

HTML5

- Hypertext Markup Language
- Markup tags (also HTML tags) describe the meaning content
- Is semantic – imparts content with meaning
- Used to structure content
- Used to reference and link content

Validating your page

- Valid HTML markup: <http://validator.w3.org/>
- Semantic section structuring: check with outline algorithm:
<http://gsnedders.html5.org/outliner>

CSS3

- Cascading Style Sheets
- Rules describe the appearance of markup tags
- Used to style content
- Combined with HTML used to define layout
- Facilitates content portability^[1]

document styling with CSS

lecture 3

style.css

- Browsers interpret CSS in order to style relevant components of HTML documents
- CSS can be authored “internally” in a document using the `<style>` meta element
- CSS can be authored per element – called “inline” CSS – using the style attribute

- CSS **should** be authored externally to a document
- CSS files are identified via the “.css” file extension
- The <link> element is used to attach external CSS files to an HTML document

```
<link rel="stylesheet" type="text/css"  
      href="">
```

- link is an empty element

```
<link rel="stylesheet" type="text/css"
      href="style.css">
```

- The rel attribute indicates the relationship between the two files
- The type attribute indicates the type of file being linked
- The href attribute indicates the path to the linked file, starting from the folder in which the HTML document is located
- Folders may be separated using the forward slash
- Two consecutive periods indicate the file is in a parent folder

CSS rules

- CSS rules tell the browser how to style an HTML document
- Each rule consists of selectors and declarations
- A selector indicates the part of the HTML document to be styled
- A declaration indicates what about the component should be styled and how
- **CSS rules “cascade” down document structure, meaning all child elements will inherit the styling of their parent element – unless otherwise indicated**

the element selector

- A common selector is the element selector – it indicates that the CSS rule should apply to all elements of a given type in a document

```
h1 { font-family: sans-serif; }
```

- A single rule may consist of **multiple selectors** – they must be separated using **commas** (,)

```
h1, p { font-family: sans-serif; }
```

declarations

- Declarations are authored in a property/value pair, which are separated with a colon (:)

```
h1 { font-family: sans-serif;}
```

- Multiple declarations in a single rule may be separated using the semicolon (;)

```
h1 { font-family: sans-serif; color: blue;}
```

properties and values

```
h1 { font-family: sans-serif; color: blue; }
```

type specification

- A *typeface* or font is a collection of glyphs that share similar visual characteristics
- Normal, italicized, bold, and even size variations of a glyph set are not the same font – they are multiple fonts grouped as a “font family” ([example](#))

- The **font-family** property indicates a font family to be used when setting a document
- The **value** should be the name of a font

```
h1 { font-family: Arial;}
```

- Font **names containing whitespace** must be surrounded by **quotation marks**

```
h1 { font-family: "Comic Sans MS";}
```


generic font families

- Not all font families are available on all devices or browsers
- All browsers are able to display at least one of each generic families

serif, sans-serif, monospace

serifs

- Serif fonts have small lines tailing from the edges of glyphs, sans-serif do not

Serif / Sans-Serif

- Traditionally serif fonts are considered to be more legible in large paragraphs of text, however the effect is diminished with high quality reproduction (such as monitors and high quality printing)

monospace

- In monospace fonts all characters occupy an equal horizontal space, proportional fonts occupy a variable width ([example](#))

Monospace. /
Proportional.

- Monospaced fonts are usually less legible than proportional fonts

font-family fallback

- If the specified font is unavailable, the browser will use it's default font
- If multiple font families are specified in the value, and the browser does not support the first font, it will try to display the next font
- Multiple font-family values must be separated by commas

```
h1 {font-family: "HelveticaNeue-Light", "Helvetica  
Neue Light", "Helvetica Neue", Helvetica, Arial,  
"Lucida Grande", sans-serif;}
```

default font families

- Each browser has a default font which will be the last and automatic fallback – this font is usually a serif font

font weight

- The **font-weight** property Indicates the weight of text ([example](#))

- Accepts values in absolute values

100 to 900

- Absolute keywords

normal (same as 400), **bold** (same as 700)

- Or relative keywords

lighter, bolder

font weight

- The font-weight property Indicates the weight of text

- Accepts values in absolute values

100 to **900**

- Absolute keywords

normal (same as 400), **bold** (same as 700)

- Or relative keywords

lighter, bolder

font slope

- The **font-style** property indicates the slant of a font
- Accepts values in keywords

normal, italic

font size

- The **font-size** property indicates the size of a type
- Size does not address individual glyphs, but rather the maximal size possible for a single glyph – this is called a **type element**

- Accepts values in absolute keywords

xx-small to medium to xx-large

- Relative keywords

smaller, larger

- and CSS measurement units

CSS measurement units

- CSS measurements may be authored in either absolute or relative units:
- Absolute units include: Pixels, Centimeters, Millimeters, Inches, Points, and Picas
- Relative units include: Percentages, EM's, and EX's

absolute CSS units

- **px** – pixels referring to the size of a physical dot on a screen
- **mm, cm, in** – millimeters, centimeters, and inches in reference to the physical world
- **pt, pc** – points and picas, relative to inches (but not really), usually used in printing

relative CSS units

- Relative units designate sizes relative to the size of their parent element
- **em, ex** – referring to the size of a type element – in CSS only the height is considered ([example](#))
- **%** - a percentage of the parent element
 - for font related properties (font-size, word-spacing, line-height) 100% is equal to 1em – that is, it is relative to the font size of a parent element
 - for width, height and other measurements: 100% is equal to the actual size of the parent container, unrelated to the font-size

default font sizes

- Each browser has a designated default font size – usually 16px on desktops and slightly larger on mobile and tablet devices

capitalization

- Text capitalization in CSS is divided in two ([example](#)):
- The **text-transform** property indicates normal capitalization rules
- Accepts keyword values:

none, capitalize, uppercase, lowercase

- The **font-variant** property indicates petite capitals
- Accepts keyword values:

normal, small-caps

styling text

lining text

- The **text-decoration** property is used to underline, over-line, or line-through text
- Accepts keyword values:

none, underline, overline, line-through, blink

- Using blink is not advised, as it decreases legibility

text color

- the **color** property designates the color of text

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

- Accepts keywords – “[color names](#)”

red, green, blue, black, white, etc...

- Accepts CSS color values

CSS color values

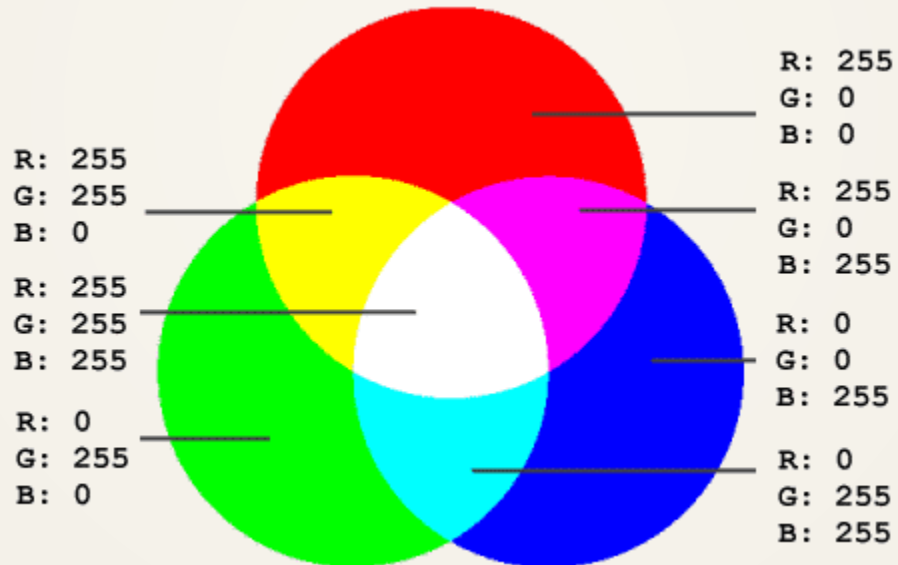
- CSS colors may be authored in RGB or HSL formats
- RGB formats indicate colors as a mixture of red, green, and blue

h1 {color: rgb(255,0,0);} or h1 {color: #FF0000;}

- HSL formats indicate the hue, saturation, and luminosity of a color

h1 {color: hsl(0, 100%, 50%);}

RGB



RGB

- RGB represents color mixture – higher values represent more of a given color

- May be authored in hexadecimal code

#RRGGBB;

- In percentages

RGB (X%, Y%, Z%);

- Or values from 0 to 255

RGB (X, Y, Z);

HSL

H: 0
S: 95
L: 56



HSL

HSL(Hue, Saturation, Luminosity);

- Hue indicates the position of a color on a color wheel – measured in degrees (from 0 to 360)
- Saturation is the amount of hue in a color – indicated as a percentage – 0% is greyscale, 100% is colorfull
- Luminosity is the brightness or lightness of a color – indicated as a percentage – 0% is black, 100% is white, 50% is the original hue

Transparency

- In addition all color values may be appended with an **alpha value** indicating the opacity of a color

rgba (X%, Y%, Z%, .A);

hsla(Hue, Saturation, Luminosity, .A);

design heuristics

- Style
- Accessibility
- Design or usability

designing color schemes

- Accessibility – the ability to read your text content
- Contrast validator: [site](#)
- Appearance: [site](#)

- `*{background:rgba(0,0,0,0.2)}`