

Feb 14

hi class

hidden styles

let's dive right into code today. split screen Sublime and Chrome. let's type up some html in the body. let's make a header and a div.

```
1 <h1>Hello everyone</h1>
2 <div class="container">
3 </div>
```

let's fill the div with stuff, a `<p>` and a list `` with ``. let's list some things that are nearby or on our bodies.

```
1 <h1>Robby</h1>
2 <div class="content">
3   <p>hello friends</p>
4   <ul>
5     <li>phone</li>
6     <li>pen</li>
7     <li>scarf</li>
8   </ul>
9 </div>
```

this is a good bit of html to start with. look at it in the browser, what do you see in terms of styling?

- the header is bigger than all the text
- the list has bullets
- the list has a margin on the left (actually padding)

let's try to remove all the styling using css. does anyone know how to get rid of the bullets on the list?

```
ul{ list-style:none; }
```

nice! open the inspector, look at all the padding and the margins. there's padding on the ``, and lots of top and bottom margins on things. let's get rid of them all.

```
ul{ padding:0; }
```

here's a trick, you can use the * asterisk as a CSS selector

```
*{ margin:0; }
```

the * is like a placeholder. this targets everything. all the margins went away.

Google: "CSS reset sheet"

- locate for Eric Meyer's 2.0 reset sheet
- copy and paste it into a NEW, EMPTY sublime file
- save it as "reset.css"

The goal of a reset sheet is to take all the styling away. Let's see it in use:

1. create a new file in sublime called "reset.html"
2. copy and paste everything from our first html into this one, and remove all of the CSS. remove EVERYTHING between `<style>` and `</style>`, then get rid of those `<style>` opening and closing tags too.
3. link the css reset sheet by typing inside of the head

```
1 <head>
2   <link href="reset.css" rel="stylesheet">
3 </head>
```

Now look at your html page, see how all the style has been removed - all the margins and font sizes are reset. I love css reset style sheets. This is really useful if you're going to do a serious project. It lets you build up an html and be very specific about the design and there won't be any hidden margins anywhere messing up your stuff.

CSS Selectors

copy and paste the contents of the recent html document (reset.html) again into a new document called "selector.html"

add some more elements:

```
1 <h1>Robby</h1>
2
```

```
3 <div class="content">
4   <p>Hello friends</p>
5   <ul>
6     <li>phone</li>
7     <li>pen</li>
8     <li>scarf</li>
9   </ul>
10 </div>
11
12 <div class="footer">
13   <ul>
14     <li>computer</li>
15     <li>vga dongle</li>
16   </ul>
17 </div>
18
19 <ul>
20   <li>twitter</li>
21   <li>instagram</li>
22 </ul>
```

now, does anyone know how to target only the `` under footer, not all of the ``s?

- use ids or classes.

Yes, that works. Today we're going to learn a better way.

first step: this will target all the ``s inside of a `<div>`, so, it doesn't target the last `` list.

```
1 div ul{
2   font-weight:bold;
3 }
```

now let's target just the `` under the "footer" div.

```
1 div.footer ul{
```

```
2   color:green;
3 }
```

Open this page:

https://www.w3schools.com/cssref/css_selectors.asp

don't worry about it! we're not learning it right now, but look at how you can get super specific about what you are targeting. Things that have really specific nesting order, things that have a certain attribute, things where the class name contains lots of characters but begins with the first 4 "page".

Q: does this work without `<!DOCTYPE html>`, what is that?

A: It tells the browser we're using HTML version 5. check out this page:

https://en.wikipedia.org/wiki/HTML#HTML_versions_timeline all the version history release dates of HTML. HTML 4 in 1997. HTML5 in 2014.

curiosity

make a `<robby>` element, with some text in it, and close it

```
<robby>hello friends</robby>
```

does it show up? yep!

now go to the CSS of your page, and try to style it.

```
1  robby{
2    color:green;
3 }
```

does it style? Yep! what? CSS is a smart language, and it is able to interpret what you want, but `<robby>` is outside of the HTML spec.

Google Fonts

goto: <http://fonts.google.com>

pick a nice font! any font. if you hit the plus sign, it selects it. notice the bottom thing that popped up for a second, click on it to expand it up. notice there are 2 lines of code.

to use a Google font you need 2 lines of code:

- one HTML
- one CSS

```
1 <link href="https://fonts.googleapis.com/css?family=Josefin+Sans" rel="stylesheet">
2 font-family: 'Josefin Sans', sans-serif;
```

we copy the line into the header. and where do we copy the font-family? it's CSS, so we need to open

do we paste it right in there? no we need like a h1 or a div or something. let's apply this font to everything in the html document.

```
1 html{
2   font-family: 'Josefin Sans', sans-serif;
3 }
```

it looks so much nicer, right!?

What happens if I disconnect my internet and try to load this page? It can't load it, but it did load a sans serif font instead of "Times New Roman". Why? That's because the ", sans-serif" bit at the end of the CSS line. you can stack fonts. so often you might see more than just 2, like this line:

```
font-family: 'Josefin Sans', 'Helvetica', sans-serif;
```

Helvetica is a popular font, many computers have it, but it's not guaranteed. it makes a good fallback, but at the very end of the list you want to have "sans-serif". So when you use font family list specific fonts to generic fonts in that order.

Font weights

Go back to Google Fonts and select a font that has font weight options. select a few, go back to the HTML command and see how it added numbers after a colon ":100,400,700". But also, notice how the load time has decreased. This is important to keep in mind there is a trade off, including more fonts increases the load time. Pretty insignificant load time but important to keep in mind.

let's use the font weights

```
1 h1{
2   font-weight: 100;
3 }
4 div ul{
5   font-weight: 700;
```

```
6 }
```

make the header font really thin, and the `` list really thick
~ nice ~

Coding exercise: Menu.html

learning outcomes:

- menu fixed up to the top, when you scroll it stays fixed
- menu item has an `` so it targets an `<h1 id="History">` header element down on the page.

Notice when you tap on the a href it puts "#History" at the end of the URL in the browser bar. You can do this on the internet too!

1. Navigate to a wikipedia page
2. look for one of the headers (inspect it and see what the #Header is called in its id.
3. in the URL append a # and the name of the ID, hit enter, watch it scroll to place.

@Media queries

Open a restaurant menu page. I used the one at

<http://www.russanddaughterscafe.com/> I found their menu if you scroll down a bit on their homepage. Shrink the HTML document width, notice how the menu will adjust until the 2 columns eventually turn into 1.

make an empty html page with a few lines of CSS:

```
1 body{ background-color: orange; }  
2  
3 @media screen and (max-width:600px){  
4     body{ background-color: purple; }  
5 }
```

the page opens with a background color of orange, but when you shrink it, it turns purple.

an important point to understand is that the code is read top to bottom. when the background color is set to purple, it still runs the code above, it sets the

background to orange first but then immediately overwrites it and sets it to purple (if the media query is true).

This prepares us for our homework: make a cafe / restaurant menu.

- header section with the title of the restaurant, maybe include the address, picture, anything
- 2 columns (must be 2!) with lists of food items, maybe drinks, food, anything else you want. Maybe it has prices, maybe ingredients, food allergies, calories, etc.. whatever.
- the 2 columns must convert to full width columns when the screen width size drops below 600, just like how it works on Russ and Daughters, and most sites on the internet.