ICT26 Web Development

Tutorial 2
Cascading Style Sheets

Overview

This week's lab exercises:

- Understand the syntax of CSS.
- Understand the different types of CSS selectors
- Be able to understand use appropriate CSS properties and values.
- Use floating property to create multicolumn layout.

Guidelines

- Create a folder "ICT726_Tutorial2" in your computer for this tutorial.
- Download the files provided in the resources of "Tutorial 2" on Moodle and put them into "ICT726_Tutorials/Tutorial1".
- The images related to the lab are available in the image folder.
- You can use any type of text or HTML editor (such as VScode, Sublime, Notepad++, Brackets, etc.) to write your code. If you would like to try an online editor, check https://vscode.dev/

Note: Since absolute links will only work on your computer, we recommend checking your website to make sure all the links work and use the same structure on your home computer.

Fonts/Text Definitions

Definition	*relates to*	*example*
font-family	typeface.	`h2 {font-family: Arial;}`
font-style	the style of the text normal, italic, small caps etc	`h3 {font-style: small caps;}`
font-size	the size of the text in points (pt), inches(in) pixels(px) or %	`h4 {font-size: 20pt;}`
font-weight	text density – extra light, light, demi-light, medium, bold, demi-bold, extra-bold	`a:link {font-weight: demilight;}`

font-variant	variation of the normal font – specify normal or small caps	`h2 {font-variant: small-caps;}`
text-align	the alignment of text – left, centre or right	`h1 {text-align: center;}`
text-decoration	other text formatting – italic, blink, underline, linethrough, overline, or none	`a:visited {text-decoration: blink;}`
text-indent	margins – most often used with if used with use closing tag ! values are “in”, “cm” or “px”	`p {text-indent: 1in;}`
word-spacing	the amount of spaces between words - values are pt, in, cm, px or %	`p {word-spacing: 10px;}`
letter-spacing	space between letters - values are pt, in, cm, px or %	`p {letter-spacing: 2pt;}`
text-transform	transformation of the text – capitalise, uppercase or lowercase	`p {text-transform: uppercase;}`
color	colour of text.	`h3 {color: #FFFFFF;}` `h3 {color:rgb (255, 255, 255);}` `h3 {color: navy;}`

Margin/Background Definitions

Definition	*relates to*	*example*
margin-left margin-right margin-top	space around the "page". Values are in pt, in, cm, or px used with ***body*** entire page is affected used with ***p*** only paragraph is affected	`body {margin-left: 2in;}`
		`p {margin-right: 12cm;}`
		`body {margin-top: 45px;}`

margin	all three margin command combined. Sequence is top, right, left	`p {margin: 3in 4cm 12px;}`
line-height	space between lines of text. Values are pt, in, cm, px or %	`p {line-height: 10px;}`
background-color	page's background colour in hexadecimal or word code – "transparent" is also a value	`body {background-color: #ffffff;}`
background-image	the background image for pages specify URL where image is kept	`body {background-image: url(image.jpg);}`
background-repeat	how the image will tile – repeat-x, repeat-y, or no-repeat	`body {background-repeat: repeat-y;}`
backgroundattachment	how the image will react to a scroll.	`body {background-attachment: fixed;}`

Position/Size Definitions

Definition	*relates to*	*example*
position	the placement of an element on the page. Values are absolute (placement) or relative (to other images)	`img {position: absolute;}`
left	amount of space allowed from the left of the browser screen when positioning an item – values are pt, in, cm, px or %	`img {position: absolute; left: 20px;}`
top	amount of space allowed from the top of the browser screen when positioning an item – values are pt, in, cm, px or %	`img {position: absolute; left: 20px; top: 50px;}`
width	width of image or page division – values are pt, in, cm, px or %	`img {position: absolute; left: 20px; top: 50px; width: 300px;}`

height	height of image or page division – values are pt, in, cm, px or %	`img {position: absolute; left: 20px; top: 50px; width: 300px; height: 200px;}`
overflow	what to do when an image or text is too large – values are visible, hidden, scroll	`img {position: absolute; left: 20px; top: 50px; width: 300px; height: 200px; overflow: hidden;}`
z-index	an item's position in the layering structure – the lower the number, the lower the layer	`img {position: absolute; left: 20px; top: 50px; width: 300px; height: 200px; overflow: hidden; z-index: 10;}`

Exercises 1: Adding Styles

In this exercise you will apply style to your webpage using inline, embedded or external CSS. You can use the Exercise1.html file available in "tutorial2_resources\exercise1\". The final completed webpage should look similar to the website shown in Figure 2:

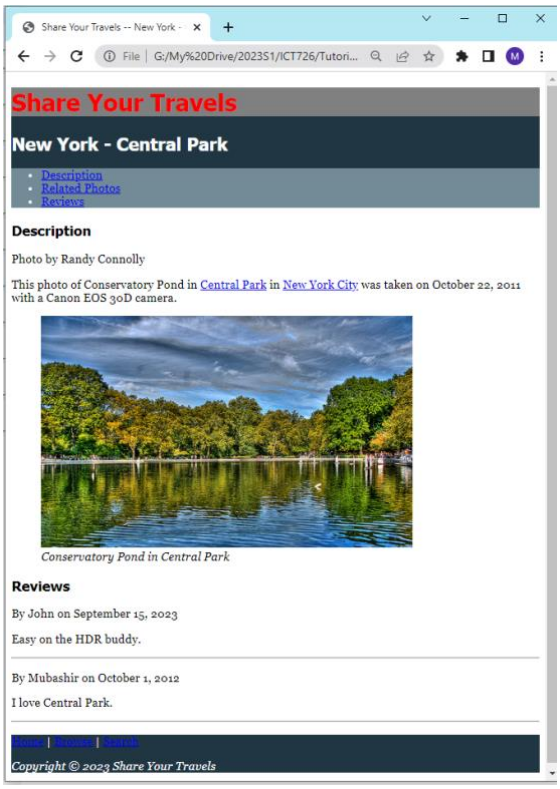

Figure 1Completed Exercise 1

Inline Style:

1. Open, examine, and test exercise1.html in browser. Change the font color of h1 element to red using inline style.

   <h1>** Share Your Travels **</h1>

2. Change the background-color to gray.

***Note:** You are allowed to use only the inline CSS.*

Embedded Style:

3. Now remove the inline CSS and change the text color to red and background color to gray using the embedded style to the **<head>** element from the previous steps.
4. Now change the font-family for h1, h2 and h3 using the grouped selector and test

```
h1, h2, h3 {font-family: "Trebuchet MS", "Lucida Grande", Tahoma, sans-serif;}
```

5. Add the following style rule after the one created in previous step and test

```
h1 {font-family: Georgia, Cambria, "Time New Roman", sans-serif;}
```

Note: *The new style for h1 overrides the earlier one due to the cascade principle of location (i.e., when rules have the same specificity, then the latest are given more weight)*

6. Change the previous style rule for **h1** to the following and test.

```
body {font-family: Georgia, Cambria, "Time New Roman", sans-serif;}
```

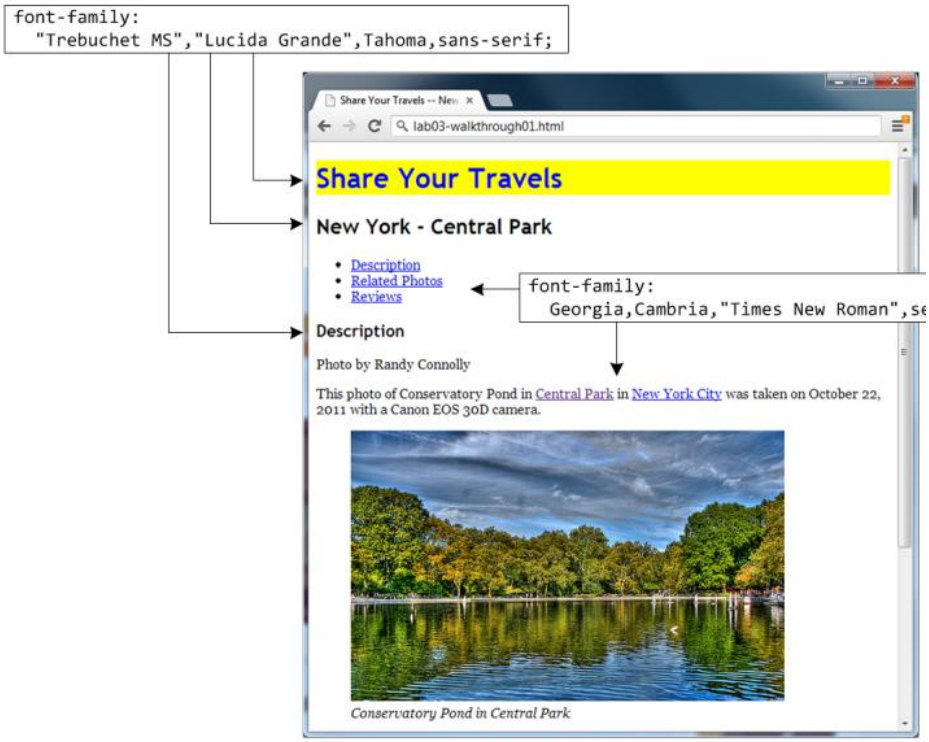

Figure 2 Embedded Style Exercise 1

External Style Sheet:

7. Create a new text document with the following content in the same folder of exercise 1 and save it as exercise1.css.

```
header, footer {color: white; background-color: #213643;}

nav {background-color: #728B96;}

h1, h2, h3 {font-family: "Trebuchet MS", "Lucida Grande", Tahoma, sans-serif;}
```

```
body {font-family: Georgia, Cambria, "Times New Roman", serif;}
```

8. In exercise1.html add the **link element** under <head> tag. Save the file and test in the browser. You output should look like Figure 1

```
<link rel="stylesheet" href="exercise1.css"/>
```

Exercises 2: Elements, Class, ID, Attribute and Pseudo selectors

In this exercise you will learn to use different CSS selectors to select one or multiple HTML elements. apply style to your webpage using inline, embedded or external CSS. You can use the Exercise2.html file available in "tutorial2_resources\exercise2\". The final completed webpage should look similar to the website shown in Figure 3:

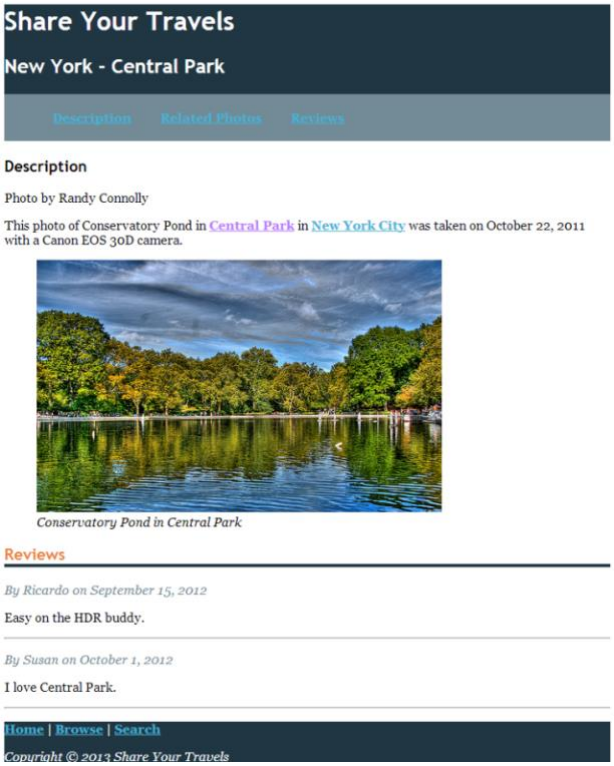

Figure 3 Exercise 2 Completed

ID Selector:

1. Open, examine, and test exercise2.html in browser. Add the "**id"** attribute to **h3** element for review.

```
<section>
    <h3 id="reviews"> Reviews </h3>
    <div>
```

2. To change the style based on id selector then add the following style in the exercise2.css and test.

```
#reviews {
border-bottom: solid 3pt #213643;
color: #ed8030;
}
```

Class Selector:

3. Now let's add the class attribute in the review section paragraphs as shown below

```
<section>

        <h3 id="reviews">Reviews</h3>
        <div>
          <p class="first">By Ricardo on <time> September 15, 2012</time></p>
          <p>Easy on the HDR buddy. </p>
        </div>
        <hr/>
        <div>
          <p class="first">By Susan on <time>October 1, 2012</time></p>
          <p>I love Central Park.</p>
        </div>
        <hr/>
</section>
```

4. To change the style using CSS, you need to use dot with class name. Add the following style to your exercise2.css and test.

```
.first {
color: #728B96;
font-style: italic;
}
```

Attribute Selector:

5. An attribute selector provides a way to select HTML elements either by the presence of an element attribute or by the value of an attribute. Add the following to exercise2.css and test. Did you see any change?

```
[title] {
cursor: help;
 text-decoration: none;
 padding-bottom: 3px;
```

```
border-bottom: 2px dotted blue;}
```

6. Modify the attribute to the following and test [title=" Central Park"]. This selects only the one element whose title attribute is exactly Central Park.

7. Modify the attribute (add the asterisk) to the following and test [title*=" Central "]. This selects all elements that contain the text Central within in.

8. Modify the attribute (add the caret) to the following and test [title^="Central"]. This selects all elements that begin with the text Central.

Pseudo-Class Selector:

9. The next exercise illustrates the use of pseudo-class selectors, which do not apply to an HTML element, but targets either a particular state or, in CSS3, a variety of family relationships. The most common use of this type of selectors is for targeting link states.

10. Add the following style to the exercise2.css and test.

```
a:link { font-weight: bold; color: #47B3D7; }
a:visited { color: #BB78FF; }
a:hover { background-color: #FFFF99; }
```

Styling the navigation bar:

11. Majority of the website has the navigation bar with some padding, no style and are inline. Try the following style and see the difference in the navigation bar

```
nav ul li {
 list-style: none;
display: inline;
margin: 1em;
 }
```

12. No add some padding and margin

```
nav {padding: 0.25em;}   // This adds space within the <nav> element
nav a {padding: 0.25em;}  //This makes <a> size same as the container.
```

Exercises 3: Contextual Selectors

In this exercise you will learn about contextual selectors that allow you to select elements based on their ancestors, descendants, or siblings. The use of contextual selectors is infrequent but almost all web developers may use descendant selectors.

1. Open “exercise3.htm” in browser. Add the following style to exercise3.css file and observe the change.

```
p {color: #983C2A}
```

This changes the text color of every occurrence of the <p> element

2. Now replace the “p” selector with “section p” and observe the change. Identify which of the paragraph elements are not change. Explain why?

3. In this step try to only change the <p> elements that are in the . Which selector should we use in this case?

4. Modify the style using child selector (>) and notice the change.

```
section>p {color: #983C2A}
```

5. Modify the style as follows using the adjacent sibling selectors (+) and test.

```
h3+p {color: #983C2A}
```

6. Modify the style again as follows and test. Write down the elements selected with the following selector?

```
h3~p {color: #983C2A}
```

Exercises 4: Box Model and Fonts

In this exercise you will learn about box model, which defines how an element exist within an element box. This model is used to refer the margin, border, padding and content. Understanding of the box element will help you identify how the size and position of elements are determined. The CSS properties within the element box are shown in Figure 4.

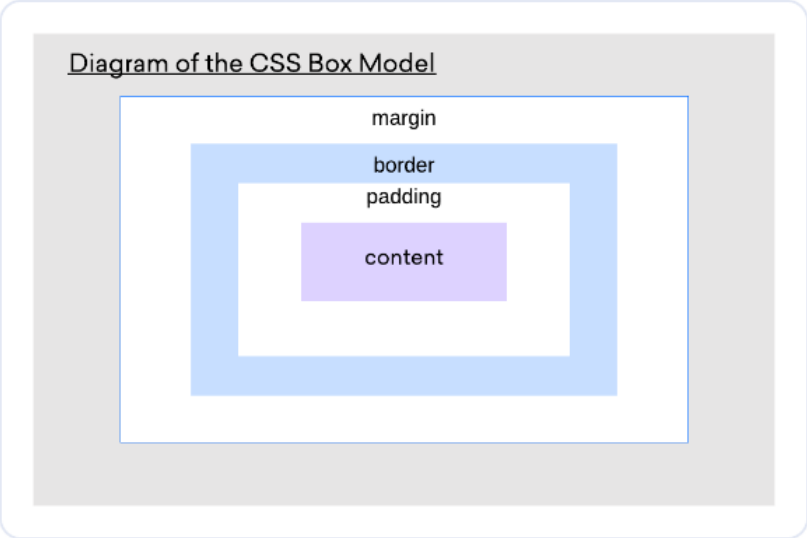

Figure 4 CSS Box Model

1. Open exercise4.html and view in the browser. Let's starts remove all the default browser styles by using the following style **to the very top** of the file. Test in the browser.

```
header, footer,  nav, main, article, section, figure, figcaption, h1, h2, h3,  ul, li, body, div, p, img
{          margin: 0;
           padding: 0;
           font-size: 100%;
           vertical-align: baseline;
           border:  0;

}
```

2. Now add the following style to the figure and test in browser

```
margin: 2em;
```

3. Change the background color of the figure to #EEEEEE. Add the padding of 1.5em and notice the difference in the browser. Apply the border across the figure. Your border should have to be solid, 1px and color should be #999999.

Exercise 5: Float

In this exercise you need to create a page full of thumbnail pictures, together with centred caption. Add a new page in exercise 1 files called gallery.html. In the body of that page add four images (download and save the images of your choice) like the following code

```
<section>
    <h1>Exercise 5</h1>
    <p>Floating Images</p>
    <p>Notice that this p tag does not align center?</p>
</section>
<section>
    <div class="floating">
        <img src="float1.jpg" width="250" height="180" alt="image 1" />
        <p>Floating Image 1</p>
    </div>
    <div class="floating">
        <img src="float2.jpg" width="250" height="180" alt="image 2" />
        <p>Floating Image 2</p>
    </div>
    <div class="floating">
        <img src="float3.jpg" width="250" height="180" alt="image 3" />
        <p>Floating Image 3</p>
    </div>
    <div class="floating">
        <img src="float4.jpg" width="250" height="180" alt="image 4" />
        <p>Floating Image 4</p>
    </div>
</section>
```

Make sure you give the appropriate name to your images and create a floating class. In your style sheet you can use the following CSS code and observe what happens when the size of browser is changed.

```
div.floating {
    float: left;
}

div.floating p {
    text-align: center;
}
```

Exercise 6: Positioning

In this exercise you need to create two classes that will help you create a two-column layout. You need to demonstrate your understanding of **class and id** selectors in CSS.

In you html file create two sections (<section class="leftcolumn">) for left and right columns. Add few headings and paragraphs in each section.

Now create a .css file with following code and demonstrate two sperate columns.

```
.leftcolumn {
    position: absolute;
    width: 200px;
    height: 300px;
    left: 10px;
    padding: 10px;
    color: white;
    background-color:#ccc;
}

.rightcolumn {
    position: absolute;
    width: 400px;
    height: 300px;
    left: 240px;
    padding: 10px;
    color: red;
    background-color:#eee;
}
```

Exercise 7: Background Image

In this exercise you need to add a repeated background image in exercise 1 file. Name your image "bg.jpg" and place it in same location as your HTML and CSS. You can use the following css code.

```
body {
    background-image: url(bg.jpg);
    background-repeat: repeat-y;
}
```

Try different variation of background of repeat and property to show their differences.

~~~~~~~~~~ END ~~~~~~~~~~
~~~~~~~~~~