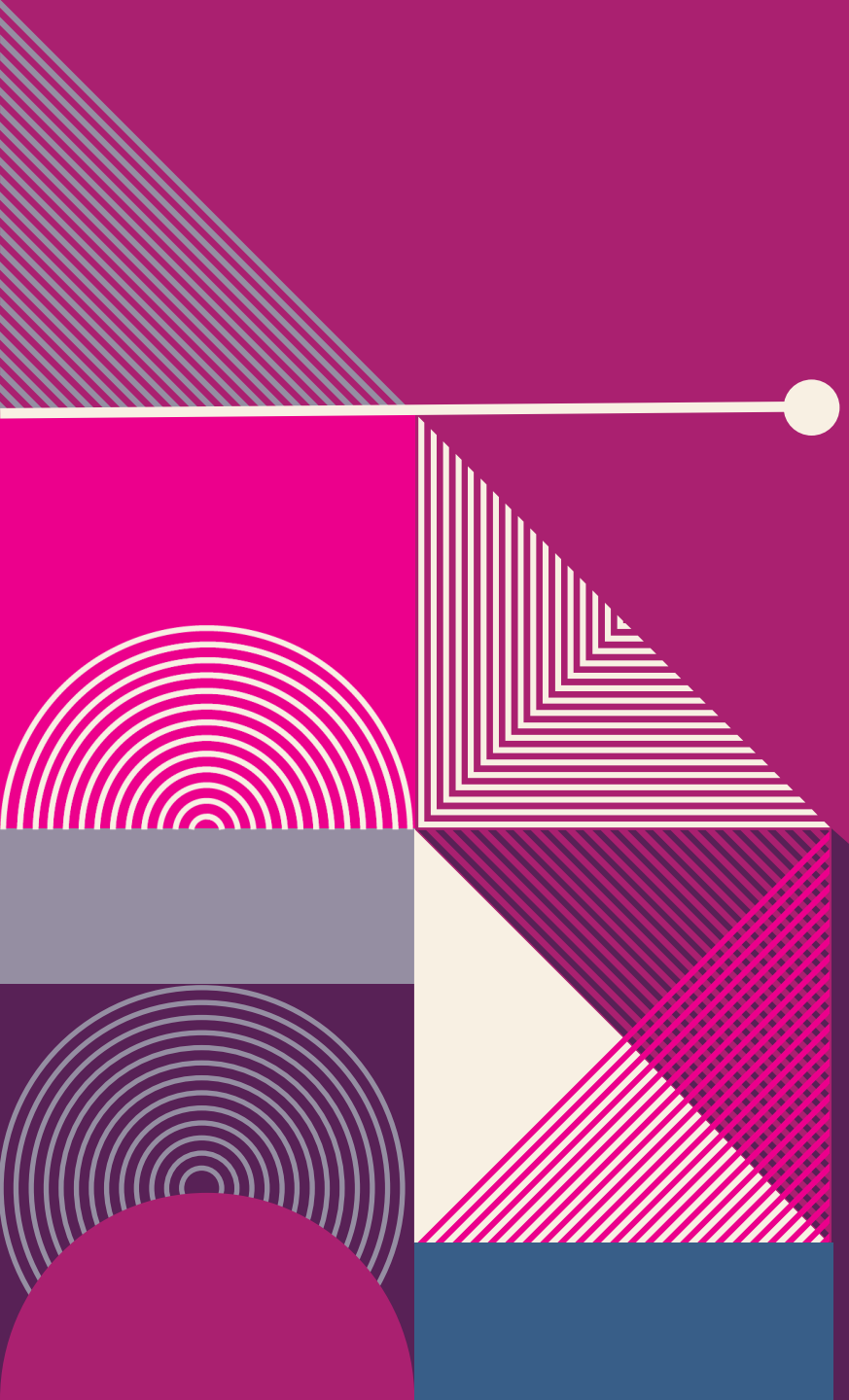


TUTORIAL: CREATING A WEBPAGE FROM SCRATCH

Part 2.1 - CSS

WHY USE CSS?

You've used HTML to build your webpage, but it doesn't have a lot of style yet. Let's add some with CSS.

CSS stands for cascading style sheet, a somewhat confusing name for a language that you use to format your page so that it looks just the way you want.

CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

CSS can get super-complicated, but the basic principles aren't hard to master.

CSS SOLVED A BIG PROBLEM

HTML was NEVER intended to contain tags for formatting a web page!
HTML was created to describe the content of a web page, like:

```
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```

When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS.

CSS removed the style formatting from the HTML page!

SELECTORS, PROPERTIES, AND VALUES

For each selector there are “properties” inside curly brackets, which simply take the form of words such as color, font-weight or background-color.

A value is given to the property following a colon (NOT an “equals” sign). Semi-colons are used to separate the properties

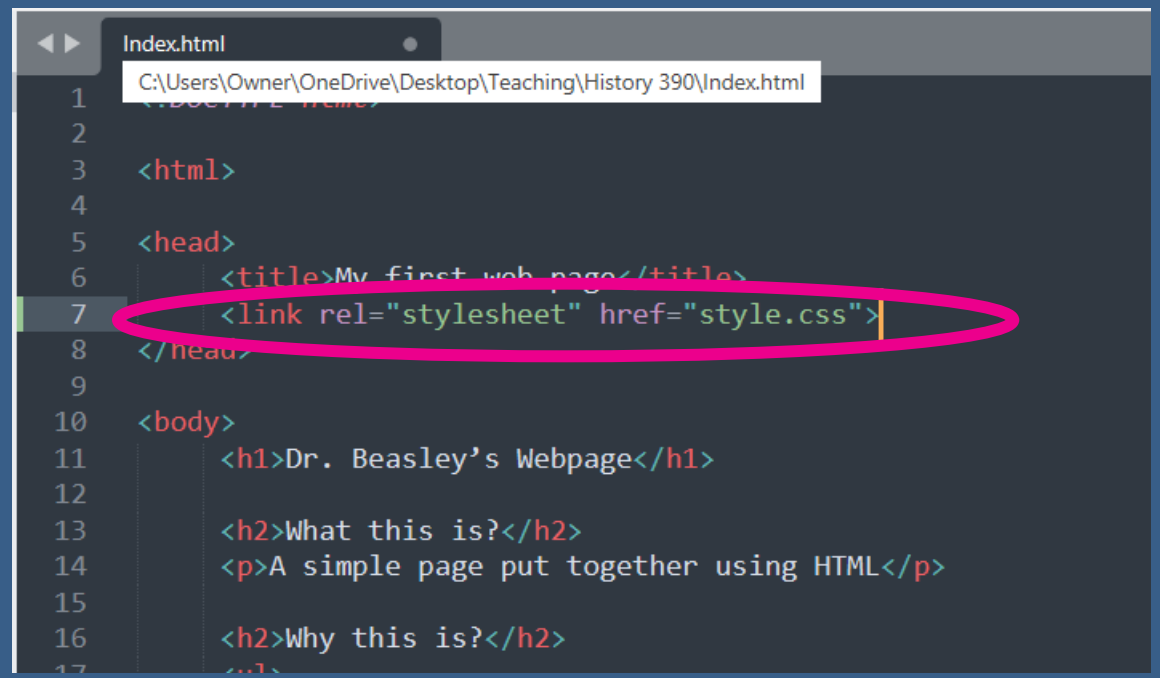
```
body {
   font-size: 14px;
   color: navy;
}
```

This will apply the given values to the font-size and color properties to the body selector.

So basically, when this is applied to an HTML document, text between the body tags (which is the content of the whole window) will be 14 pixels in size and navy in color.

Whereas HTML has tags, CSS has selectors. Selectors are the names given to styles in internal and external style sheets. In this CSS Tutorial we will be concentrating on HTML selectors, which are simply the names of HTML tags and are used to change the style of a specific type of element.

LINK YOUR HTML DOCUMENT TO A CSS DOCUMENT



You can add CSS styles to your HTML document in a few different ways, but I like to keep all of my CSS rules in a separate document.

Open a new text document in Sublime and save it in the same place as your html document, with the name style.css.

(You can call your stylesheet whatever you want, but style is customary.)

Now we have to tell the HTML document to look for the CSS document in order to receive information about styles. Luckily, that's not too hard. Inside the<head> tags on your html document, type

link rel="stylesheet" href="style.css"

Now you should be linked!

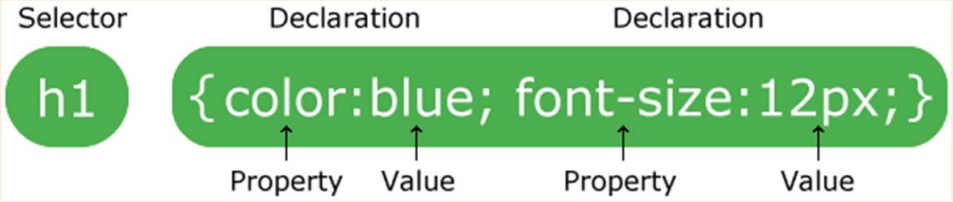
CSS SYNTAX

The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.



Example

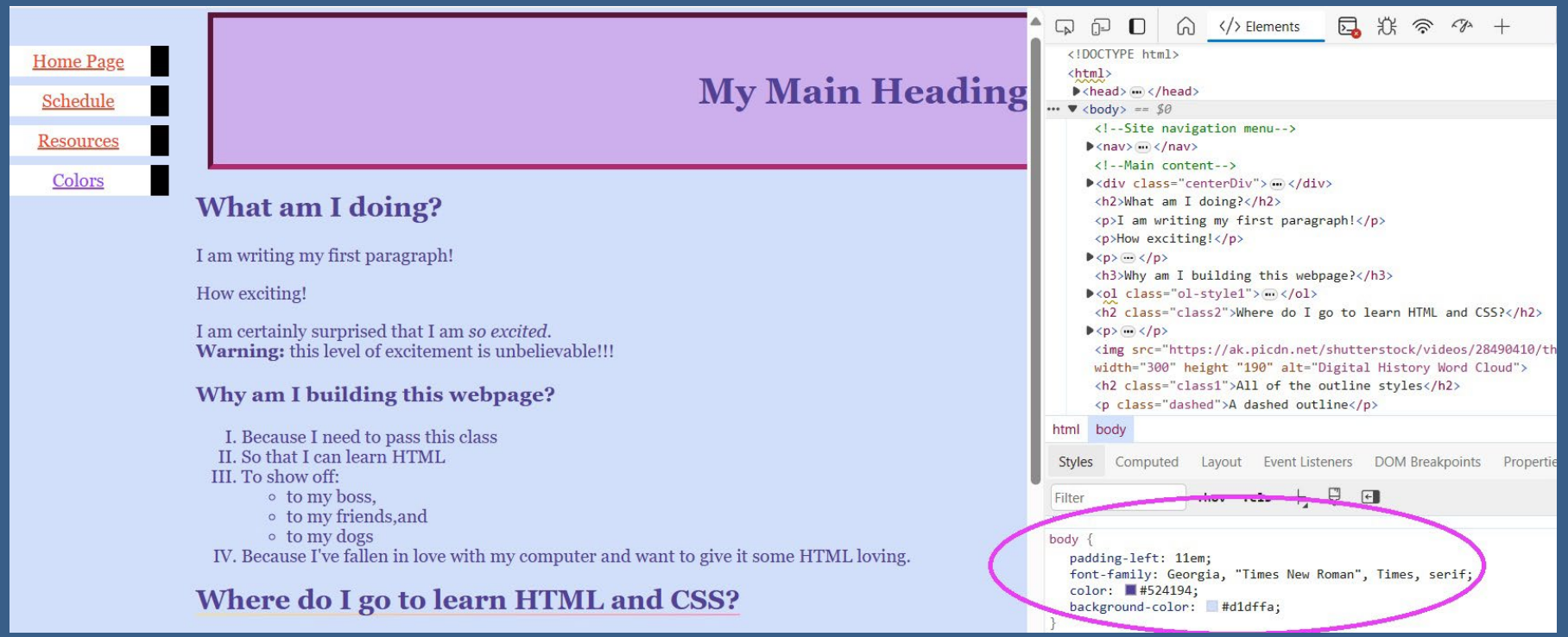
In this example all <p> elements will be center-aligned, with a red text color:

```
p {
  color: red;
  text-align: center;
}
```

Example Explained

- p is a selector in CSS (it points to the HTML element you want to style: <p>).
- color is a property, and red is the property value
- text-align is a property, and center is the property value

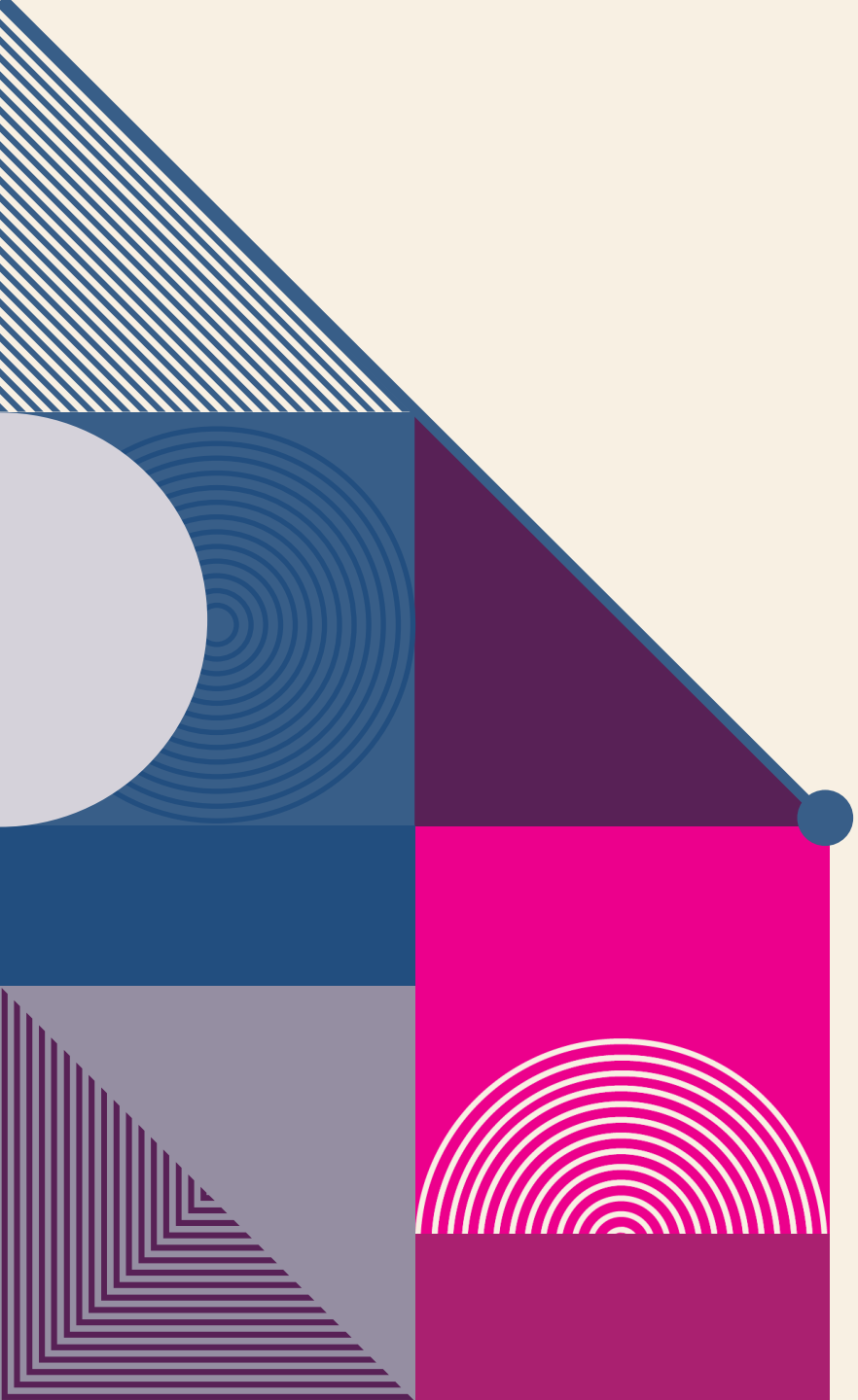
HOW CSS STYLE WORKS



1. The basic rule is that you specify the html tag you'd like your rule to affect and then say what you want to do to the content inside the tag. Then all of the content inside of that tag will be affected.

2. In the example above, I've specified that all of the content inside <body> tags should be made blue and transformed into the Georgia font. Notice that the navigation tags aren't affected. That's because the navigation menu is not inside the <body> tags.

3. As you can see, your rules go inside angle brackets, which look like this { } and are separated by semicolons.



ADDING SOME COLORS

You probably see some black text on a white background, but it depends on how the browser is configured. So one easy thing we can do to make the page more stylish is to add some colors. (Leave the browser open, we will use it again later.)

ADDING SOME COLORS, EXPLAINED

```
body {
    color: purple;
    background-color: #d8da3d }
```

- Style sheets in CSS are made up of rules. Each rule has three parts:
  - The first part says that this is a style sheet and that it is written in CSS – {
  - The second part says that we add style to the "body" element
  - The third part sets the color of the text to purple and the next line sets the background to a sort of greenish yellow

- The selector (in the example: "body"), which tells the browser which part of the document is affected by the rule
- The property (in the example, 'color' and 'background-color' are both properties), which specifies what aspect of the layout is being set
- And the value ('purple' and '#d8da3d'), which gives the value for the style property

MAKE EVERYTHING DIFFERENT COLORS!

This example shows that rules can be combined.

We have set two properties, so we could have made two separate rules by closing the bracket for each:

```
body { color: purple }

body { background-color: #d8da3d }
```

But since both rules affect the body, we only wrote “body” once and put the properties and values together.

The background of the body element is also be the background of the whole document. We haven't given any of the other elements (p, li, address…) any explicit background, so by default they will have none (or: will be transparent).

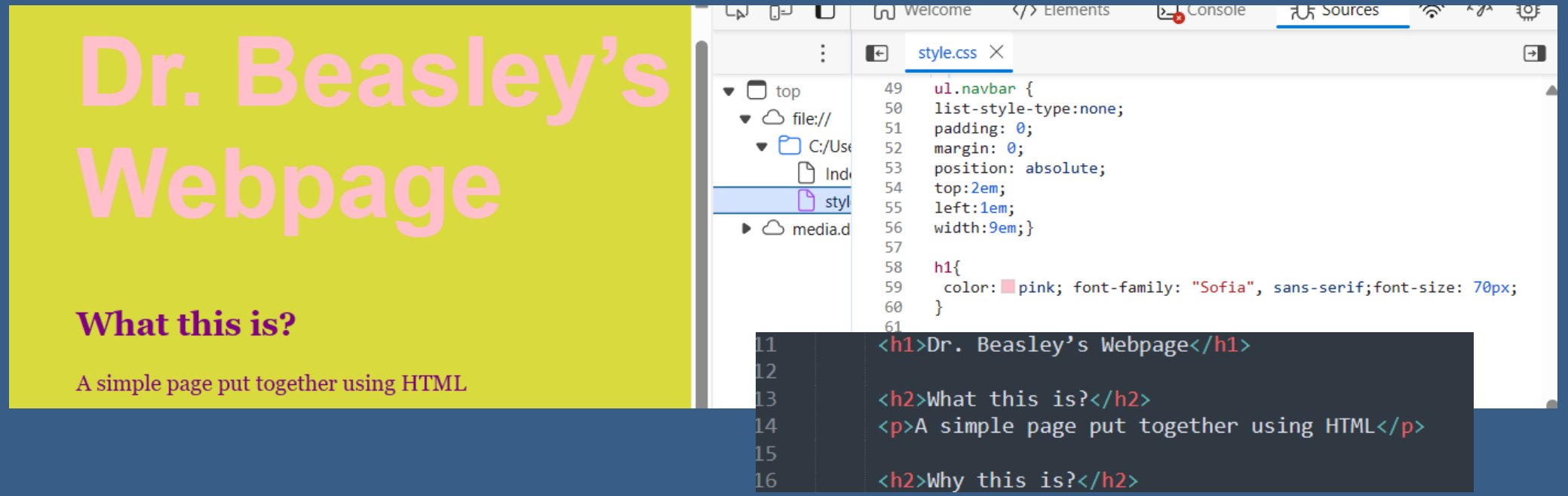
The 'color' property sets the color of the text for the body element, but all other elements inside the body inherit that color, unless explicitly overridden. (We will add some other colors later.)

Colors can be specified in CSS in several ways. This example shows two of them: by name (“purple”) and by hexadecimal code (“#d8da3d”). There are about 140 color names and the hexadecimal codes allow for over 16 million colors.

I tend to use Adobe Color to play around with colors: Color wheel, a color palette generator | Adobe Color

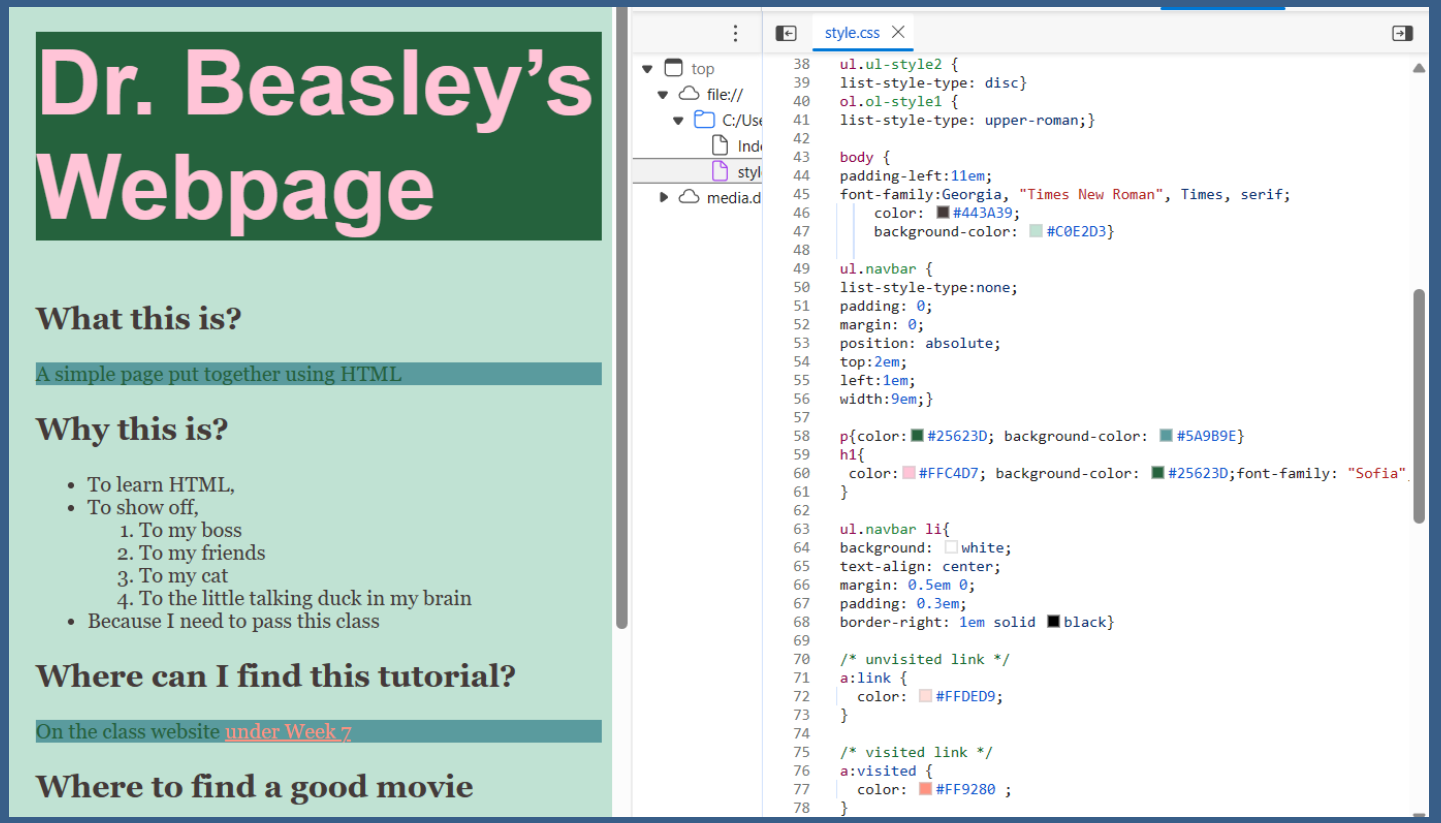
Now save this your “style.css” file (use “Save” from the File menu) and go back to the browser window. If you press the “Reload” button, the display should change from the “boring” page to a colored (but still rather boring) page. Apart from the list of links at the top, the text should now be purple against a greenish yellow background.

MAKE EVERYTHING DIFFERENT COLORS!



In the example above, I've specified that all of the content inside <h1> tags should be made pink and transformed into the Sofia font. Notice that the content outside of the <h1> tags isn't affected. That's because the other tags are not inside the <h1> tags.

CHANGE BACKGROUND COLORS



To CSS, every element on your webpage forms a box. You can change the background color of this box by using the background-color style rule.

```
body {
padding-left:11em;
font-family: Georgia, "Times New Roman", Times, serif;
   color: #443A39;
   background-color: #C0E2D3}
```

```
p {color:#25623D; background-color: #5A9B9E}
h1{
 color:#FFC4D7; background-color: #25623D;
}
```

As you can see above, even the body tag forms a box that contains everything on the page. Who wants a boring background color? Make that page stylish!

ADDING FONTS

Another thing that is easy to do is to make some distinction in the fonts for the various elements of the page. So let's set the text in the “Georgia” font, except for the h1 heading, which we'll give “Helvetica.”

On the Web, you can never be sure what fonts your readers have on their computers, so we add some alternatives as well: if Georgia is not available, Times New Roman or Times are also fine, and if all else fails, the browser may use any other font with serifs. If Helvetica is absent, Geneva, Arial and SunSans-Regular are quite similar in shape, and if none of these work, the browser can choose any other font that is serif-less.

Another way to explore fonts: CSS Font Stack: Web Safe and Web Font Family with HTML and CSS code.

Google Fonts allows some options, but merging to CSS takes a little more work: Browse Fonts - Google Fonts

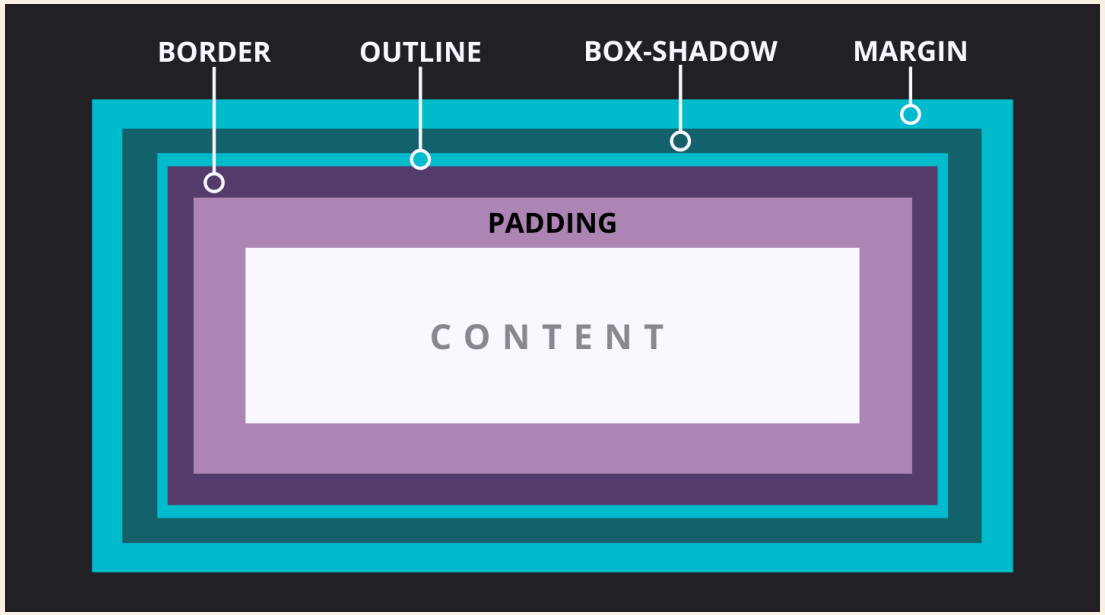
ADDING FONTS, CONT.

In the text editor add the following lines in your CSS:

```
body {
   font-family: Georgia, "Times New Roman",
        Times, serif;
   color: purple;
   background-color: #d8da3d }
 h1 {
   font-family: Helvetica, Geneva, Arial,
        SunSans-Regular, sans-serif;
color: #2C3252 }
```

If you save the file again and press “Reload” in the browser, there should now be different fonts for the heading and the other text.

ADDING ELEMENT BORDERS



The border is precisely the boundary of the element, sitting between its padding and margin, and its width will impact the computed element dimensions

The outline is next to but outside of the border, overlapping both box-shadow and margin, but not affecting the element's dimensions

By default, box-shadow extends out from edge of the border covering the amount of space in the direction(s) defined, and it will also not affect the element's dimensions

ADDING ELEMENT BORDERS

Since everything on your page is a box, you can add borders around everything.

First, we need to add some more html to play with:

…

```
<body>
```

….

```
<p class="one">The border color of this paragraph is a gradient.</p>

    <p class="two">The outline color of this paragraph is same as the element's text color.</p>
```

Add this to your CSS

```
 p.one {
        color: linear-gradient (to right, purple, pink) 1;
        border: 2px solid; border-image: linear-gradient (to right, purple, pink) 1;
    }
    p.two {
        color: #3F1D5C;
        outline: 6px solid;
    }
```

The border color of this paragraph is a gradient.

The outline color of this paragraph is same as the element's text color.

```
46      color: #443A39;
47      background-color: #C0E2D3}
48
49
50  p{color: #25623D; background-color: #5A9B9E}
51  h1{
52   color: #FFC4D7; background-color: #25623D;
53   font-family: "Sofia", sans-serif;font-size: 70px;
54  }
55   p.one {
56        color: linear-gradient(to right, purple, pink) 1;
57        border: 2px solid; border-image: linear-gradient(to right,
58      }
59      p.two {
60        color: #3F1D5C;
61        outline: 6px solid;
62
```

{ } Line 43, Column 7 Coverage: n/a

LET ME KNOW WHEN YOU HAVE FINISHED AND I WILL SHARE CSS 2.2