie potrzebne węzły należy wykorzystać pętlę (**<for-each>**).
16. **(0,5pkt)** Zmodyfikować wywołanie szablonu wyświetlającego linki związane z systemami tak, aby wyświetlane były wszystkie pozycje za wyjątkiem ostatniej (można wykorzystać funkcje: `last()`, `position()`, `not()`)
17. **Poprosić prowadzącego o sprawdzenie pracy**

# Hypertext & hypermedia

## Hypertext hypermedia

### Scientists

#### Vannevar Bush

Date of birth: 1890  
Date of death: 1974

Vannevar Bush was a famous scientist in his days and was the science advisor to President Roosevelt during the Second World War, when science-based issues like inventing nuclear weapons were of great importance. After "As We May Think" ran in the Atlantic Monthly, it caused considerable discussion, and both Time and Life ran stories on the Memex. Life even had an artist draw up illustrations of how the Memex would look and a scenario of its projection positions as the user was completing a link. Doug Engelbart, who later became a pioneer in the development of interactive computing and invented the mouse, got part of his inspiration by reading Bush's article while waiting for a ship home from the Philippines in 1945. In spite of all this early interest surrounding the Memex it never got built. As hinted above, our current computer technology is still not able to support Bush's vision in its entirety. We do have computers with most of the Memex functionality but they are based on a completely different technology from the microfilm discussed by Bush.



#### Achievements

##### 1) Memex - "memory extender"

Bush described the Memex as "a sort of mechanized private file and library" and as "a device in which an individual stores his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility." The Memex would store this information on microfilm, which would be kept in the user's desk. This desk was intended to have several microfilm projection positions to enable the user to compare different microfilms, in a manner very similar to the windows that became popular on personal computers more than forty years later. The Memex would have a scanner to enable the user to input new material, and it would also allow the user to make handwritten marginal notes and comments. But Bush envisaged that most of the Memex contents are purchased on microfilm ready for insertion. Books of all sorts, pictures, current periodicals, newspapers, are thus obtained and dropped into place. Business correspondence takes the same path. Actually we have not yet reached the state of hypertext development where there is a significant amount of preprocessed information for sale that can be integrated with a user's existing hypertext structure.

- [Vannevar Bush](#)

#### Tim Berners-Lee

Date of birth: 1955

Sir Tim Berners Lee is an English computer scientist best known as the inventor of the World Wide Web. He is a Professorial Fellow of Computer Science at the University of Oxford and a professor at the Massachusetts Institute of Technology (MIT). Tim Berners-Lee proposed an information management system then implemented the first successful communication between a Hypertext Transfer Protocol (HTTP) client and server via the Internet.



#### Achievements

##### 1) Browser

A computer program with a graphical user interface for displaying and navigating between web pages.

##### 2) HTML

Hypertext Markup Language - The markup (formatting) language for the web.

##### 3) HTTP

Hypertext Transfer Protocol. Allows for the retrieval of linked resources from across the web.

##### 4) URI

Uniform Resource Identifier. A kind of "address" that is unique and used to identify to each resource on the web.

- [Sir Tim Berners-Lee](#)

### Systems

#### HyperCard

HyperCard was a powerful, yet extremely easy to use tool for creating "stacks" — essentially flat-file databases that used hyperlinks as a way of navigating a stack of "cards".

- [HyperCard main webpage](#)
- [Why HyperCard was so f\\*\\*\\*ing Great!](#)
- [HyperCard](#)

#### World Wide Web

The World Wide Web is a collection of linked resources and documents, connected by hyperlinks and URLs

- [HTML w3schools](#)
- [World Wide Web Consortium](#)

Copyright 2022, student's surname student's name

18. (1 pkt) Zapoznać się z plikiem HH2.xslt oraz HH2.xml. Transformacja pozwala na utworzenie nowego pliku XML, zawierającego tylko informacje o linkach. Zapoznać się ze sposobem tworzenia nowych elementów. Efekt działania pliku możemy obejrzeć standardowo w Visual Studio (opcja dla dokumentu wynikowego Edit->Advanced->Format Document pozwala pokazać dokument wynikowy w czytelniejszej formie) lub np. wyświetlając plik HH2.xml w przeglądarce (strukturę nowego pliku XML możemy obejrzeć wybierając narzędzia deweloperskie F12). Należy dodać do pliku wynikowego element o nazwie student, dla którego należy dodać atrybuty name oraz surname. Wartości atrybutów należy pobrać z pliku XML, wcześniej należy w pliku XML (HH2.xml) zmienić wartości na swoje imię i nazwisko.

```
<?xml version="1.0" encoding="utf-8"?>
<webpages>
  <address>https://www.britannica.com/topic/Harvard-University</address>
  <address>https://www.britannica.com/biography/Tim-Berners-Lee</address>
  <address>https://www.w3schools.com/html/</address>
  <address>https://hypercard.org/</address>
  <address>https://www.youtube.com/watch?v=2K1C5BZKP3I</address>
  <address>https://eshop.macsales.com/blog/64930-hypercard-one-stack-at-a-time/</address>
  <address>https://www.w3schools.com/html/</address>
  <address>https://www.w3.org/</address>
  <student name="student's name" surname="student's surname" />
</webpages>
```

19. Poprosić prowadzącego o sprawdzenie pracy

dr inż. Wioleta Szwoch, Katedra Inteligentnych Systemów Interaktywnych, WETI, PG

## XSLT krótka ściągą ☺

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

```
<xsl:template match="zakres_stosowania(wyrażenie XPath)" 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_sortowania"/>
```

porządek sortowania

**ascending**      rosnąco (wartość domyślna)

**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>
```