Advanced CSS and Layouts

Making things pretty and getting them in order

Reset&Normalize

What are they?
Reset or normalize files give you a clean starting point for your styles.

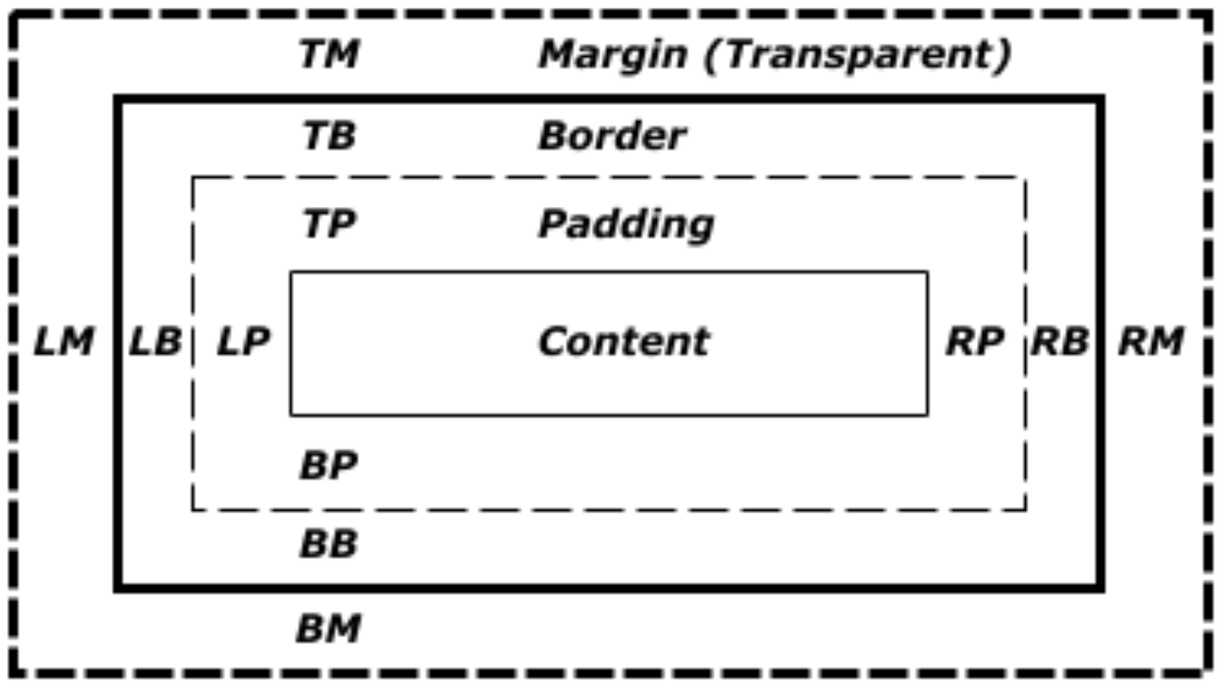
Reset: The common reset from MeyerWeb will take away all the default styles for every HTML tag giving you a clean slate to work with.

Normalize: The normalize file will make all default styles look the same across all browsers so you have the standard reference for each. You can use either one; we will use reset.css in this class for most demos.

Box Model

What is it and why does it matter? (Good question me from the past!)

Each element of your website can be thought of as a box of content consisting of:



Margins & Padding

Margins will control the space around the outside edge of a box.
Padding will control the space around the inside edge of a box.

Padding will actually increase the dimensions of the element that it has been applied to! This is important to keep in mind, especially when you are doing a layout and you are tracking how much space each element takes up.

The border also will increase the dimensions of your elements.

Time to walk through an example!

Display Property

block: Take up the width of the section it is in; can apply margins, paddings, etc...

inline: Do not have width or height, cannot apply margins, paddings, etc...

inline-block: Acts as an inline-element but can be given a width and height

Descendant, Child, AND Compound Selectors

Descendant Selectors (Can also be called Nested selectors)

The following will style all elements with class "warning" that are descendants of elements with class "heading".
.heading .warning { }

The following will style all elements with class "tab" that are descendants of the element within the nav tag.
nav .tab { }

The following style all paragraph elements that are descendants of the element within the footer tag.
footer span { }

Child Selectors

Child Selectors select child elements but not their children.

The following style all elements that are selected by selector2 which are children of elements selected by selector1.

```
selector1 > selector2  { }
```

Example time!

Check out this great article to learn more about child and sibling selectors.
http://css-tricks.com/child-and-sibling-selectors/

Compound Selectors

Compound Selectors

Selectors are very powerful and you can style very specific elements with a compound selector!

```
nav .tab .icon { width: 10px; }
nav .tab .description { font-weight: bold; }
```

These will style the "icon" elements that are descendants of "tab" elements, which in turn are descendants of the element within the nav tag.

Example time!

Complex CSS selectors Lab

Lab Link: http://codepen.io/Adamor/pen/PPoxQK

Select each item with the class 'tagged' and set it's CSS value to whatever is specified in the data-cssVal attribute
The data-cssVal is just an arbitrary attribute, it doesn't do anything

Rules:
You CANNOT modify the markup
You must (somehow) target all the HTML elements purely with CSS selectors

Complete Lab Result

Result

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi eu tortor congue, accumsan sem convallis, cursus erat. Integer vitae nisi metus. Integer volutpat porta justo, vel dictum justo pharetra a.

Ut ultricies dignissim ante, a lobortis nisi congue nec. Maecenas viverra tortor vel diam tincidunt pulvinar. Sed ut massa turpis.

Pellentesque rhoncus augue eget aliquam tincidunt. Nam laoreet dapibus sem, nec mollis urna varius non. Nam non mi enim. Duis ultricies mi in nunc ultrices egestas. Morbi dignissim gravida ornare.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi eu tortor congue, accumsan sem convallis, cursus erat. Integer vitae nisi metus. Integer volutpat porta justo, vel dictum justo pharetra a. Ut ultricies dignissim ante, a lobortis nisi congue nec. Maecenas viverra tortor vel diam tincidunt pulvinar. Sed ut massa turpis. Pellentesque rhoncus augue eget aliquam tincidunt. Nam laoreet dapibus sem, nec mollis urna varius non. Nam non mi enim. Duis ultricies mi in nunc ultrices egestas. Morbi dignissim gravida ornare.

- Test
- Test
- Test
- Test
- Test
- Test
    - Test2
    - Test
    - Test
- Test
- Test
- Test

KNOWLEDGE GAINED FROM OLD MAN...AKA STUFF TO KNOW FOR THE NEXT LAB

Pseudo-Classes

Selector	Description
:first-child	selects an element if it is the first child
:last-child	selects an element if it is the last child
:nth-child(2)	selects every element that is the nth child
:nth-child(odd/even)	selects an element if it is an odd/ even child
:link, :visited, :active, :hover	selects link, visited link, current click link, link about to be click

Pseudo-Selectors

Selector	Description
::first-letter	creates an element wrapping the first letter of enclosed text
::before	creates an element before the selected element
::after	creates an element after the selected element

Fonts

You don’t have to use just web fonts! YAY!

You can include new fonts either using a service like Google, typekit, etc... or locally using the font-face property. We’ll take a look at Google and the font face property today.

Fonts

Google is our friend letting us use fonts outside of the standard web fonts. You may include lots of different fonts on your website by linking to a file online such as Google Fonts.

To embed a font from a service, like Google, grab the url and link to it in the HTML
<link href='http://fonts.googleapis.com/css?family=Roboto +Condensed' rel='stylesheet' type='text/css'>

Then, refer to the font in the CSS font-family: 'Roboto Condensed', sans-serif;

Fonts

You can also add fonts locally, using the @font-face property.

Font-Face: We learned that the property font-family can specify a font to use on your web page. There is also a CSS property called @font-face which you can use to import a font of your choice.

How to add @font-face to your projects:
The name may be anything you want as long as you refer to it (see below). Also, if you’re going to use spaces in the name then you must surround the name in quotes (see “Times New Roman”)

@font-face { font-family: "FEWD69rocks"; src: url(‘myfont.eot?#iefix') format('embedded-opentype'), url('myfont.woff') format('woff'), url('myfont.ttf') format('truetype'), url('myfont.svg#') format('svg'); font-weight: normal; font-style: normal; }

Fonts

Now, in your html, you need to specify font-family.

body { font-family: "FEWDrocks", sans-serif; // sans-serif is the fallback }

Fonts historically could only be displayed in the browser if the font existed on the user's computer. With web fonts and font-face, you can load any font you want, and people will see your fonts without having to have the font on their machine.

Fonts

Font Units - EMS

In typography, an em-dash (—) is equal in width to that of a lowercase “m”. In CSS, ems are relative to the type size. If you apply a font-size in pixels to the body, then assign a font-size in ems to the h1, the h1 size will be relative to the body size, with 1em as the baseline.

In the h1 below, the font-size in pixels of the h1 is 48.
body { font-size: 24px; } h1 { font-size: 2em; } h1 small { font-size: .5em; }

Why use ems or percentages vs pixels? By using a relative unit, it can make it easier to make changes. For example, if the body text size changes from 24px to 20px, the size of other text styles will automatically change if the units are relative to the body text size.

Box Sizing

In typical CSS box-sizing (called content-box), if you specify the width of something, the padding gets added to the outside.

So if you make a div 100px wide, then add padding of 10px all the way around, your box is now 120px wide, even though its width property is set at 100px!

CSS3 Box-Sizing allows you to set the width as the absolute outer-most dimension, and padding gets subtracted from the outer width. So that 100px wide box with 10px padding now has an available width of 80px inside for content. This box sizing method is called border-box.

Box Sizing

Normal CSS

Width: 300px
Padding 10px
Total Width: now 320px

Layout is broken!
You get to recalculate widths
and do work over again!

Border-Box

Width: 300px
Padding 10px
Total Width: Still 300px

Stays within the
confines of your box size

Floats&Clear

Float Property
left: pushes the element to the absolute left
right: pushes the element to the absolute right
Example Time!

Clear Property
Sometimes when you float an element you may run into an issue with alignment. Some elements may collapse or squish. Using a clear method will fix the layout looks the way it was intended to look.

clear: accepts three values to realign elements
left, right, or both

Floats&Clear

This method works fine but developers believe that extra HTML should not be added for styling purposes alone.

Therefore, you may use a method known as clearfix.
Just applying the class ".clearfix" in our case to the containing div instead of creating an empty div.

```
.clearfix:before, .clearfix:after { content: ""; display: table; }
.clearfix:after { clear: both; } .clearfix { zoom: 1; }

<div class=”clearfix”>
<img class=”float_left” src=”my_image.jpg />
</div>
```

Floats&Clear

Why does overflow: hidden clear floats?

Because you establish a new Block Formatting Context when using overflow with anything other than visible.

Remember to use overflow:hidden you'll need to give the style to the element that contains the floating element, not the floating element itself!!

Positioning

Property	Value
static	Default. Elements render in order, as they appear in the document flow
absolute	The element is positioned relative to its first positioned ancestor element
fixed	The element is positioned relative to the browser window
relative	The element is positioned relative to its normal position
inherit	The value of the position property is inherited from the parent element

Positioning

I like to think of the differences between these positioning properties in the following way:

Relative: Set top, right, left and bottom relative to the position this element would normally have been in your regular document flow. (This is kind of like taking the normal, expected position of an element and moving it up 5 pixels, or over 10% from where it would normally be).

Absolute: Set position from the top, left, right and bottom of your browser window. Absolute elements are positioned relative to the upper left-hand corner of a parent element that is not static (often times, this is the body). Absolute elements will scroll with your content.

Fixed: Almost exactly like absolute, but will NOT scroll with your content. Set position from the top, left, right and bottom of your browser window.

Positioning

.square { position: absolute; top:100px; left:200px; z-index:5; }
When you use set positioning you can also use z-index to control which elements appear above or below others. Often uses for layering effects in UIs. Elements with higher numbers will be on top of elements with lower numbers.

Example Time: Play around with it to see how different positioning affects the location of the boxes.

A more practical use of positioning: http://codepen.io/Adamor/pen/ZYLwLj

This example uses absolute positioning for image captions. One of the many ways you could use it for layering.

Layouts FINALLY

When creating a website, you typically have distinct sections:
Header, Navigation, Primary content, Sidebar, Footer
With these powers combined I am "layout"...go LAYOUT!

Layouts FINALLY

Laying things out with Divs

A occupies the entire width of its parent
's get stacked on top of each other like blocks
Each forces a new line
Generally you'll use divs to both wrap around your full site to able global style and to mark up the "divisions" of content.

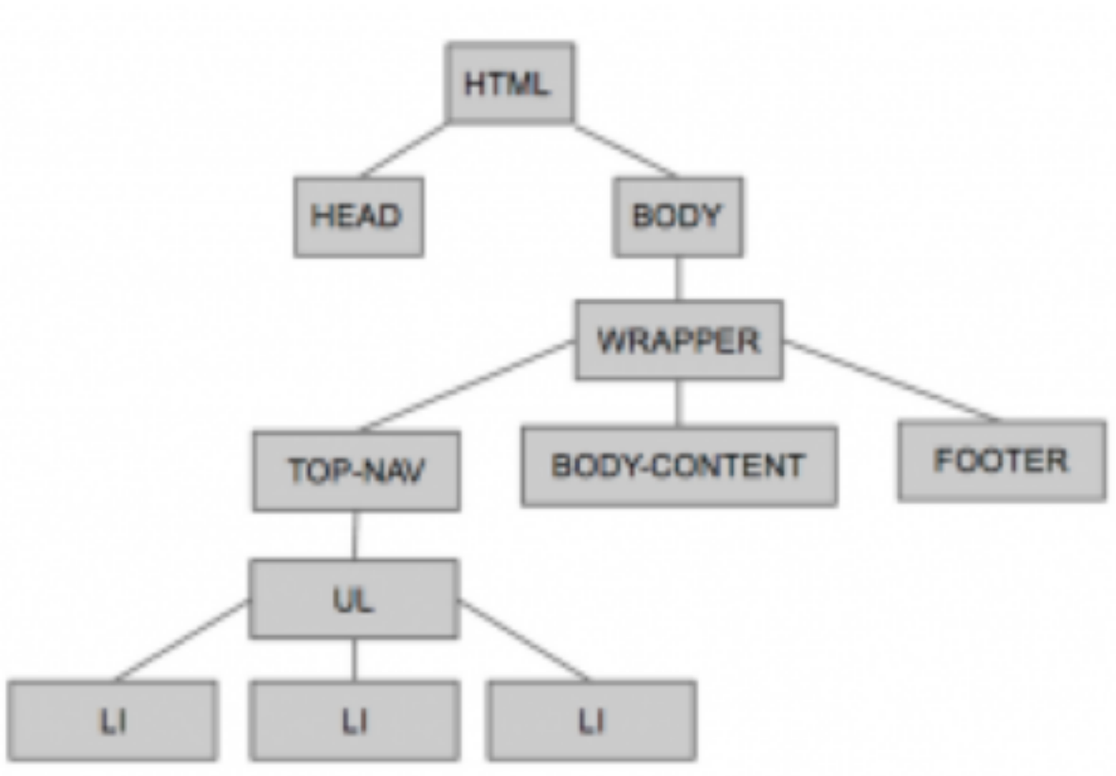
Main content, sidebars, etc...
...

LAB TIME!

What are the requirements?

1. Setting up a proper folder structure for our project
2. Inspecting the design
3. Figure out a DOM Tree for Content layout (DOM Tree...what?)

We actually are setting up a DOM Tree multiple times in this class. Take a look.



HOMEWORK

A hot, new NY Tech Startup, Relaxr, has approached you to help them develop a new landing page for their company. They've handed you the design file for the site, along with the copy and assets. You need to take these files and turn it into a landing page.

HOMEWORK

Real-World Applications

Build a website from a design file
Integrate advanced CSS properties
Use HTML5 structural elements
Use CSS Resets to "normalize" the rendering of your page across different browsers
Use the Box Model to style element borders and structure your page
Create a page with a multi-column layout

HOMEWORK

Requirements:

- Use HTML5 structural elements (nav, header, footer)
- Use a CSS Reset file in addition to your style.css file to style the page
- Use IDs and Classes to to select and style elements on the page
- Style your text with the Google Fonts provided by your style guide
- Follow naming conventions, maintain consistency across .html and .css files and use best practices for naming IDs and Classes
- Indent nested elements to increase your code's readability
- Use display property, float/clear, or positioning to create more than one column on your page

Resources

When you’re done push "/FEWD_yourname/assignmentweek2" to your Github so TAs can grade!