# Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ StudentID: \_\_\_\_\_\_\_\_\_\_\_

## Activity 1: Introduction to Cascading Style Sheets (CSS)

In the lecture, we covered three ways of including **styles** (or **stylesheets**) into a html document. They are: **inline style**, **embedded style** and **external style**.

|  |
| --- |
| 1. <html> 2. <head> 3. <title>Webpage Title</title> 4. <link rel='stylesheet' href='style.css'/> 5. <style> 6. #content span { color:red; } 7. .more { color:rgba(0,255,0,0.5); } 8. </style> 9. </head> 10. <body> 11. <div id='content'> 12. <h1 style='color:navy;'>Arbitrary Heading</h1> 13. <p>Arbitrary paragraph containing <span>words</span>.</p> 14. <p><span class='more'>... more words here!</span></p> 15. </div> 16. </body> 17. </html> |

**Line 4:** What are the benefits for having external stylesheets?

It would be easier to manage and find classes/ ids to make changes

Having a separate file would also make it easier to read

**Line 6 & 7:** What do the '#' and **'**.' characters refer to in CSS? What type of selector is used on line 6?

# is to create an id for css. It cannot be repeated

. is to create a class which is used for multiple elements in html.

Selector type: Descendant selector

**Lines 11 & 14:** What is the difference between class and id selectors? Are there any restrictions on the number of elements that can have the same class or id?

Class can be used for different elements while id is more specific (one element)

**Line 14:** What color will this span's text be?

Text will have red color

## Activity 2: CSS Pseudo Classes and Elements

To denote a **pseudo class**, we use the colon character : — "*Pseudo" in Greek means "false" or "deception".*

What is a **pseudo class** or **element**?

It will modify depending on the state of the element

Explain how the following pseudo class selectors affect hyperlinks.

|  |
| --- |
| a:link { ... }  a:visited { ... }  a:hover { ... }  a:active { ... } |

link – modify link

visited – modify visited link (link that was accessed before)

hover – only modify the link when user hover the mouse over the link

active – only modify currently visiting link

What benefits are there when grouping according to the **L**o**V**e **HA**te rule?

* **l**inked and **v**isited states together? It will only modify when link was visited while require fewer coding lines
* **h**over and **a**ctive states together? It will modify when the link is active (after hovering)

|  |
| --- |
| /\* Order matters when styling hyperlinks: Remember the LoVe HAte rule! \*/  a:link, a:visited { ... }  a:hover, a:active { ... } |

What will the following pseudo class selectors do to list and table row elements?

|  |
| --- |
| /\* Make list elements more readable \*/  li:first-child { ... }  li:last-child { ... }  /\* Make table row elements more readable \*/  tr:nth-child(2n+1) { ... } |

Modify the last and final elements in the list

Modify the odd rows

To denote a **pseudo element** we use two colon characters ::

Pseudo elements are like pseudo classes except that a new "virtual" element is created. This is how pseudo classes different from pseudo elements.

Describe the elements affected and the pseudo elements that are created with the following styles:

|  |
| --- |
| /\* Make paragraph page elements more enticing \*/  #leader p::first-line { ... }  .story p::first-letter { ... }  /\* Increase the impact of blockquote elements \*/  blockquote::before {  content: "\201C" ;  font-size: 300%;  margin-right: 0.2em;  }  blockquote::after {  content: "\201E";  font-size: 300%;  margin-left: 0.2em;  } |

The paragraph styles are also examples of **contextual** selectors. Explain what this means? Is there a limit on the number of id='leading' and class='story' blocks we can have in the webpage?

Contextual allow us to modify select smaller and more specific group of elements depending on their positions rather than all elements. There is no limit to class but there is limit to id.

Name some new HTML5 pseudo states. How do these new pseudo states reduce work for web developers?

They don’t have to create another id/class and add it to each state of the element

## 

## 

## 

## Activity 3: Replacing Class Selectors with a Contextual Selector

If you over-rely on class selectors, an "inevitable" typo is bound to be made (see example below). How can switching to a contextual based solution reduce the number of class attributes below and make the styling foolproof?

**WARNING:** THIS IS A COUNTER EXAMPLE, ie an example of bad code. Don't copy! See lab for better code!

|  |
| --- |
| a.navlinks:link, a.navlinks:visited {  /\* Non-interacted css styles here \*/  }  a.navlinks:hover, a.navlinks:active {  /\* Interacted css styles here \*/  }  <nav>  <ul>  <li><a href='home.php' class='navlinks'>Home</a></li>  <li><a href='contact.php' class='navlinks'>Contact</a></li>  <!-- Uh oh ... typo below! -->  <li><a href='shop.php' class='badlinks'>Shop</a></li>  </ul>  </nav> |

We can have li a instead so we don’t have to type in the name of the class every time

## 

## 

## 

## Activity 4: CSS Attribute Selectors

What type of hyperlink does the following css selector style? What is displayed and where?

|  |
| --- |
| a[target='\_blank']::after {  content: url('external-icon.png');  } |

## Hyperlink that doesn’t reach to an existing page. It will display the image after a

## 

## 

## 

## Activity 5: Color Models

What colors do #DB8 (12 bit color) and #DDBB88 ( 24 bit color) represent?

Medium/ light shade of brown

How many colors can be represented with these two formats?

300

What does each character in the color format represent?

The hexadecimal for rgb scale

Are #GGA and #GGGGAA real colors? If not, why not?

No, G doesn’t exist in hexadecimal

What is the difference between rgb(255, 100, 255) and rgba(255, 100, 255, 0.5)?

Different level of transparency

How does the **HSL** color model work and what advantages does it have over the **RGB** color model?

Change color based on Hue, Saturation and brightness.

User can visualize easier and don’t have to remember the numbers like RGB.