

## ICT-4540 Homework 4

### Purpose

This homework exercise allows you to explore the basics of XML Stylesheet Language (XSL) transformation, which allows manipulation of the XML document, rather than just styling it. You will create an (X)HTML table of the presidents, then sort them by party, and then add a portrait!

### What to Hand In

Hand in your XSL stylesheet, the XML document as modified to apply the stylesheet, and a browser screenshot of the XML document as interpreted and displayed by the browser after applying the transformation for each problem below.

### Problems

#### I. A Table Of Presidents: Convert the USPresidents.xml to an (X)HTML table

1. Begin with the *USPresidents.xml* and *presidents.css* files from a previous problem. Make a copy, because we will change these in new ways.
2. Attach an XML stylesheet called *president\_table.xsl* to the XML document
3. Rename the css file to *president\_table.css*.
4. Create an xml transform in the file *president\_table.xsl* and create a template to match the root element (see note on whether to use XSL version 1.0 or 2.0)
5. Within the root element template, establish the html root element in the output, the head element, its title, link to the *president\_table.css* stylesheet; the body element, and its h1 element. The title and h1 element content should be *Table of US Presidents*
6. After the h1 element, you will invoke *apply-templates* to all the nodes matching each *president* element (presidents/president). Create the template that matches each *president*, and just for now, output a paragraph whose content is the *value-of* the *President name*.  
At this point, if you view the document in your browser, you should see a list of the US President names, in order.
7. Replace the *president* match template content with creation of a table. Be sure to include columns for the President *name*, *birthday*, *took\_office*, *left\_office*, and *party*. Do this by having the table structure in this template, and then *apply-templates* to each field in turn to create the table cell. Create these templates as well; these should be very simple templates, which place the content in a table cell, with an appropriate class.
8. Update the style sheet to make a reasonable presentation
9. #1 Extra challenge: Can you add the *set* of Vice Presidents for each President nicely??
10. #2 Bonus: Can you construct an element and attribute to make the class for each party match the party name? If so, your CSS can *color* parties by party name

#### II. Sort Presidents By Party

1. Make a copy of the USPresidents.xml file, and the president\_table.xsl files from Problem I
2. Make a one-line change to the copy of president\_table.xsl to make the Presidential table be sorted by party.
3. Your output from this exercise is the transform (xslt file), and a screen shot of the result displayed in one browser

#### III. Twenty-First Century Presidents

1. Begin with the *USPresidents.xml* and *presidents.css* files from a previous problem. Make a copy, because we will change these in new ways, different than the other problem.
2. Attach an XML stylesheet called *president\_21c.xsl* to the XML document
3. Rename the css file to *president\_21c.css*
4. Find (and download) a small portrait of every US President who was in office during the 21st century. Save them all in the same format (e.g., GIF, JPG, PNG), and name them with the number of the US President (e.g., the Bill Clinton portrait would be stored as, say, 42.jpg)
5. Create a transform that is similar to that in the first (or second) problem, except that it must qualify which presidents go to the *apply-templates*, and it must add a column for the US President portrait. The pictures you've saved become the *src* attribute for the *img* tag in your table cell.
6. Modify the *president\_21c.css* stylesheet (and, if necessary, your images) to make a reasonable presentation, including the new columns. In particular, the image cells should be presented sized alike.

## Table of US Presidents

Name	Birthday	Took Office	Left Office	
George Washington	1732-02-22	1789-04-30	1797-03-04	no party
John Adams	1735-10-30	1797-03-04	1801-03-04	Federalis
Thomas Jefferson	1743-04-13	1801-03-04	1809-03-04	Democra
James Madison	1751-03-16	1809-03-04	1817-03-04	Democra


Possible problem 1 fragment

## Table of US Presidents

Name	Birthday	Took Office	Left Office	
Andrew Jackson	1767-03-15	1829-03-04	1837-03-04	Democr
Martin Van Buren	1773-02-09	1837-03-04	1841-03-04	Democr
James K. Polk	1795-11-02	1845-03-04	1849-03-04	Democr
Franklin Pierce	1804-11-23	1853-03-04	1857-03-04	Democr

Possible problem 2 fragment

## Table of 21st Century US President

Name	Birthday	Took Office	Left Office	Party	P
George W. Bush	1946-07-06	2001-01-20	2009-01-20	Republican	

Possible problem 3 fragment

### Notes

- The *apply-templates* instruction is given explicitly for this exercise. The use of *for-each* is prohibited.
- You will find USPresidents.xml and president.css in the HWstarters ZIP file. However extracting the version from the previous assignment (HW3 part 2), where the dates have been converted to standard format is a better starting point.
- You will need to re-integrate the US President's birthdays, as you did in HW2.
- Integration of the presidents and birthdays may be done with or without the namespace declarations you used in HW2. If you use the namespace declarations, you will also apply what you've learned about how XSLT deals with namespaces.
- Your result of the transform must be saved in a file with the .htm or .html extension to display properly in some browsers
- If you have access to an XSLT version 2.0 processor, make the XSL version 2.0 and specify an output method of *XHTML*. If not, and you are using the browser's XSLT version 1.0 processor, specify *HTML* output. XSLT version 1.0 does not support the XHTML output method.
- Adding an element is done by use of the *xsl:element* instruction. Within the body of that instruction, one can add attributes (*xsl:attribute*) and content
- For Problem III, you can either use the *xsl:attribute* method or the alternate curly-brace method to fill in the *src* attribute in the *img* element
- Step 7 of the first problem is essentially a debugging technique you can use. When applying XSLT transforms, it is important to understand your results. Until you have developed an intuition for this, stepping through each level of result is very valuable.
- Hint for the bonus: You will need to make the class for the party cell represent the party name
- For problem 3, you will need to add the function namespace to your XSL transform to be able to use the function that extracts the year from a date
- For the portrait in problem 3, you can use the *brace* notation, or construct the *img* tag with the *src* attribute to be set to the president number with the correct image format suffix.

### Evaluation

Criteria	Weight
Problem I, XML with changes	15
Problem I, XSLT to display President table	30
Problem I, Browser screenshot (one page including title and G. Washington)	10
Problem II, XSLT	10
Problem II, Browser screenshot (one page, including F. D. Roosevelt)	05
Problem III, XML	10
Problem III, XSLT	15
Problem III, Browser screenshot (one page)	05
Problem I, Bonus: Vice Presidents as elements in a column	+6
Problem I, Bonus: Coloring by party	+10