CS381
Web Application Development

CSS3

These class notes are based on the material from our textbook,
Learning PHP, MySQL & JavaScript, 5th ed., by Robin Nixon

Types of Stylesheets

1. Inline stylesheet applies to a single element; written in the style attribute of the element.

```
<div style='font-style:italic; color:blue;'> Hello there </div>
```

2. Internal stylesheet written in the head section of the html

```
<style> h1 { color: red;} </style>
```

3. External stylesheet written in a separate file .css

- Using import directive — stylesheets import stylesheets. <style> **@import url('styles.css');** </style>

```
@import "styles.css";
or
@import url("styles.css");
```

- Importing CSS from within HTML **<link rel='stylesheet' href='styles.css'>**

The **at-rule** is a statement that provides CSS with instructions to perform or how to behave.

Tag, ID & Class Selector

Tag to specify the styles for the tags.

```
body {color: purple;}
```

ID to specify a style for a single unique element.

```
#p1 { text-align: center; }
<p id="p1"> My Car</p>
```

Class to specify a style for a group of elements.

```
.map {text-align: center;}
<p class="map"> My Car</p>
<h1 class="map" My Map </h1>
```

CSS Rules

Semicolons: separate multiple CSS statements and not needed for one or the last one.

CSS Rules: selector { property-name : property-value; }

Comments: /* */

Specified the same property to the same selector twice:

```
h1 { color : red; }
h1 { color : blue; }
```

The last value specified is applied (blue).

Style Types (Order):

Default Styles → User Styles → External Stylesheets → Internal Styles → Inline Styles

Override: give the selector a higher specificity

```
p { color:#ff0000 !important; }
```

When an !important rule is used on a style declaration, this declaration overrides any other declarations.

The precedence is always given to the more specific selector.

```
a {color:red}
a.som {color:blue}
```

is more specific selector

Type Selector

Descendant Selector applied to elements that are contained within other elements.

<i>*Hello*<i>there `p b { color:red; }`

Child Selector applied only to elements that are direct children of another element.

<i>***Hello***<i>there `p > b { color:red; }`

ID Selector `#p1`

Class Selector `.map` `h1.map`

Attribute Selector saves from having to use IDs and classes to refer to them

`[type="submit"] { width:100px; }`

Universal Selector wildcard or universal selector matches any element

`* { border:1px solid green; }` Everything - bad one

Selecting by Group `p , #time , .map { border-bottom:1px dotted orange; }`

Selector Examples

Universal	* { color:#555; }
Type	b { color:red; }
Class	.classname { color:blue; }
ID	#id { background:cyan; }
Descendant	span em { color:green; }
Child	div > em { background:lime; }
Adjacent sibling	i + b { color:gray; } are children of the same parent
General Sibling	div ~ p { background-color: yellow; }
Attribute	a[href='info.htm'] { color:red; }
Pseudoclasses	a:hover { font-weight:bold; }
Pseudoelements	p::first-letter { font-size:300%; }

Display Measurements

Pixel its size varies according to the dimensions and pixel depth of the user's screen.

```
.map { margin:5px; }
```

margin: used to create space around the borders of the elements.

em equal to the current font size

```
.map { font-size:2em; }
```

ex equivalent to the height of a lowercase letter x of the current font size. less popular

```
.map { width:20ex; }
```

Percent 100% is equivalent to 1 em.

```
.map { font-size:200%; }
```

Print Measurements

Point	equivalent in size to 1/72 of an inch	**.map { font-size:14pt; }**
Inche	equivalent of 72 points	**.map { width:3in; }**
Centimeter	a little over 28 points.	**.map { height:2cm; }**
Millimeter	1/10 of a centimeter (or almost 3 points).	**.map { font-size:5mm; }**
Pica	equivalent to 12 points.	**.map { font-size:1pc; }**

font-size:37px;
font-size:28pt;
font-size:0.39in;
font-size:1cm;
font-size:10mm;
font-size:2.37pc;
font-size:2.37em;
font-size:5.28ex;
font-size:237%;

block-level, inline, inline-block

Every HTML element has a default display value block or inline.

Block-level elements ** , <p>**

- Start on a new line
- Fill up the horizontal space left and right on the web page.
- Margins and padding can be added on all four sides of any block element - top, right, left, and bottom.

Inline elements ** , , <img>**

- Appear on the same line as the content and tags beside them.
- Space can be added to the left and right. **hello**
- Height can not be added to the top or bottom, padding or margin.

Inline-block elements has to be declared

- Similar to inline elements. but,
- Padding and margins can be added on all four sides.

Margins outermost level of the box model. It separates elements from each other.

- **{display: inline;}**
- **{display: block;}**
- **{display: inline-block;}**
- **{display: none;}**

Pseudoclasses & Pseudoelements

box-sizing defines how the width and height of an element are calculated: should they include padding and borders, or not.

- **content-box** the size will be element's width and height + border and padding values.

  width + padding + border = actual width height + padding + border = actual height

- **border-box** padding and border are included in the width and height.

  a { box-sizing: border-box; }

Pseudoclasses

```
a:hover { background: red; }
a:link { color: blue; }
a:visited { color: lightblue; }
a:active { color: darkblue; }
*:focus { border: 2px dotted #888888; }
```

Pseudoelements

```
.p1::first-letter {font-size: 400%;}
.p1::first-line{ text-transform: uppercase;}
```

vs

Both elements are containers.

** ** has infinite width.

applying border to the whole paragraph in docx

**** is as wide as the text it contains, and follow text or other objects as they wrap around them.

applying border to the text in docx

Responsive site the layout changes dynamically depending on the size and orientation of the screen.

W3.CSS

- A modern CSS framework with built-in responsiveness.
- Supports responsive mobile by default.
- Smaller and faster than similar CSS frameworks.
- Speeds up and simplifies development.
- Easier to learn and use than other frameworks.

W3 Containers

w3-container	*adds a 16px left and right padding to any HTML element.*
w3-panel	*adds a 16px top and bottom margin and a 16px left and right padding to any HTML element.*
w3-display-container	a container for w3-display-classes that *allows to display elements in specific positions inside other HTML elements.*

```
<html><title>W3 Containers</title><link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css"><body>
  <div class="w3-container w3-teal">                    <h1>Header</h1></div>
  <div class="w3-container">                            <p>The w3-container can be used to headers.</p></div>
  <div class="w3-panel w3-green">                       <p>I am a panel.</p></div>
  <div class="w3-panel w3-blue w3-round-xlarge">        <p>London is the most populous city in the UK.</p></div>
  <div class="w3-container w3-red">                     <p>Container with w3-red.</p></div>

  <article class="w3-container">          <h2>Paris</h2><p>The Paris area is one of the largest in Europe.</p></article>
  <section class="w3-container">          <h2>Tokyo</h2><p>Tokyo is the center of the Greater Tokyo Area.</p></section>

  <div class="w3-panel w3-red w3-display-container"><span onclick="this.parentElement.style.display='none'"
       class="w3-button w3-red w3-large w3-display-topright">x</span><p>Click on the X to close this panel.</p></div>

  <button class="w3-button w3-red" onclick="document.getElementById('id01').style.display='block'">Show Panel</button>

  <div id="id01" class="w3-panel w3-green w3-display-container" style="display:none">
    <span onclick="this.parentElement.style.display='none'" class="w3-button w3-red w3-display-topright">x</span>
    <p>Click on the X to close this panel.</p></div>

  <footer class="w3-container w3-teal"> <h5>Footer</h5><p>Footer information goes here</p></footer>
</body></html>
```

W3 Cards

W3 provides the following classes for displaying cards:

Class	Defines
w3-card	Same as w3-card-2
w3-card-2	Container for any HTML content (2px bordered shadow)
w3-card-4	Container for any HTML content (4px bordered shadow)

Jeddah

Old Jeddah

Jeddah

New Jeddah 2

Jeddah

New Jeddah 3

Jeddah

New Jeddah 4

W3 Cards

Viewport the user's visible area of a web page.
Initial-scale controls the zoom level when the page is first loaded.
Flexbox aligns and distributes space among items in a container.

```
<!DOCTYPE html><html><head><title>W3 Cards</title><meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
<style>
        * { box-sizing: border-box; }
        .cards { display: flex; flex-wrap: wrap; width:100%; justify-content: space-between; }
        img { width:100%; }
</style></head>

<body>
<div class='cards'>

  <div class="w3-container">
   <div class="w3-panel w3-teal"><h4>Jeddah</h4></div>
   <div class="w3-card-4" >
        <img src="Jeddah/J1.jpg" >
        <div class="w3-container w3-center"> <p>Old Jeddah</p></div>
    </div>
  </div>

</div>
</body></html>
```

W3 More

W3 provides W3-table, W3-ul for tables and lists.

It also provides the following classes for navigation bars:

Class	Defines
w3-bar	Horizontal container for HTML elements
w3-bar-block	Vertical container for HTML elements
w3-bar-item	Container bar elements
w3-sidebar	Vertical sidebar for HTML elements
w3-mobile	Makes any bar element mobile-first responsive
w3-dropdown-hover	Hoverable dropdown element

w3-top	forces the navigation bar to stay at the top of the page.
w3-bottom	forces the navigation bar to stay at the bottom.
w3-cell	makes elements Side-by-side; same as inline-block

W3 Nav

cdnjs a free and open-source content delivery network service. (fa: font awesome)

```
<html><title>W3 Nav</title><meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"><body>
<div class="w3-top">
<div class="w3-bar w3-blue w3-border">
  <a href="#" class="w3-bar-item w3-button w3-mobile"><i class="fa fa-home"></i></a>
  <a href="#" class="w3-bar-item w3-button w3-mobile"><i class="fa fa-search"></i></a>
  <a href="#" class="w3-bar-item w3-button w3-mobile"><i class="fa fa-envelope"></i></a>
  <a href="#" class="w3-bar-item w3-button w3-mobile"><i class="fa fa-sign-in"></i></a>
  <input type="text" class="w3-bar-item w3-input" placeholder="Search..">
  <a href="#" class="w3-bar-item w3-button w3-green">Go</a>

  <div class="w3-dropdown-hover">
    <button class="w3-button">Dropdown</button>
    <div class="w3-dropdown-content w3-bar-block w3-card-4">
      <a href="#" class="w3-bar-item w3-button">Link 1</a>
      <a href="#" class="w3-bar-item w3-button">Link 2</a>
      <a href="#" class="w3-bar-item w3-button">Link 3</a>
    </div>
  </div>
</div>
</div>
Line<br>Line<br>Line<br>Line<br>Line<br>Line<br>Line<br>Line<br>Line<br>Line<br>Line<br>Line<br>Line<br>Line<br>Line<br>
</body></html>
```

W3 Sidebar

```
<!DOCTYPE html><html><title>W3  Sidebar</title>
<meta name="viewport" content="width=device-width,  initial-scale=1">
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
<body>

<div id="mySidebar" class="w3-sidebar w3-bar-block w3-border-right" style="display:none">
   <button onclick="w3_close()" class="w3-bar-item w3-large">X</button>
   <a href="#" class="w3-bar-item w3-button">Link  1</a>
   <a href="#" class="w3-bar-item w3-button">Link  2</a>
   <a href="#" class="w3-bar-item w3-button">Link  3</a>
</div>

<button class="w3-button w3-teal w3-xlarge" onclick="w3_open()">☰</button>

<div>
LineLineLineLineLineLineLineLine<br>LineLineLineLineLineLineLineLine<br>
</div>

<script>
      function w3_open() { document.getElementById("mySidebar").style.display  = "block"; }
      function w3_close() { document.getElementById("mySidebar").style.display  = "none"; }
</script>

</body>
</html>
```

Positioning Elements

Absolute position: position an element according to the distance from its parent element. The object will rest either on top of or behind other elements, depending on z-index property, the highest displayed on top.

```
#object {   position: absolute;
            top :100px;
            left :200px;
            z-index:100;
}
```

Relative position: position an element relative to the location it would occupy in the normal document flow.

```
#object {   position: relative;
            top :10px;
            left :10px;
}
```

Fixed position: position an element within the current browser viewport. When the document is scrolled, the object remains exactly where it has been placed.

```
#object {
            position: fixed;
            top :0px;
            left :0px;
}
```

Positioning Elements

- **static** default, all elements are in order as they appear in the document.
- **relative** element positioned relative to its normal position.
- **absolute** element positioned absolutely to its nearest positioned parent.
- **fixed** element positioned relative to the viewport.
- **sticky** element positioned based on the user's scroll position.
  acts like relative until an element is scrolled beyond a point, then turns into fixed,
  causing the element to stick to its position instead of being scrolled out of view.

float places an element on the left or right side of its container, allowing text and inline elements to wrap around it. `li {float: right; }`

clear specifies on which sides of an element floating elements are not allowed to float.

`li {clear: both; }` or `li {clear: right; }` nothings floats on its right.

rounded borders `border-radius: 60px;`

Shorthand Rules

Shorthand Rules

Concatenating a group of related CSS properties into a single assignment.

```
border-width: 2px;
border-style: dotted;
border-color: #ff8800;
```

instead **`border: 2px dotted #ff8800;`**

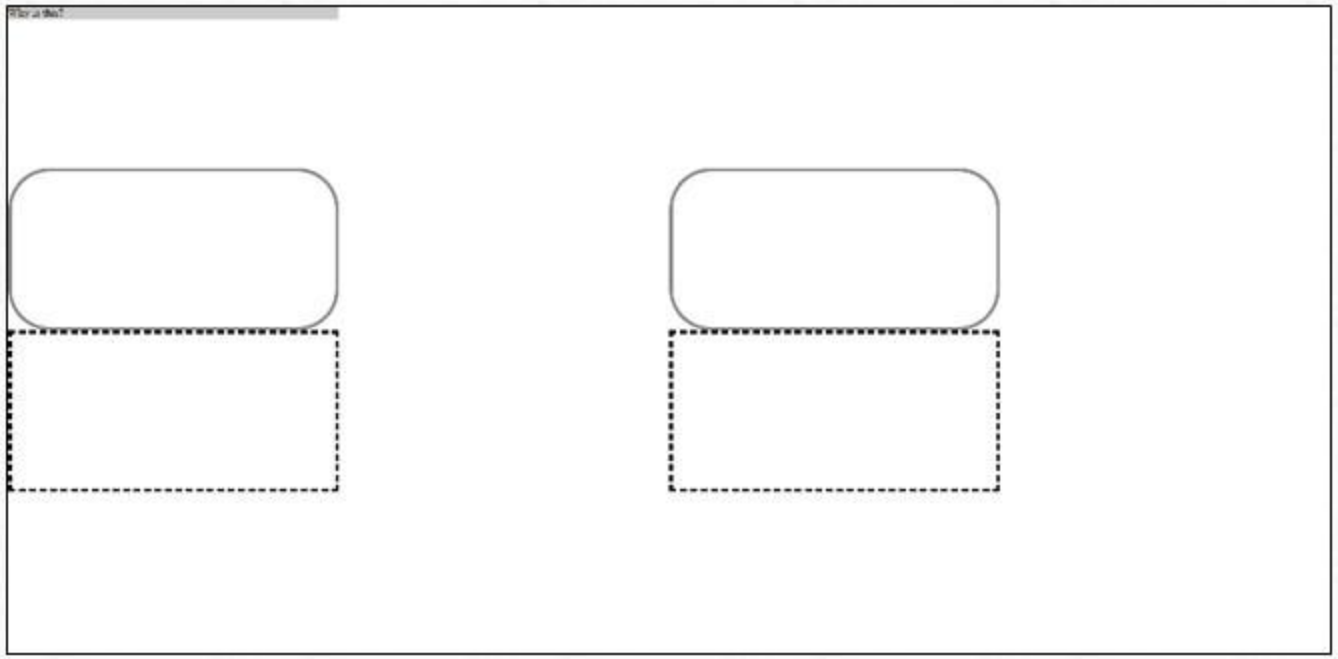
```
background-color: #555;
background-image: url(images/01.png);
background-repeat: no-repeat;
background-position: left top;
```

instead **`background: #555 url(images/01.png) no-repeat left top;`**

Apply the properties up to the point where you wish to change values, the order in which the properties are placed in a shorthand rule could be important - not always.

Position Boxes

```
<!DOCTYPE html><html> <head> <title> Position Boxes </title>

<style>
      * { box-sizing: border-box; }
      body{ margin: 0;}

      .a, #a { position: absolute;      top:25%; width:25%; height: 25%; border: #888 5px solid ; border-radius:60px;}
      .b, #b { position: fixed;         top:50%; width:25%; height: 25%; border: 5px dashed #000;}
      .c, #c { position: relative;      top:500%; width:25%; height: 25%; background-color:#ccc;}

</style> </head>

<body>
      <div class="b"></div>
      <div id="b" style="right:25%"></div>
      <div class="c">Why is this?</div>
      <div class="a"></div>
      <div id="a" style="right:25%"></div>
</body>

</html>
```



First Letter Selector

```
<!DOCTYPE html><html><head> <title> First Letter Selector </title>
<style>
        h1 { color:green  !important;}
        h1 { color:red;}
        *:focus { border:10px dotted #888888; }
        a:active { color:darkblue; }
        p::first-letter { font-size:200%; }
        .para1::first-line{ text-transform: uppercase;}
</style></head><body>
        <h1> green or red </h1>
        <p><a href="https://www.google.com/"> google  </a></p>
        <div style="border:4px solid green; background-color:yellow;">hello there</div>
        <span style="border:1px solid green;">hello you </span>
        <p class="para1" style="border:1px solid green;">hello ooo</p>
</body></html>
```

Questions

1. Which directive do you use to import one stylesheet into another (or the <style> section of some HTML)?
2. What HTML tag can you use to import a stylesheet into a document?
3. Which HTML tag attribute is used to directly embed a style into an element?
4. What is the difference between a CSS ID and a CSS class?
5. Which characters are used to prefix (a) IDs and (b) class names in a CSS rule?
6. In CSS rules, what is the purpose of the semicolon?
7. How can you add a comment to a stylesheet?
8. Which character is used by CSS to represent any element?
9. How can you select a group of different elements and/or element types in CSS?
10. Given a pair of CSS rules with equal precedence, how can you make one have greater precedence than the other?

1. @import url('styles.css');
2. <link rel='stylesheet' href='styles.css'>
3. use style attribute
4. ID is applied to a single element, class applied to many elements.
5. #ID and .class
6. used as a separator between declarations.
7. /* and */
8. using the * universal selector.
9. placing , between each element, ID, or class.
10. append the !important to the declaration

Effects & Transformation

box-shadow: h-offset v-offset blur color; box-shadow:15px 15px 10px #888;

text-shadow: h-offset v-offset blur color; text-shadow:3px 3px 4px #444;

Opacity img { opacity: 0.5; } img:hover { opacity: 1.0; }

web font .className { @font-face { font-family:FontName; src:url('FontName.otf'); } }

Transformation allows rotating, scaling, moving, and skewing elements.

- transform: rotate(45deg); rotates an element around a fixed point (origin) on the 2D plane.
- transform: scale(1.5, 2) rotate(45deg);
- transform: perspective(200px) rotateX(10deg) rotateY(20deg) rotateZ(30deg);

Square shape created using a simple div element with a 1px border

perspective releases an element from 2D space and creates a third dimension. The value defines how far the object is away from the user. So, a lower value will result in a more intensive 3D effect.

Transition

the animation effect we want to occur when an element is transformed. The browser automatically takes care of all the in-between frames.

- transition-property: width; — width, height, opacity, color,…, all
- transition-duration: 2s; — time taken.
- transition-timing-function: ease-in; — ease, linear, ease-in, ease-out, ease-in-out.
- transition-delay: 0.2s; — waiting time before the transition effect starts.
- transition: all 2s linear .2s;
- transition: all 2s;

Transformation & Transitions

```
<!DOCTYPE html><html> <head> <title>Transformation and Transition</title><style>
        div {       width: 200px;
                    height: 200px;
                    background: red;
                    box-shadow: 10px 10px #888;
                    text-shadow:8px 8px 2px #fff;
                    transition: width 1s;
        }           div:hover { width: 600px; }
        img {       margin: 100px;
                    transition: all 2s;
        }           img:hover { transform: rotate(360deg) scale(2, 2); }</style> </head>
<body>
        <img src = "SmallCar.png">
        <div> My New Car </div>
</body></html>
```

Animation

```
<!DOCTYPE html> <html> <head> Animation</title>
<style>
    div
    {     width: 40px;
          height: 40px;
          background-color: blue;
          position: relative;
          animation: divframes linear 20s infinite;
    }
    @keyframes divframes
    {     0%   {left: 0px;    top: 0px;    background: orange;}
          25%  {left: 1000px; top: 0px;    background: red;}
          50%  {left: 1000px; top: 500px;  background: green;}
          75%  {left: 0px;    top: 500px;  background: blue;}
          100% {left: 0px;    top: 0px;    background: black;}
    }
</style></head><body> <div></div></body>
</html>
```

Animation:

The process of changeing an element from one style to another.

animation: keyframes| timing-function| duration| iteration-count

animation: frames linear 10s infinite;

Skewing

```
<!DOCTYPE html><html><head> <title>Skew</title>
<style>
        .skew
        {    background: lightgreen;
             height: 100px;
             width: 200px;
             text-align: center;
             font-size: 500%;
             border: 3px solid DarkGreen;
             border-radius:15px;
             animation: frames 3s infinite linear;
        }

        @keyframes frames
        {     from   { transform: skew(0deg, 0deg);   }
               25%  { transform: skew(0deg, 45deg); }
               50%  { transform: skew(0);                }
               75%  { transform: skew(-45deg, 0deg);}
               to      { transform: skew(0);                }
        }

 </style>
</head>
<body>  <div class = "skew">YUC</div>   </body></html></html>
```

Home News Contact About

```
<!DOCTYPE html><html><head><title>Menu</title><style>
  body { margin: 0; }

  #menu { list-style-type:  none;
          margin: 0;
          padding: 0;
          overflow: hidden;
          background: rgb(139, 18, 18); }

  #menu li       { float: left; }
  #menu li.right { float: right; }

 #menu li a {display: block;
          color: white;
          text-align:  center;
          padding: 14px 16px;
          text-decoration: none; }

  #menu li a.active { background: #888; }
  #menu li a:hover:not(.active)  { background: #111; }

  .view {     height: 400px;
          padding:0 16px;
          background:#aaa; }

   @media screen and (max-width:  600px) {  /* Media Queries: on screens that are 600px or less */
          #menu li.right, #menu li { float: none; }
   }  </style></head>
```

```
<body>
 <ul id="menu">
   <li><a class="btn  active"  href="#home">Home</a></li>
   <li><a class="btn"  href="#news">News</a></li>
   <li><a class="btn"  href="#contact">Contact</a></li>
   <li class="right"><a  class="btn"  href="#about">About</a></li>
 </ul>


<script>      // change the active class
    var myMenu  = document.getElementById("menu");
    var btns  = myMenu.getElementsByClassName("btn");

    for (var i = 0; i < btns.length;  i++) {
            btns[i].addEventListener("click",  function() {
                    var current = document.getElementsByClassName("active");
                    current[0].className  = current[0].className.replace("  active",  "");
                    this.className  += " active";
            });
    }
</script>
<div class="view">  </div>
</body></html>
```