Școala
informală
de IT

CSS Properties

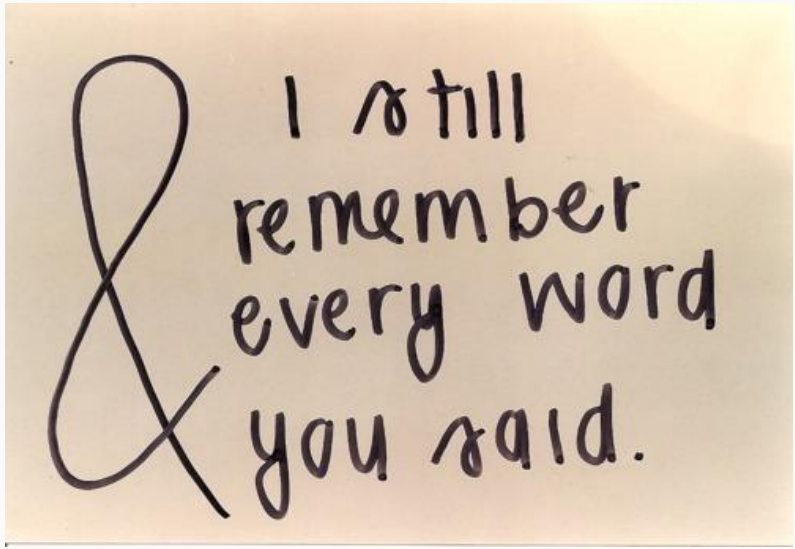
Agenda

- Recap
- Text Styling
- Colors and Backgrounds
- Box Model
- Positioning

Recap

Last session's topics

- HTML Forms
- HTML IFrames
- CSS - what it is and why it's useful
- Linking HTML and CSS
- CSS Syntax
- CSS Selectors (type, class, id, combinators, attribute, pseudo-classes and pseudo-elements)
- Cascading Rules (importance, specificity, source order)



CSS Text Styling

Text Styling

- 2 categories for text styling properties:
  - Font styles
    - what font is applied to the text
    - how big it is
    - whether it is bold, italic, etc.
  - Text layout styles
    - spacing and other layout features of the text
    - the space between lines and letters
    - how the text is aligned within the content box

Font-families

- font-family property
  - applied only if it is available on the machine the website is being accessed on
- web safe fonts (fonts that are generally available across all systems)
  - Arial (& Helvetica)
  - Courier New
  - Georgia
  - Times New Roman
  - Trebuchet MS
  - Verdana
- http://www.cssfontstack.com/

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

Font-families

- **Default fonts**

Term	Definition	Example
`serif`	Fonts that have serifs (the flourishes and other small details you see at the ends of the strokes in some typefaces)	My big red elephant
`sans-serif`	Fonts that don't have serifs.	My big red elephant
`monospace`	Fonts where every character has the same width, typically used in code listings.	`My big red elephant`
`cursive`	Fonts that are intended to emulate handwriting, with flowing, connected strokes.	My big red elephant
`fantasy`	Fonts that are intended to be decorative.	My big red elephant

Web Fonts

- @font-face

```
@font-face {
    font-family: Fancy;
    src: url('Fancy-Bold.ttf');
}

.my_class {
    font-family: Fancy;
    font-size: 3.2em;
}
```

Font Sizes

- font-size property
- Most common units:
  - px (pixels)
  - ems
    - 1 em = the font size set on the parent element of the current element we are styling (the width of a capital letter M contained inside the parent element)
  - rems
    - Just like ems, but it's relative to the font size set on the root element

```
h1   { font-size: 16px; }
p    { font-size: 1.33em; }
span { font-size: 0.83rem; }
```

Other Font Properties

- **font-style** property
  - Used to turn italic text on and off
  - Values: normal, italic, oblique
- **font-weight** property
  - Sets how bold the text is
  - Values: normal, bold, lighter, bolder, 100-900
- **text-transform** property
  - Allows to transform text
  - Values: none, uppercase, lowercase, capitalize, full-width
- **text-decoration** property
  - Sets/unsets text decorations on fonts
  - Values: none, underline, overline, line-through

```
p {
    font-style: italic;
    font-weight: bold;
    text-transform: capitalize;
    text-decoration: underline;
}
```

Text Shadow

- **text-shadow property**
- Takes up to 4 values
  a. The horizontal offset of the shadow from the original text - required (left/right)
  b. The vertical offset of the shadow from the original text (up/down)
  c. The blur radius - a higher value means the shadow is dispersed more widely
  d. The base color of the shadow (defaults to black)

```
p { text-shadow: 2px 2px 7px #000000; }
```

Text Layout Properties

- text-align property
  - Control how text is aligned within its containing content box
  - Values: left, right, center, justify
- line-height property
  - Sets the height of each line of text
  - Values: length and size units, and also a unitless value (multiplies the font-size)
- letter-spacing and word-spacing properties
  - Sets the spacing between letters and words in your text
  - Values: length and size units
- word-break property and overflow-wrap property
  - Word-break is used to specify whether to break lines within words
  - Overflow-wrap - specify if the browser may break lines within words in order to prevent overflow

Text Overflow

- **text-overflow property**

- Determines how overflowed content that is not displayed is signaled to users
- May be specified using one or two values
  a. One => the right end of the line
  b. Two => first: the left end of the line, second: the right end of the line
- Values
  a. Clip - indicates to truncate the text at the limit of the content area
  b. Ellipsis - display an ellipsis to represent clipped text

```
p { text-overflow: clip ellipsis; }
```

Font Shorthand Property

- font property
- Must be written in the following order:
  a. font-style
  b. font-variant
  c. font-weight
  d. font-stretch
  e. font-size
  f. line-height
  g. font-family

```
p { font:
    italic
    normal
    bold
    normal
    3em/1.5
    Arial, sans-serif;
}
```

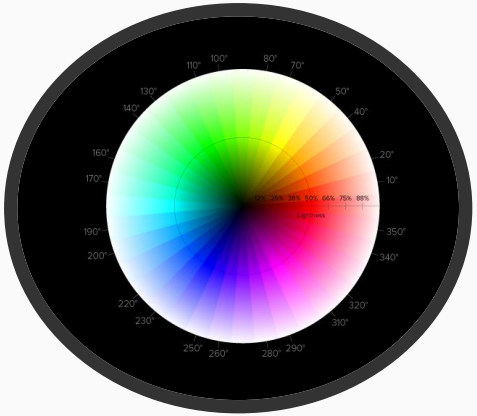
Colors and Backgrounds

Color and Background-color

- **color - the color of the foreground content of the selected elements**
- **background-color - sets the background color of an element (color value or transparent)**
- **Color values:**
  a. **Keywords: red, blue**
  b. **Hexadecimal values: #ff0000; #0000ff;**
     - **6 hexadecimal digits, representing values from 0-255 for red, green and blue**
  c. **RGB: rgb(*red*, *green*, *blue*) - receives values from 0-255 for each parameter**
  d. **HSL: hsl(*hue*, *saturation*, *lightness*)**
     - **hue:** the base shade of the color (0-360, = the angle round a color wheel)
     - **saturation:** 0-100% (0% - no color, shade of gray; 100% full color saturation)
     - **lightness:** 0-100%, where 0 is no light (completely black) and 100% is full light (completely white)

Transparency

- **RGBA & HSLA**
  - add a transparency parameter, called alpha channel: values between 0 - 1
- **Opacity**
  - sets the transparency of all selected elements and their children



```
.p1 { color: rgba(255, 255, 0, 0.5); }
.p2 { background-color: hsla(240, 100%, 50%, 0.5); }
.p3 { opacity: 0.5; }
```

More Background Properties

- background-image property
  - URL of image to be used as background
- background-color property
  - It’s possible to use color and image at the same time
- background-repeat property
  - Specify how the background image is repeated
  - Values: repeat, repeat-x, repeat-y, no-repeat
- background-position property
  - Position the background image wherever we want inside the background
  - Values: the coordinates for the top left corner of the image
- background-attachment property
  - Specifying how the background scrolls when the content scrolls
  - Values: scroll, fixed, local

Background Shorthand Property

- background property
- Properties:
  a. background-color
  b. background-image
  c. background-repeat
  d. background-attachment
  e. background-position

```
div { background:
    #FFF0C0
    url("bg.gif")
    no-repeat
    fixed
    top;
}
```

Background Image or <img>?

- **Background images allow you to save many image tags from the HTML**
  a. **Leads to less code**
  b. **More content-oriented approach**
- **All images that are not part of the page content (and are used only for "beautification") should be moved to CSS**

Background Gradients

- Linear and radial gradients set on background-image property
- linear-gradient() - most used
- Three parameters separated by commas:
  a. the direction the gradient should be going in
  b. the color at the beginning
  c. the color at the end

```
div { background-image: linear-gradient(
    to bottom,
    yellow,
    orange);
}
```

Multiple Backgrounds

- CSS3 allows multiple backgrounds
- Simple comma-separated list of images
- Comma separated list for the other properties

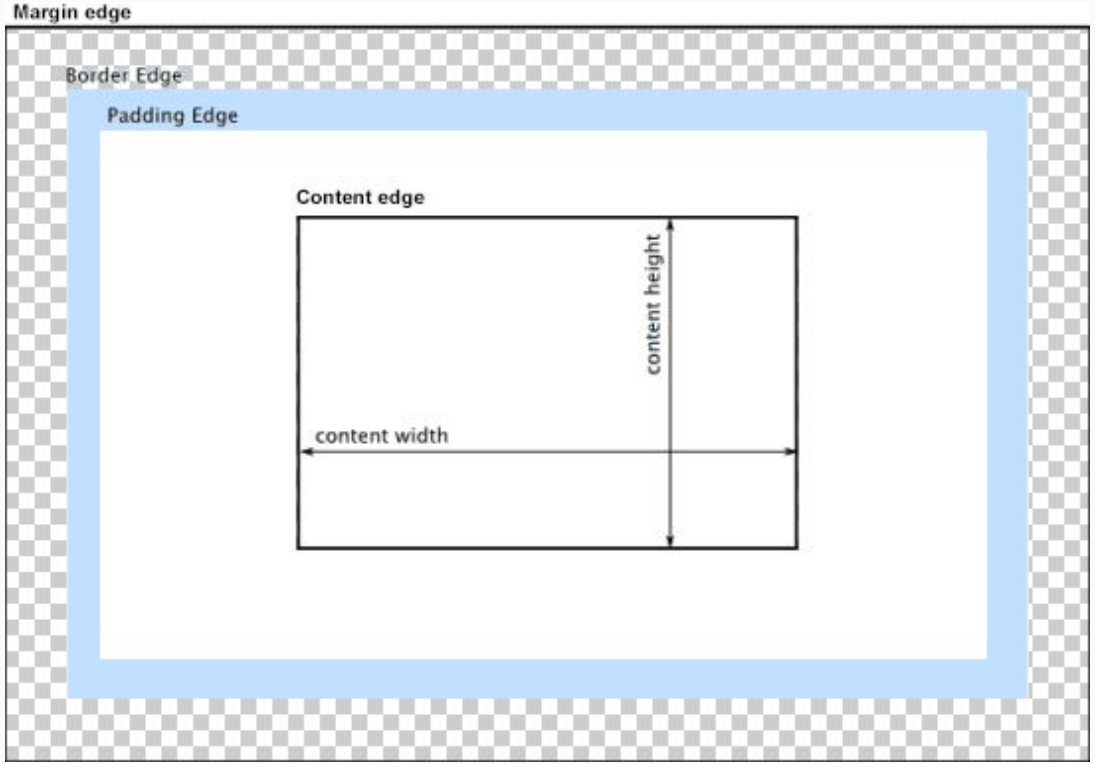
```
div { background-image: url(“sheep.png”), url(“grass.png”); }
```

CSS Box Model

CSS Box Model

- In a document, each element is represented as a rectangular box
- The box model describes the space of the content taken by an element
- Each box has four edges:
  a. Margin edge
  b. Border edge
  c. Padding edge
  d. Content edge

CSS Box Model



Content area

- = the area containing the real content of the element
- Located inside the content edge
- box-sizing property
  a. Alters the way the browser calculates width and height of the elements
  b. If it's set to default (content-box), the content size is controlled by:
     - width
     - height
     - min-width
     - min-height
  c. If it's set to border-box, the width and height include *content, padding and border*

Padding, Border, Margin

- **Padding**:
  - padding
  - padding-top, padding-right, padding-bottom, padding-left
- **Border**:
  - border or border-width
- **Margin**:
  - margin
  - margin-top, margin-right, margin-bottom, margin-left

Collapsing Margins

- Top and bottom margins of blocks are sometimes combined into a single margin whose size is the largest of the margins
- 3 basic cases:
  - Adjacent siblings
  - Parent and first/last child
    - The collapsed margin ends up outside the parent
  - Empty blocks
    - If there is no border, padding, inline content, height, or min-height to separate a block's margin-top from its margin-bottom, then its top and bottom margins collapse.

```
<p>The bottom margin of this p is collapsed</p>
<p>with the top margin of this paragraph.</p>
```

CSS Positioning

Normal Layout Flow

- = the way the browser lays out HTML pages by default
  - Elements are displayed in the order in which they appear in the source code, stacked on top of one another
- To override this behavior, 3 techniques can be used:
  - Floats
  - Position property
  - Display property

Floats

- a technique that allows the elements to float to the left or right of one another, rather than the default of sitting on top of one another
  - Used to lay out columns or float text around an image
- 4 possible values
  - left - floats the element to the left
  - right - floats the element to the right
  - none - no floating at all - default value
  - inherit - property's value should be inherited from the parent element
- Workshop:
  - Create a layout with 3 equal columns using float

Positioning Techniques

- a technique that allows you to move an element from it's original spot on the page to another spot with great accuracy
- 4 main types of positioning
  - static - the default that every element gets
  - relative - move an element relative to its position in normal flow
  - absolute - moves an element completely out of the page's normal layout flow, like it is sitting on its own separate layer
    - Can be fixed in a position relative to the edges of the page's <html> element or its nearest positioned ancestor element
  - fixed - similar to absolute positioning, but fixes an element relative to the browser viewport, not another element

Positioning Techniques - Workshop

- Understand how the following positioning techniques work:
  - Relative
  - Absolute
  - Relative + Absolute
  - Fixed

Display Property

- specifies the type of rendering box used for an element
  - Default values are taken from behaviors described in the HTML specifications or from the browser/user default stylesheet
- A lot of possible values, out of which 3 are very important:
  - inline - no breaks are placed before and after the element
  - block - breaks are placed before and after the element
  - none - element is completely removed from the flow (along with its children)
- Visibility property determines whether an element is visible or not
  - visible - element is rendered normally
  - hidden - element is hidden, but not removed from the flow

CSS Tables

- = a technique that evolved from the times when HTML Tables were used for creating the page layout
- Do not have the issues of HTML tables
  - Inflexible layouts
  - Heavy on markup
  - Difficult to debug
- Created using a set of display properties:
  - display: table
  - display: table-row
  - display: table-cell
  - display: table-caption

Flexible Boxes

- There are situations when positioning and float are quite complex and don’t achieve the expected results:
  - Vertically center a box of content (not just text)
  - Make several columns containing different amounts of content have the same height, without using a fixed height, or faking it with background images
  - Make several boxes in a line take up the same amount of the available space, regardless of how many of them there are, and if they have padding, margin, etc. applied
- display: flex for the win!
- (Short) Workshop: vertical alignment

Z-Index

- Default stacking, when no element has z-index
  - Background and borders of the root element
  - Descendant blocks in the normal flow, in order of appearance (in HTML)
  - Descendant positioned elements, in order of appearance (in HTML)
- Z-Index - used to specify a different stacking order
  - receives an integer value (positive or negative), which represents the position of the element along the z-axis
- Z-index only has an effect on positioned elements

Readings & Tutorials

- Text Styling (MDN)
- CSS Values and Units (MDN)
- Box Model (MDN)
- Old but very good positioning tutorial: http://www.barelyfitz.com/screencast/html-training/css/positioning/
- CSS positioning interactive: https://medium.freecodecamp.com/css-positioning-explained-by-building-an-ice-cream-sundae-831cb884bfa9 & https://www.rtfmanual.io/csssundae/
- Learn layout: http://learnlayout.com/
- Position (MDN) & Positioning example (MDN)
- A Guide To Flexbox (CSS Tricks)