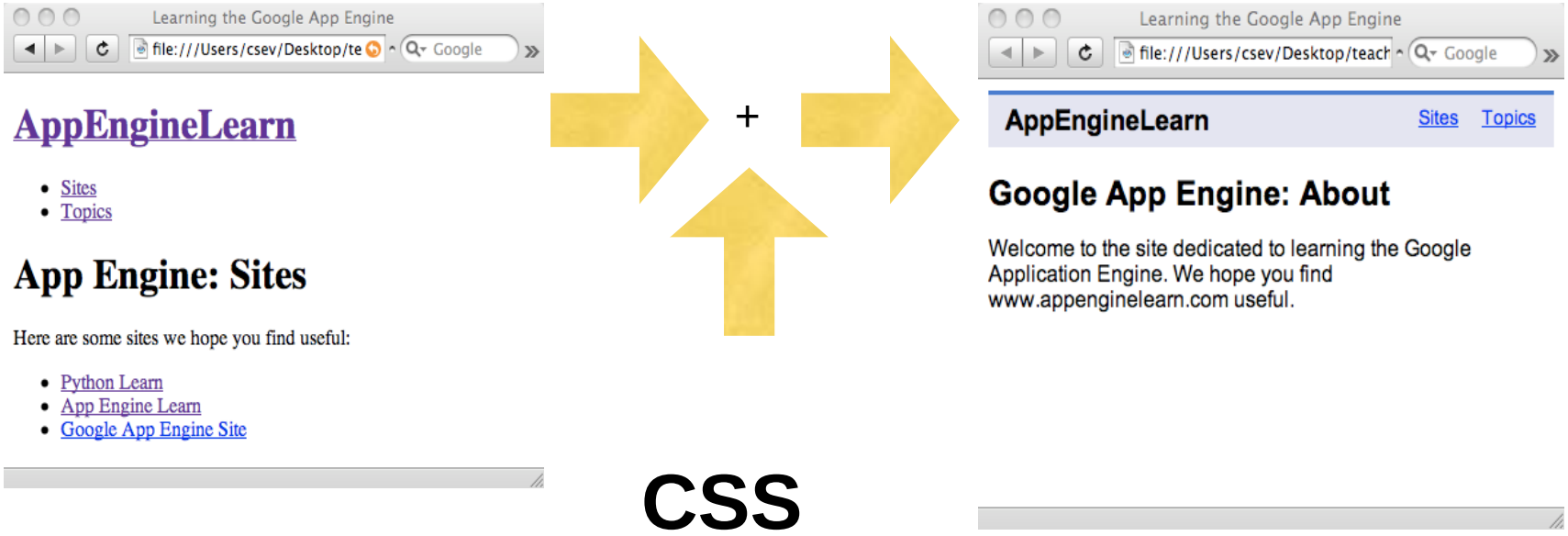
Cascading Style Sheets (CSS)

Lecture # 03

Cascading Style Sheets (CSS)

- CSS is the language for describing the presentation of HTML document, including colors, layout, and fonts.
- CSS describes how HTML elements should be displayed.



Types of CSS

There are three different ways used to insert CSS definitions in a web page. These are:

- **Inline Style -** Add style information to a tag
- **Embedded Style** - Add style information to the document at the beginning
- **External Style Sheet** - Put all of your style in an external file

Inline Styles

- Advantage
  - Useful if you only want a small amount of markup
- Limitations of inline styles
  - do not separate presentation from content and structure
  - mixes display information into HTML
  - to apply similar styles to multiple elements, you need to redefine them for each element
- Solution
  - Use embedded styles sheets or external style sheets
  - External style sheets can save a lot of work

Adding Styles to HTML

```
<head>
<link rel="stylesheet“  type="text/css“  href="myStyles.css" />
  <style type="text/css">
  body {
       background-color:blue;      }
   </style>
 </head>
 <body>
     <div style="padding:2px; ... ">
 </body>
```

Separate style sheet (best way)

Page-specific styles

Element-specific styles

External Style Sheet

```css
body {
  font-family: Tahoma, Arial, sans-serif;
  font-size: 13px;
  color: black;
  background: white;
  margin: 8px;
}
h1 {
  font-size: 19px;
  margin-top: 0px;
  margin-bottom: 5px;
  border-bottom: 1px solid black
}
.shaded {
  background: #d0d0ff;
}
```

CSS:

```html
<body>
  <h1>First Section Heading</h1>
  <p>
    Here is the first paragraph, containing
    text that really doesn't have any use
    or meaning; it just prattles on and on,
    with no end whatsoever, no point to
    make, really no purpose for existence
    at all.
  </p>
  <div class="shaded">
    <h1>Another Section Heading</h1>
    <p>
      Another paragraph.
    </p>
  </div>
</body>
```

HTML:

CSS Rule

selector - which part of the document does this rule apply to

```
body{
    background: white;
    font-size: 100%;     }
```

property - which aspect of CSS are we changing

value - What are we setting the property to.

Multiple tags with same styling

```
h1, h2, h3 {
   color: yellow;
   background-color: black;
}
```

Multiple Style Sheets

- If some properties have been defined for the same selector (element) in different style sheets, the value from the last read style sheet will be used.
- Assume that an external style sheet has the following style for the <h1> element:

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

- Then, assume that an internal style sheet also has the following style for the <h1> element:

```
h1 {  color: orange; }
```

- If the internal style is defined after the link to the external style sheet, the <h1> elements will be "**orange**":

```
<head>
<link  rel="stylesheet"   href="mystyle.css">
<style>
   h1 { color: orange; }
</style>
</head>
```

Multiple Style Sheets

- if the internal style is defined before the link to the external style sheet, the <h1> elements will be “red":

```
<head>
<style>
   h1 {
         color: orange;
         }
</style>
<link rel="stylesheet"    href="mystyle.css">
</head>
```

mystyle.css:

…….

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

CSS Attributes

- There are two attributes with special meaning to CSS
- **id=** - Marks a unique block within the document for styling (use only once)
- **class=** - Marks a non-unique tag within the document for styling (multi-use)

Styling with class=

- class can be used many times in a document.

```
.fun {
  color: #339999;
  font-family: Georgia, Times, serif;
  letter-spacing: 0.05em;
}
```

```
<p class="fun">class can be used many times in a document.</p>
<p>Have a nice day.<p>
<p class=”fun”>More fun stuff</p>
```

Styling with id=

#id can be used only once in a document.

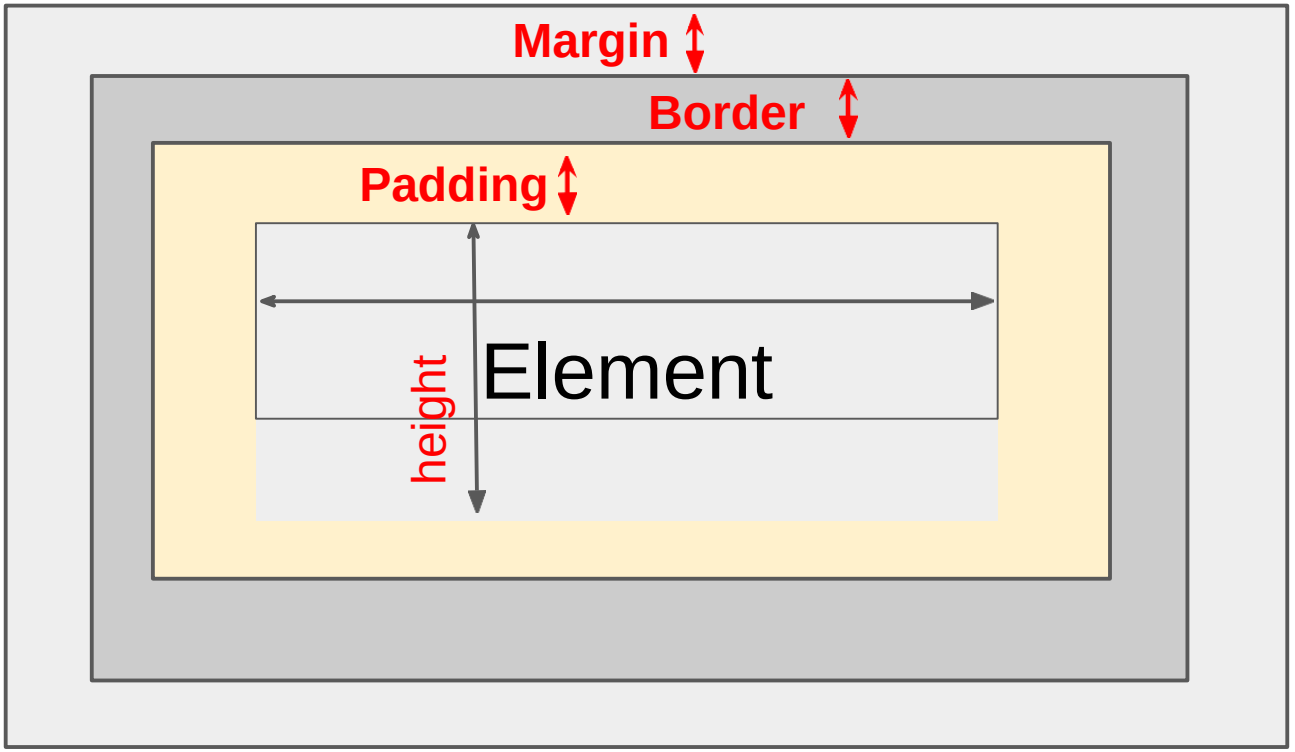
```
#fun {
  color: #339999;
  font-family: Georgia, Times, serif;
  letter-spacing: 0.05em;
}
```

```
<p id="fun"> Have a nice day </p>
```

CSS Properties

- Control many style properties of an element:
  - Coloring
  - Size
  - Position
  - Visibility
  - Many more ……

CSS Box Model



CSS Margins

- **Margins** are used to create space around elements, outside of any defined borders.
- CSS has properties for specifying the margin for each side of an element:
  - margin-top
  - margin-right
  - margin-bottom
  - margin-left
- All the margin properties can have the following values:
  - auto - the browser calculates the margin
  - length - specifies a margin in px, pt, cm, etc.
  - % - specifies a margin in % of the width of the containing element
  - inherit - specifies that the margin should be inherited from the parent element

CSS Padding

- The padding properties are used to generate space around an element's content, inside of any defined borders.
- CSS has properties for specifying the padding for each side of an element:
  - padding-top
  - padding-right
  - padding-bottom
  - padding-left
- All the padding properties can have the following values:
  - length - specifies a padding in **px, pt, cm**, **em,** etc.
  - % - specifies a padding in % of the width of the containing element
  - inherit - specifies that the padding should be inherited from the parent element

CSS distance units	
2px	pixels
1mm	millimeters
2cm	centimeters
0.2in	inches
3pt	printer point 1/72 inch
em	1 em is equal to the current font size. The default text size is 16px.

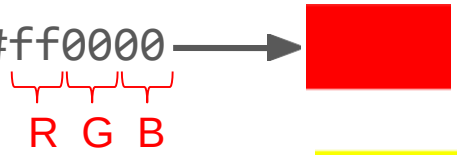
Padding and Margins

```
<!DOCTYPE html>
<html>
<head>
    <style>
        div {  border: 1px solid black;
                background-color: lightblue;
                margin: 30px;
                padding-top: 15px;
                padding-right: 10px;
                padding-bottom: 15px;
                padding-left: 10px; }
    </style>
</head>
<body>
     <div>
       <p>CSS is the language we use to style an HTML document.</p>
     </div>
</body>
</html>
```

Color - Properties: color & background_color

Must ultimately turn into red, green, and blue intensities between 0 and 255:

- Predefined names: `red`, `blue`, `green`, `white`, etc. (140 standard names)
- 8-bit hexadecimal numbers for red, green, blue: `#ff0000`



- 0-255 decimal intensities: `rgb(255,255,0)`

- Percentage intensities: `rgb(80%,80%,100%)`

Example: `h1 { color: red; }`

```
<p style="background-color:blue;"> hello </p>
```

CSS Backgrounds

- The CSS background properties are used to add background effects for elements.
- Backgrounds may be a color, an image, or both.
- CSS background properties:
  - background-color
  - background-image
  - background-repeat

background-color

- The background-color property specifies the background color of an element.
- Example:

```
body {
        background-color: lightblue;
        }
```

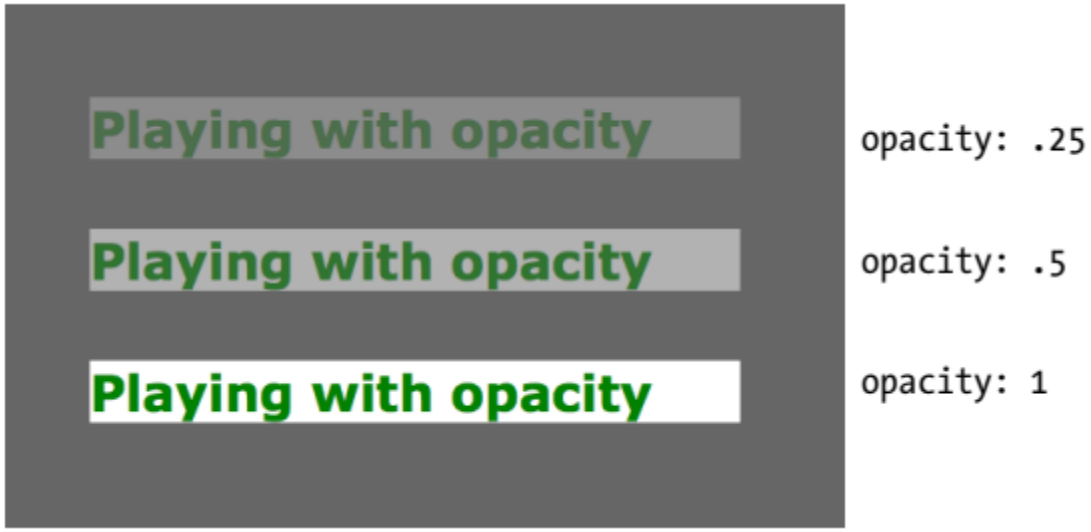
- **Opacity / Transparency:**

The opacity property specifies the opacity/transparency of an element. It can take a value from 0.0 - 1.0.

- Example:

```
body {
        background-color: green;
        opacity: 0.3;
            }
```

Playing with opacity property




```
background: rgba(0, 128, 0, 0.25) /* Green background with 25% opacity */
```

background-image

- The background-image property specifies an image to use as the background of an element.
- By default, the image is repeated so it covers the entire element.
- Example:

```
body {
        background-image: url("paper.gif");
      }
```

```
p {
        background-image: url("paper.gif");
    }
```

background-image

```
<!DOCTYPE html>
<html>
<head>
    <style>
            body {
                background-image: url("paper.gif");
                  }
     </style>
</head>
<body>

     <h1>image background</h1>

     <p>This page has an image as the background!</p>

</body>
</html>
```

background-repeat

- The background-image property repeats an image both horizontally and vertically.
- Example:

```
body {
        background-image: url("paper.gif");
        background-repeat: repeat-x;
    }
```

- To repeat an image vertically, set:

  background-repeat: repeat-y;

- Show the background image only once, set:

  background-repeat: no-repeat;

background-position

- The background-position property is used to specify the position of the background image.
- Example:

```
body {
    background-image: url("img_tree.png");
    background-repeat: no-repeat;
    background-position: right top;
}
```

CSS Borders

- The CSS border properties allow you to specify the style, width, and color of an element's border.
  - borders on all sides.
  - bottom border.
  - rounded borders.
  - left border.

Border Style

- The border-style property specifies what kind of border to display.
- The following values are allowed:

- `dotted` - Defines a dotted border
- `dashed` - Defines a dashed border
- `solid` - Defines a solid border
- `double` - Defines a double border
- `groove` - Defines a 3D grooved border. The effect depends on the border-color value
- `ridge` - Defines a 3D ridged border. The effect depends on the border-color value
- `inset` - Defines a 3D inset border. The effect depends on the border-color value
- `outset` - Defines a 3D outset border. The effect depends on the border-color value
- `none` - Defines no border
- `hidden` - Defines a hidden border

- The border-style property can have from one to four values (for the top border, right border, bottom border, and the left border).

```
<!DOCTYPE html>
<html>
<head>
  <style>
    h1 {
        border-style: dotted;
       }

     div {
          border-style: dotted;
        }
  </style>
</head>
<body>

    <h1>A Heading with a dotted border</h1>

    <div>A div element with a dotted border.</div>

</body>
</html>
```

CSS Text Alignment

Text Alignment: text-align

- The text alignment property is used to set the horizontal alignment of a text.
- Text can be centered, or aligned to the left or right, or justified.
- When text-align is set to “justify”, each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazine and newspaper).

```
h1 { text-align: center; }
p { text-align: left; }
```

CSS Text Decoration

Text Decoration: text-decoration

- The text-decoration property is used to set or remove decoration from text.
- It is mostly used to remove underlines from links for design purpose.
- Values:

  none | underline | overline | line-through

```
h1 { text-decoration: underline; }
a { text-decoration: none; }
```

```
h1 {  text-decoration: underline wavy blue 5px;}
```

CSS Text Decoration

- The text-decoration property is a shorthand property for:
  - text-decoration-line (required)
  - text-decoration-color
  - text-decoration-style
  - text-decoration-thickness

Value	Description
text-decoration-line	Sets the kind of text decoration to use (like underline, overline, line-through)
text-decoration-color	Sets the color of the text decoration
text-decoration-style	Sets the style of the text decoration (like solid, wavy, dotted, dashed, double)
text-decoration-thickness	Sets the thickness of the decoration line

CSS Text Transformation

Text Transformation: text-transform

- The text-transform property is used to specify uppercase and lowercase letters in a text.
- It is can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word.

```
h1 {text-transform: capitalize;    }

h2 {text-transform: uppercase;   }

h1 { font-size: 40px;
     text-transform: lowercase;    }
```

CSS Text Color

Text color: color

- a color name - like "red"
- a HEX value - like "#ff0000"
- an RGB value - like "rgb(255,0,0)“

```
p {color: red;   }
```

- Note: To color a part of a text use the HTML tag. The tag contains a short piece of text within a block

```
<p>This is a <span style="color: blue; font-weight: bold">blue</span> text.</p>
```

This is a **blue** text.

CSS Text Font

Text Font: font-family

- The font-family property is used to specify the font of a text.
- If the font name is more than one word, it must be in quotation marks, like: "Times New Roman".

```
p { font-family: "Times New Roman", Times, serif; }
.p3 { font-family: "Lucida Console", "Courier New", monospace; }
```

Note: The font-family property should hold several font names as a "fallback" system, to ensure maximum compatibility between browsers/operating systems. Start with the font you want, and end with a generic family (to let the browser pick a similar font in the generic family, if no other fonts are available).

CSS Text Font

- **Some Font Examples**

Generic Font Family	Examples of Font Names
Serif	Times New Roman Georgia Garamond
Sans-serif	Arial Verdana Helvetica
Monospace	Courier New Lucida Console Monaco
Cursive	Brush Script MT Lucida Handwriting
Fantasy	Copperplate Papyrus

CSS Text Font

Font Style: font-style

- This property has three values:
  - **normal** - The text is shown normally
  - **italic** - The text is shown in italics

```
<head>
  <style>
     .normal {  font-style: normal; }
     .italic { font-style: italic; }
</style>
</head>
<body>
   <h1>The font-style property</h1>
   <p class="normal">This is a paragraph in normal style.</p>
   <p class="italic">This is a paragraph in italic style.</p>
</body>
```

list-style-type

- Apply the list-style-type property to the ul, ol, or li element select the type of marker that appears before each list item.

- Values:

none | disc | circle | square | decimal | decimal-leading-zero | lower-alpha | upper-alpha | lower-latin | upper-latin | lower-roman | upper-roman | lower-greekdisc

```
Li {   list-style: upper-alpha;   }
```

CSS Links <a> tag

- HTML links (<a> tag) can be styled differently depending on what state they are in. The four links states are:
  - **a:link** - a normal, unvisited link.
  - **a:visited** - defines the style for links that have already been visited in the past.
  - **a:hover** - defines the style for links when the mouse pointer is hovering over them.

    - a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.
  - **a:active** - defines the style for links the moment the user clicks on them.

    - a:active MUST come after a:hover in the CSS definition in order to be effective.

CSS Links <a> tag

```
/* unvisited link */
a:link {   color: red;    }

/* visited link */
a:visited {  color: green;   }

/* mouse over link */
a:hover {  color: yellow;   }

/* selected link */
a:active {  color: blue;   }
```

Element visibility control properties

- **visibility: hidden**; - Element is hidden but space still allocated.
- **visibility: visible**; - Element is normally displayed.
- Hiding an element can be done by setting the display property to none.
- The element will be hidden, and the page will be displayed as if the element is not there

```
h1 {visibility: hidden;  }
```

Display property

- The **display** CSS property sets whether an element is treated as a block or inline element

  - **display:** none; - the element is completely removed
  - **display**: inline; - displays an element as an inline element
  - **display:** inherit; - inherits this property from its parent element
  - **display**: block; - displays an element as a block element
  - **display**: inline-block; - Displays an element as an inline-level block container. the element itself is formatted as an inline element, but you can apply height and width values
  - **display**: flex; -displays an element as an flex container

Display property

```
<head>
<style>
    .main {
            margin:30px; }
     #para1{
                background: red;
               display: inline; }
    #para2{
               background: blue;
               display: inline; }
</style>
</head>
<body>
          <h2> Display Inline Example </h2>

         <div class = "main">
            <div id="para1"> CSS </div>
            <div id="para2"> HTML </div>
         </div>
</body>
```