

CSS!

CSS stands for Cascading Style Sheets! Using CSS allows for us to style our sites beyond basic HTML elements! We've already run into styling from the last lesson! Now we're going to take what we've learned from that lesson and expand on it!

We can write our css stylings either inline (which we've already done in the past lesson), in the `<style>` section of our page's `<head>`, or in a completely separate file (we'll get to that at the end, for now we'll start moving our CSS to the style section in the head).

Syntax:

CSS Syntax looks completely different from HTML! This is because it's a different type of language!

When we write CSS in the style section or a CSS file, we don't wrap it in quotations, like we did with inline styling:

```
1 | selector { property: value, property: value }
```

In CSS, the property:value pair is a declaration.

```
1 | p { color: 'blue', font-size: 12px }
```

CSS Doesn't care how many spaces/new lines your code has, but just make sure you have both opening and closing curly braces!

Using CSS:

Like mentioned above, we've already seen inline styling. This works completely fine and well, but that takes too long to change every single inline style you use. Inline styling is great for one or two changes, but when you have to change every line, that becomes painful. What happens if our website has multiple pages, each with hundreds of lines? Let's quickly convert some inline styled code to use CSS!

```

1 <!doctype html>
2 <html>
3   <head>
4   </head>
5   <body style='background-color:lightblue'>
6     <h1 style='border-left: 1px solid blue; padding: 10px'> Images are some
of the best parts of the internet! </h1>
7     <hr />
8     <div>
9       Check out this beautiful image of my old pet, Rosie!
10      <img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/christmasTarantula.JPG' width='500px' />
11    </div>
12  </body>
13 </html>

```

We have styling for our `body`, `h1`, and our `img`. Instead of our stylings and put them in a `<style>` section our `<head>`! [See how the code renders like the above code, but with styling in the header?](#)

```

1 <!doctype html>
2 <html>
3   <head>
4     <style>
5       body { background-color: lightblue }
6       h1 { border-left: 1px solid blue;
7         padding: 10px
8       }
9       img { width: 500px }
10    </style>
11  </head>
12  <body>
13    <h1> Images are some of the best parts of the internet! </h1>
14    <hr />
15    <div>
16      Check out this beautiful image of my old pet, Rosie!
17      <img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/christmasTarantula.JPG' />
18    </div>
19  </body>
20 </html>

```

Notice how we put our styling in the `<style>` section of our site, following the below format:

```
1 | selector { property: value, property: value}
```

By using the above format, we've told our site that every time we use a `h1`, it will always have a border-left, and a padding of 10 pixels, and all of our `` elements will have a width of 500 pixels! As far as our body is concerned, we've only got one, but it's nice to have our styling all in one place instead of spread out throughout the HTML's code!

Classes and Identities:

What if we wanted our divs to all have different styles? We can't do that with the CSS above! By using Classes and Identities, we can use as many different styles as we want!

The differences between classes and IDs is that IDs are unique, and classes aren't! Multiple elements can have the same class, whereas only single elements can have an id.

What does this mean?

IDs

Creating an ID in a stylesheet, you need to start your selector with the `#`. [See it in action.](#)

```
1 | <!doctype html>
2 | <html>
3 |   <head>
4 |     <style>
5 |       #IamAnID { color: red }
6 |     </style>
7 |   </head>
8 |   <body>
9 |     <div id='IamAnID'>WHAT COLOR AM I!</div>
10 |   </body>
11 | </html>
```

The ID that we made in our `<style></style>` element is used by the `<div></div>` element in our body. Remember that the ID absolutely has to match (so it's case sensitive).

By using the ID, we then get to turn our div's color to red!

A downside to using ID is that only one element should use it in your whole page! [We shouldn't do this:](#)

```

1 <!doctype html>
2 <html>
3   <head>
4     <style>
5       #IamAnID { color: red }
6     </style>
7   </head>
8   <body>
9     <div id='IamAnID'>WHAT COLOR AM I!?!</div>
10    <div id='IamAnID'>I'm the same color!</div>
11  </body>
12 </html>

```

This is because IDs are just that, a specific ID! It's like if somebody was trying to use your driver's license (or school ID, or anything). They're not you! (Note: because our browsers are really good at trying to fix our mistakes, it will render the colors properly, but having more than 1 ID can lead to more problems later on!)

Classes

To share CSS among multiple elements, we should use Classes!

Instead of a #, we use a . [This way we get to share everything!](#)

```

1 <!doctype html>
2 <html>
3   <head>
4     <style>
5       .IamAClass { color: red }
6     </style>
7   </head>
8   <body>
9     <div class='IamAClass'>WHAT COLOR AM I!?!</div>
10    <div class='IamAClass'>I'm the same color!</div>
11  </body>
12 </html>

```

So, now we can use the same class over and over and over again! When we try to use the class, we add the attribute 'class' in our element, and assign it to the name of our class in the css 'IamAClass'.

[You can mix and match as many classes and IDs as you want \(remember, some classes might override others\):](#)

```

1  <!doctype html>
2  <html>
3    <head>
4      <style>
5        #IamAnID { color: red }
6        .bold { font-weight: bold }
7        .italic { text-style: italic }
8        .underline { text-decoration: underline }
9      </style>
10   </head>
11   <body>
12     <div id='IamAnID' class='bold italic underline' >My color is red, I am
    bold, italic, and underlined!</div>
13   </body>
14 </html>

```

Commenting Your CSS!

When we wrote comments before in HTML, we used `<!-- -->`. Because CSS is a different language, we comment differently. We use `/* */`. Just like html, you put your comments between the text: `/* I am a comment! */`. [See \(or really don't see\) the comments in action!](#)

```

1  <!doctype html>
2  <html>
3    <head>
4      <style>
5        /* I AM COMMENTED OUT */
6        .IamAClass { color: red }
7      </style>
8    </head>
9    <body>
10     <div class='IamAClass'>WHAT COLOR AM I!?!</div>
11     <div class='IamAClass'>I'm the same color!</div>
12   </body>
13 </html>

```

Stylings:

Now let's explore the differing kinds of stylings we can use!

Background colors:

We've already seen background-color, but it's good to revisit it! To change background colors is background-color:

```
b { background-color: green }
```

[See it again in action:](#)

```
1  <!doctype html>
2  <html>
3      <head>
4          <style>
5              body { background-color: cyan }
6          </style>
7      </head>
8      <body>
9          <div> Oh wow, look at this cyan body! </div>
10     </body>
11 </html>
```

Background images!

You can add a background image if you want (though not always the greatest idea)! [Can you spot the cat in the background??](#)

```
1  <html>
2      <head>
3          <style>
4              div { background-
5                  image:url('https://ichef.bbci.co.uk/images/ic/720x405/p0517py6.jpg');
6                  height:400px; width:700px; }
7          </style>
8      </head>
9      <body>
10         <div> I HAVE A CAT BEHIND ME!! </div>
11     </body>
12 </html>
```

Try changing the image height and width to be larger. Notice that the image repeats? You can change that using the attribute: 'background-repeat:no-repeat' in your css!

With background-repeat, you can use the values:

- [no-repeat](#)
- repeat // this is the default value
- [repeat-x](#) // repeats along the x-axis (horizontally)

- [repeat-y // repeats along the y-axis \(vertically\)](#)

You can use both direct and relative links too! If you have an image file named myImage.jpg in the same folder as your index.html file, you could just write `url('./myImage.jpg')`

Background-Attachment:

This allows your image to [scroll with the page](#)!

```
1 <!doctype html>
2 <html>
3   <head>
4     <style>
5       .scrollImage { background-
image:url('https://ichef.bbci.co.uk/images/ic/720x405/p0517py6.jpg');
height:1200px; width:1200px;
6         background-repeat:no-repeat;
7         background-attachment: scroll}
8     </style>
9   </head>
10  <body>
11    <div class='scrollImage'> TRY SCROLLING! </div>
12  </body>
13 </html>
14
```

You can also have your image fixed, so that the text scrolls, [while your image does not](#)!

```
1 <!doctype html>
2 <html>
3   <head>
4     <style>
5       .dontScroll { background-
image:url('https://ichef.bbci.co.uk/images/ic/720x405/p0517py6.jpg');
height:1200px; width:1200px;
6         background-repeat:no-repeat;
7         background-attachment: fixed}
8     </style>
9   </head>
10  <body>
11    <div class='dontScroll'> TRY SCROLLING!! </div>
12  </body>
13 </html>
```

Background position:

You don't always want to have your background chilling in the top left of your page. That's boring. Put yours wherever you [want with background-position](#) :

```
1  <!doctype html>
2  <html>
3    <head>
4      <style>
5        .dontScroll { background-
image:url('https://ichef.bbci.co.uk/images/ic/720x405/p0517py6.jpg');
height:1200px; width:1200px;
6          background-repeat:no-repeat;
7          background-position: center }
8      </style>
9    </head>
10   <body>
11     <div class='dontScroll'> There's a cat in the center!   </div>
12   </body>
13 </html>
```

You can position it to have the values:

- [top](#)
- [center](#)
- [bottom](#)
- [left](#)
- [right](#)

We'll come back to this, but you can also set your position by using pixels and percentages too!

Text Formatting!

We did some text formatting earlier, but let's take a deeper dive!

Text Color

To change your text's color in an element, you can use the color property. You can name your colors with either the [color's name OR use the hex code value](#).


```

1  <!doctype html>
2  <html>
3      <head>
4          <style>
5              .blue { color: blue }
6              .blueHex { color: #0000FF }
7          </style>
8      </head>
9      <body>
10         <div class='blue'> my color's blue!! </div>
11         <div class='blueHex'> my color's also blue, but from using hex code!!
12     </div>
13     </body>
14 </html>

```

Italics

Sometimes you want to italicise things. When you do that, you can use `font-style` property. This takes in either italic, oblique, or none ([italic and oblique are more or less the same thing](#)):

```

1  <!doctype html>
2  <html>
3      <head>
4          <style>
5              .italicsClass { font-style: italic }
6              .obliqueClass { font-style: oblique }
7          </style>
8      </head>
9      <body>
10         <div class='italicsClass'> I'm italic!! </div>
11         <div class='obliqueClass'> I'm oblique!! </div>
12     </body>
13 </html>
14

```

Bolds

When you want to make something bold, you want to use the font-weight property! Font-weight takes the values of: [bold and normal](#)

```

1 <!doctype html>
2 <html>
3 <head>
4 <style>
5     .boldMove { font-weight: bold }
6 </style>
7 </head>
8 <body>
9     <div class='boldMove'> I'm bold!! </div>
10 </body>
11 </html>
12

```

Underlines Overlines and Strikethroughs

When you want to underline, overline, or strikethrough things - we need to use another [property called text-decoration](#):

```

1 <!doctype html>
2 <html>
3 <head>
4 <style>
5     .underline { text-decoration: underline }
6     .overline { text-decoration: overline }
7     .strikethrough { text-decoration: line-through }
8
9 </style>
10 </head>
11 <body>
12     <div class='underline'> I'm very important, so I'm underlined!! </div>
13     <div class='strikethrough'> I'm wrong, so I've got a line through me!!
14 </div>
15     <div class='overline'> I've got a line over my head!! </div>
16 </body>
17 </html>

```

Font Properties!

Font-Size:

Font-Size - you guessed it - it changes the size of our font. Font-size takes in px values (pixel values, like 12px), but also has relatively normal values like xx-small, x-small, smaller, small, medium, large, larger, x-large, and xx-large. Using the px allows for a greater level of customization, but if you just want some big font and don't want to play with what specific pixel amount you want, [go for the worded values!](#)

```
1  <!doctype html>
2  <html>
3      <head>
4          <style>
5              .small { font-size: small }
6              .medium { font-size: medium }
7              .large { font-size: large }
8              .px12 { font-size: 12px }
9              .px20 { font-size: 20px}
10             .px30 { font-size: 30px}
11         </style>
12     </head>
13     <body>
14         <div class='small'> I'm small </div>
15         <div class='medium'> I'm medium... </div>
16         <div class='large'> I'm large </div>
17
18         <br />
19
20         <div class='px12'> I'm 12 pixels! </div>
21         <div class='px20'> I'm 20 pixels! </div>
22         <div class='px30'> I'm 30 pixels! </div>
23     </body>
24 </html>
```

Font Family:

This is how you change the actual font (like Times New Roman, Helvetica, ect). Most browsers will be able to render most fonts, but sometimes a browser might not have the proper font installed. For you to ensure that your site looks good, you can include a whole list of fonts to make sure the browser will properly render your site.

Because font-family takes in a list of fonts, the values need to be comma separated. If your font has a space in the name, then we need to have the names wrapped in quotations:

```
div { font-family: "Fira Code", "Comic Sans", Helvetica, sans-serif }
```

[Check out the code:](#)

```

1 <!doctype html>
2 <html>
3   <head>
4     <style>
5       div { font-family: "Fira Code", "Comic Sans", Helvetica, sans-serif }
6       p { font-family: monospace, "Times New Roman" }
7     </style>
8   </head>
9   <body>
10    <div>Oh hey! What do you think my font will be? </div>
11    <p> What about me?</p>
12  </body>
13 </html>

```

Notice that the div didn't find Fira code or comic sans and went to Helvetica? That's because Fira Code is a very special font that most browsers don't support, and comic sans is actually called "Comic Sans MS", give that a go and see how that works!

Text Properties:

Text align:

To align your text is where the text is aligned on your page (like in MS Word). The values we need to use are [left](#), [center](#), [right](#), and [justify](#).

```

1 <!doctype html>
2 <html>
3   <head>
4     <style>
5       .center { text-align:center }
6       .left { text-align:left }
7       .right { text-align:right }
8       .justify { text-align:justify }
9     </style>
10  </head>
11  <body>
12    <div class='center'>Let me tell you all a little about myself. My name is
Matt Lane, and I grew up down in Imperial, Missouri. Back when I lived there
the town was right off the highway, but only 1 block long.</div>
13    <br/>
14    <div class='left'>We had a local butcher named Harrel who used to make
the best cuts of meat. Harrel would literally just have a whole cow hanging
in the back room that you could see if you snuck around the counter.</div>
15    <br/>

```

```

16     <div class='right'>A few doors down from Harrel's butcher shop was the
    local barber. Now, when I was kid, this was back in the day when bleaching
    only the very tips of your hair was cool. I however, got my hair bleached
    entirely white. That's because at that time, I really liked dragon ball z. I
    would do my hair so that it looked like a super saiyan Goku (I thought I was
    really cool). </div>
17     <br/>
18     <div class='justified'>At my parents house, we lived across from a really
    big farm. There weren't many lights out, so you could see stars fill the sky.
    The downside about living next to a farm, though, was that when the cows
    would get out, they'd come eat my mom's flower garden (that is, if the deer
    didn't already get to the flower first). There were always lots of animals
    around. Most of them were cool, but we sure had a lot of spiders. So many
    spiders, in fact, that the priest's house at the church we used to go to had
    a spider infestation so bad, that the entire attic was filled with spiders
    from wall to wall. They had to fumigate the entire building. At that time, I
    hated spiders, but now, I actually love them! Many years later, when I worked
    for the botanical garden, we had dozens of different kinds of spiders to play
    with!
19     </div>
20 </body>
21 </html>

```

Text indent:

This will indent your text a new section of words according to its value! The values for the property are: [length and percentage](#):

```

1  <!doctype html>
2  <html>
3    <head>
4      <style>
5        .indent5percent { text-indent: 5% }
6        .indent10percent { text-indent: 10% }
7        .indent15px { text-indent: 15px }
8        .indent3em { text-indent: 3em }
9      </style>
10   </head>
11   <body>
12     <div> If you look at the indent styles, you'll notice that not only are
    we using percentages but we're also using lengths! </div>
13     <div class='indent5percent'> When working with inputting certain lengths
    before, we've only used pixel length!</div>

```

```

14     <div class='indent10percent'>As it turns out, we can use a ton of
different types lengths (though pixels do work really well on their own)!
</div>
15     <div class='indent15px'> When using HTML, it's good to know that there
are always a ton of options to do whatever it is that you want. Some of them
may be better than others</div>
16     <div class='indent3em'>While some may be better than others, just use
what you're most comfortable with to begin! You'll come across other styles
of coding in time and learn from there!</div>
17
18     <br />
19     <div style="text-align:left">
20         Here are some examples of units to use! (There are many more! You can
find a list <a href='https://www.w3schools.com/cssref/css_units.asp'>here</a>
)
21         <table>
22             <tr>
23                 <th>Unit</th>
24                 <th>Description</th>
25             </tr>
26             <tr>
27                 <td>cm</td>
28                 <td>centimeters</td>
29             </tr>
30             <tr>
31                 <td>mm</td>
32                 <td>millimeters</td>
33             </tr>
34             <tr>
35                 <td>in</td>
36                 <td>inches</td>
37             </tr>
38             <tr>
39                 <td>px</td>
40                 <td>pixels</td>
41             </tr>
42             <tr>
43                 <td>em</td>
44                 <td>Font size relation! 2 em would mean 2 times the size of the
font</td>
45             </tr>
46             <tr>
47                 <td>%</td>
48                 <td>Font size relation as a percentage of the font in your parent
element (a parent element is the one that your current element lives in. The
parent element of this table data element is a table row element)</td>

```

```

49         </tr>
50     </table>
51
52     </div>
53 </body>
54 </html>
55

```

Text transform

Suppose you're feeling lazy, and don't want to type out a certain format in a sentence. Text transform takes the values [capitalize](#), [uppercase](#), [lowercase](#), and [none](#).

```

1  <!doctype html>
2
3  <html>
4    <head>
5      <style>
6        .capitalize { text-transform: capitalize }
7        .lowercase { text-transform: lowercase }
8        .uppercase { text-transform: uppercase }
9
10     </style>
11   </head>
12   <body>
13     <div class='capitalize'> the first letter of each word in my sentence
will be capitalized! </div>
14     <div class='lowercase'>  EVEN THOUGH I TYPED THIS SENTENCE WITH CAPS
LOCK ON, IT SHOWS UP LOWERCASED! </div>
15     <div class='uppercase'> this whole sentence was written in lowercased
letters. the uppercase text-transform did all the work for me! </div>
16   </body>
17 </html>

```

Styling Lists

You've seen this before, but let's style our lists not inline! Lists are boring! So, we should style them in some way to spice them up a little bit! We can do this with list-style-type. This property can take a value of: [disc](#), [circle](#), [square](#), [decimal](#), [upper-roman](#), [lower-roman](#), and [upper-alpha](#).

Bullets, numbers, etc:

```

1 <!doctype html>
2 <html>
3   <head>
4     <style>
5       .disc { list-style-type:disc }
6       .circle { list-style-type:circle }
7       .square { list-style-type:square }
8       .decimal { list-style-type:decimal }
9       .upperRoman { list-style-type:upper-roman }
10      .lowerRoman { list-style-type:lower-roman }
11      .upperAlpha { list-style-type:upper-alpha }
12    </style>
13  </head>
14  <body>
15
16    <h2>Unordered Lists!</h2> <hr />
17    <h4>Discs: </h4>
18    <ul class='disc'>
19      <li> Disc item </li>
20      <li> Another disc item</li>
21    </ul>
22
23    <h4>Circles: </h4>
24    <ul class='circle'>
25      <li> A circle item! </li>
26      <li> Another circle!</li>
27    </ul>
28
29    <h4>Squares: </h4>
30    <ul class='square'>
31      <li> Check out this square </li>
32      <li> We're all squares here </li>
33    </ul>
34
35    <h2>Ordered Lists</h2> <hr />
36    <h4>Decimals: </h4>
37    <ol class='decimal'>
38      <li> Numbers are fun! </li>
39      <li> Especially when they count!
40        <ol>
41          <li>This is great for when you have a lot of sub lists!
42            <li>As a sublist, I agree!</li>
43          </ol>
44        </li>
45      </li>

```



```

46         </ol>
47
48     <h4>Uppercase Roman Numerals: </h4>
49     <ol class='upperRoman'>
50         <li> Roman numerals look very official</li>
51         <li> You could use this to number your latin quotes!
52             <ol>
53                 <li>CARTHAGO DELENDA EST! ("Carthage Must Be Destroyed")
- Cato the Elder</li>
54                 <li>VENI VIDI VICI ("I came I saw I conquered") - Julius
Caesar </li>
55             </ol>
56         </li>
57     </ol>
58
59     <h4>Lowercase Roman Numerals: </h4>
60     <ol class='lowerRoman'>
61         <li> Lowercase roman numerals are also great!</li>
62         <li> Who doesn't love counting with tiny i's? </li>
63     </ol>
64
65     <h4>Uppercase Letters: </h4>
66     <ol class='upperAlpha'>
67         <li> We're pretty used to alphabetical ordering </li>
68         <li> A lot of text editors (like MS Word) default to this</li>
69     </ol>
70
71     <h2> YOU CAN MIX AND MATCH! </h2><hr />
72     <ol class='upperRoman'>
73         <li> Uppercase Roman numerals are great</li>
74         <li> But why have them all the way down? Why not have subelements
with DIFFERENT counting systems?
75             <ol class='lowerRoman'>
76                 <li>LABOR OMNIA VINCIT ("Hard work conquers all!") -
Virgil </li>
77                 <li>PULVUS ET UMBRA SUMUS ("We are but dust and shadow")-
Horace </li>
78             </ol>
79         </li>
80         <li> By mixing and matching, it makes it easier to keep track of
where you are! </li>
81         <li class='decimal'> If you wanted to be crazy though, you could
also mix classes mid-list - which I personally think is a bad idea, but don't
let me stop you!
82             <ul class='square'>

```

```

83         <li>You can also add unordered lists to ordered lists as
sublists, and vice versa</li>
84         <li>So many cool list possibilities!</li>
85     </ul>
86 </li>
87 </ol>
88 </body>
89 </html>
90

```

Custom List Images:

Suppose you wanted to make a really bonkers list, and you wanted your own special list item image. What we're really doing is basically setting a background image for all of our list items and then we remove the actual list-style-type, [so that there aren't any bullets](#):

```

1  <!doctype html>
2  <html>
3      <head>
4          <style>
5              li {
6                  background-
image:url('https://media.giphy.com/media/V0YMxvqOXOkEw/giphy.gif');
7                  background-repeat: no-repeat;
8                  list-style-type: none; /* this makes it so we don't show any
bullets or numbers */
9                  padding-left:75px;
10                 padding-bottom:50px;
11             }
12         </style>
13     </head>
14     <body>
15         <h2>Unordered Lists!</h2> <hr />
16         <h4>CATS: </h4>
17         <ul class='disc'>
18             <li> Check out our cat list </li>
19             <li> Cat lists are pretty dang cool</li>
20             <li> Cat!</li>
21         </ul>
22     </body>
23 </html>

```

Adding Width and Height to Elements:

You can add width and height to any element (we did this with images earlier!). We can set special elements to take up smaller amounts of space, or significantly larger!

Width:

Takes a value of either some length or a percentage. If you don't specifically say what you want the width to be, the browser will automatically adjust it to [whatever it thinks the right size is](#).

```
1  <!doctype html>
2
3  <html>
4    <head>
5      <style>
6        .auto { width:auto } /* automatic width - we don't need to put this in
since it's default, but it's good to see*/
7        .half { width: 50% }
8        .pixel { width: 500px }
9        .border { border: 2px solid black }
10     </style>
11   </head>
12   <body>
13     <div class='auto border'> I am an automatically scaled div </div>
14     <div class='half border'> I'm half of the screen! </div>
15     <div class='pixel border'> I'm 500 pixels wide!</div>
16   </body>
17 </html>
18
```

Height:

This works the exact same as width, but It's mostly recommended to use pixels or some other length measurement. The downside is that height doesn't always work with auto and percentages (you'll see below). Try taking this code and opening it on a browser on your own machine ([that is, save it to a file called height.html and double click on it](#)):

```
1  <!doctype html>
2  <html>
3    <head>
4      <style>
5        .auto { height:auto }
6        .half { height: 50% }
7        .pixel { height: 500px }
```

```

8      .border { border: 2px solid black }
9      </style>
10     </head>
11     <body>
12         <div class='auto border'> I am an automatically scaled div, I should be
the whole screen, but I don't always work like I should </div>
13         <div class='half border'> I should be half of the screen! </div>
14         <div class='pixel border'> I'm 500 pixels wide!</div>
15     </body>
16 </html>

```

Borders:

We've seen a lot of borders already. Borders take values of either, thin, medium, thick, and some value of length. Up to now, we've only been using solid borders, but borders can be: none, solid, dotted, ridge, inset, double, groove, outset, and dashed. You absolutely need to choose a type of border for it to work! [For the 3d borders, we need to add border-color too.](#)

```

1  <!doctype html>
2  <html>
3  <head>
4  <style>
5      div { padding: 5px;
6      }
7      .badBorderNoType { border: 3px;}
8      .solidBorder { border: 3px solid;}
9      .dottedBorder { border: 3px dotted;}
10     .doubleBorder { border: 3px double;}
11     .dashedBorder { border: 3px dashed;}
12     .insetBorder { border: 6px inset; border-color: green}      /* This
is a 3d border - needs a border color defined*/
13     .grooveBorder { border: 6px groove; border-color: teal}    /* This is
a 3d border - needs a border color defined*/
14     .outsetBorder { border: 6px outset; border-color: blue}    /* This is
a 3d border - needs a border color defined*/
15     .ridgeBorder { border: 6px ridge; border-color: purple}    /* This
is a 3d border - needs a border color defined*/
16
17     </style>
18 </head>
19 <body>

```

```

20     <div class='badBorderNoType'> I am a bad border, since I have no type,
    so my border won't render </div> <br />
21     <div class='solidBorder'> I'm a solid border. You've seen me before.
    </div><br />
22     <div class='dottedBorder'> I'm a dotted border! </div><br />
23     <div class='doubleBorder'> Double borders! Just in case you wanted to
    circle something twice!</div><br />
24     <div class='dashedBorder'> I'm surrounded by dashes! </div><br />
25     <div class='ridgeBorder'> Check out my ridges </div><br />
26     <div class='grooveBorder'> Groovey </div><br />
27     <div class='insetBorder'> I'm INSET - does it look like inside the
    page? </div><br />
28     <div class='outsetBorder'> I'm OUTSET - does it look like I'm coming
    out of the page?</div><br />
29     </body>
30 </html>

```

You can change specific parts of the border width by adding a direction: [top](#), [right](#), [bottom](#), [left](#):

```

1  <!doctype html>
2    <html>
3      <head>
4        <style>
5          .leftBorder {
6            border-left-width: 3px;
7            border-left-color: darkslateblue;
8            border-left-style: solid;
9          }
10
11         .leftPadding {
12           padding-left: 5px
13         }
14
15       </style>
16     </head>
17     <body>
18       <div class='leftBorder'>
19         <h2 class='leftPadding'>I only have a left border!</h2>
20         <div class='leftPadding'>We don't always want a full border! Maybe
    we just want a left border to make it look like we're quoting someone!</div>
21       </div>
22       <br />
23       <div> Borders are extremely useful! Think about if you had a table!
    You'd need a border to show which cells are which! </div>

```

```
24     </body>
25 </html>
26
```

Spacing Margins and Padding!

You've probably noticed that we've been adding padding on some of our examples to be a bit more clear! Let's go over that now!

Margins are the spacing AROUND the element (like OUTSIDE of our border)

Padding is the spacing INSIDE the element (like INSIDE of our border)

Just like 'border', we can access specific sides for our [margins and padding](#):

- margin-left
- margin-right
- margin-top
- margin-bottom
- padding-left
- padding-right
- padding-top
- padding-bottom

```
1  <!doctype html>
2  <html>
3    <head>
4      <style>
5        .margin { margin: 15px; border: 2px solid slateblue }
6        .padding { padding: 15px; border: 2px solid darkred }
7      </style>
8    </head>
9    <body>
10
11      <div class='margin'> I have a margin of 15 pixels, look at where my
border is!</div>
12      <div class='padding'> I have padding of 15 pixels, look at where my
border is!</div>
13      <div> Both of the above divs have 15 pixels above and below them, but
notice that the margin has the extra space outside of the border while the
padding has the extra space inside the border!</div>
14    </body>
15  </html>
16
```

Changing Mousetype!

If you want to have a mouse cursor that's different than just the normal mouse, you can create your own! This may be for when your site's loading, it may just be because you want to have something different!

There are a some cursors that you can use that are pre-supplied values: crosshair, default, help, move, text, wait, and resize elements: ([n-resize](#), [s-resize](#), [e-resize](#), [w-resize](#), [ne-resize](#), [nw-resize](#), [se-resize](#), and [sw-resize](#))

```
1  <!doctype html>
2  <html>
3    <head>
4      <style>
5        .border { border: 1px solid black;
6          padding: 25px;
7          width:10px;
8          height:10px }
9        .crosshair { cursor: crosshair }
10       .help { cursor: help }
11       .move { cursor: move }
12       .text { cursor: text }
13       .wait { cursor: wait }
14       .resizeN { cursor: n-resize }
15       .resizeS { cursor: s-resize }
16       .resizeW { cursor: w-resize }
17       .resizeE { cursor: e-resize }
18       .customFox { cursor:
19         url('https://www.cursor.cc/cursor/850/82/cursor.png'), auto}
20       .customRainbow{ cursor:
21         url('https://66.media.tumblr.com/365bb77e17bd28f0e65b884ebd61370c/tumblr_inli
22         ne_mywcw3nnGD1qellpe.gif'), auto}
23       .customThinkingEmoji { cursor:
24         url('https://www.cursor.cc/cursor/236/89/cursor.png'), auto}
25     </style>
26   </head>
27   <body>
28     <div>
29       Anything outside of the table here has a normal cursor!
30     </div>
31     <table>
```

```

28     <tr class='border'>
29         <td class='crosshair border'> CROSSHAIR </td>
30         <td class='help border'> HELP</td>
31         <td class='move border'> MOVE </td>
32         <td class='wait border'> WAIT </td>
33     </tr>
34     <tr class='border'>
35         <td class='resizeN border'> N-RESIZE </td>
36         <td class='resizeS border'> S-RESIZE </td>
37         <td class='resizeE border'> E-RESIZE </td>
38         <td class='resizeW border'> W-RESIZE </td>
39     </tr>
40     <tr class='border'>
41         <td class='text border'> TEXT </td>
42         <td class='customFox border'> CUSTOM: FOX </td>
43         <td class='customRainbow border'> CUSTOM: RAINBOW </td>
44         <td class='customThinkingEmoji border'> CUSTOM: THINKING EMOJI </td>
45     </tr>
46 </table>
47
48 </body>
49 </html>

```

Display!

Display is a very powerful property that we'll use more in the future, once we start working with javascript, but for now, we'll just look at a few things we can do with display.

Turn any non-block element into a block element:

Remember how we used `` to make words bold? Strong only surrounded a given word, and that was it. It didn't take up any more space than it needed. We can change that, though, with the display property. [This works for literally any non-block element!](#)

```

1  <!doctype html>
2  <html>
3    <head>
4      <style>
5        .block{ display: block;
6          border: 2px solid black}
7      </style>
8    </head>
9    <body>

```



```

10     <div>Regularly, when using a <strong>strong</strong>, we don't have to
      worry about it taking up more space than it needs. Sometimes, we want a
      <strong class='block'>strong</strong> to take up way more space than it
      needs!</div>
11     </body>
12 </html>
13
14

```

Vice Versa, sometimes we don't want block elements to take up an entire section! [We can then use the value 'inline' for the display property to take care of that!](#)

```

1  <!doctype html>
2  <html>
3    <head>
4      <style>
5        .inline{ display: inline;
6          border: 2px solid black
7        }
8      </style>
9    </head>
10   <body>
11     <div>Regularly, when using a <div>div</div> it takes up all the space
      because it's a block element! If we use the inline value on the display
      property our <div class='inline'>div</div> won't take up any more than we
      need!
12     </div>
13   </body>
14 </html>

```

Suppose we wanted our inline block to have a certain amount of width? It won't work because display's inline value takes away all [width, height, margin-top, margin-bottom, and float properties](#).

```

1  <!doctype html>
2  <html>
3    <head>
4      <style>
5        .inline{ display: inline;
6          border: 2px solid black;
7          width: 100px;
8        }
9      </style>
10   </head>

```

```

11     <body>
12         <div>Regularly, when using a <div>div</div> it takes up all the space
           because it's a block element! If we use the inline value on the display
           property our <div class='inline'>div</div> won't take up any more than we
           need!
13     </div>
14 </body>
15 </html>

```

To get around this, we can use the [inline-block value](#)!

```

1  <!doctype html>
2  <html>
3    <head>
4      <style>
5        .inline{ display: inline-block;
6          border: 2px solid black;
7          width: 100px;
8          height: 25px;
9          text-align: center;
10       }
11     </style>
12 </head>
13 <body>
14   <div>Regularly, when using a <div>div</div> it takes up all the space
           because it's a block element! If we use the inline value on the display
           property our <div class='inline'>div</div> won't take up any more than we
           need!
15   </div>
16 </body>
17 </html>

```

Float!

Let's say you want your divs to all be right next to each other, but don't want to constantly be using display's inline-block value. We can get around this by using float!

[Remember that divs are block elements, so they take up as much space as possible \(width wise\):](#)

```

1  <!doctype html>
2  <html>
3    <head>
4      <style>
5        .border{ border: 2px solid black; margin: 5px}
6      </style>
7    </head>
8    <body>
9      <div class='border'>I am a block element! </div>
10     <div class='border'>I am ALSO a block element</div>
11   </body>
12 </html>

```

When we use float, though, we're able to get around the whole "divs taking up the entire width of the page, thing" in a more dynamic way than using display: inline-block. [Let's check it out:](#)

```

1  <!doctype html>
2  <html>
3    <head>
4      <style>
5        div { float: left }
6        .border{ border: 2px solid black; margin: 5px}
7      </style>
8    </head>
9    <body>
10     <div class='border'>I am a block element! </div>
11     <div class='border'>I am ALSO a block element</div>
12   </body>
13 </html>

```

Making divs right next to each other is neat, but what's the point? Well, let's say we wanted to have a cool photo blog? Instagram is great and all, but I want to have my own special site. [Let's try to make one of those:](#)

```

1  <!doctype html>
2  <html>
3    <head>
4      <style>
5        div { float: left }
6
7        .border{ border: 2px solid black; margin: 5px}
8
9      </style>
10   </head>

```

```

11     <body>
12         <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/batwing.JPG' /></div>
13         <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/christmasTarantula.JPG' /></div>
14         <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/cicada.JPG' /></div>
15         <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/desertHairy.JPG' /></div>
16         <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/giantFlower.JPG' /></div>
17         <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/glassWing.JPG' /></div>
18         <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/greenJay.png' /></div>
19         <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/leafwing.JPG' /></div>
20         <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/lubber.JPG' /></div>
21         <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/luna.JPG' /></div>
22         <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/luna.JPG' /></div>
23         <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/morphoScales.png' /></div>
24         <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/moth.JPG' /></div>
25     </body>
26 </html>
27

```

We haven't done any extra work to make sure our images are all similar sizes. Let's try again, by adding a [width to our images in css](#)

```
1 <!doctype html>
2 <html>
3   <head>
4     <style>
5       div { float: left }
6       img { width: 500px; height: 500px }
7       .border{ border: 2px solid black; margin: 5px}
8
9     </style>
10  </head>
11  <body>
12    <div><img
13      src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/batwing.JPG' /></div>
14    <div><img
15      src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/christmasTarantula.JPG' /></div>
16    <div><img
17      src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/cicada.JPG' /></div>
18    <div><img
19      src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/desertHairy.JPG' /></div>
20    <div><img
21      src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/giantFlower.JPG' /></div>
22    <div><img
23      src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/glassWing.JPG' /></div>
24    <div><img
25      src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/greenJay.png' /></div>
26    <div><img
27      src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/leafwing.JPG' /></div>
28    <div><img
29      src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/lubber.JPG' /></div>
30    <div><img
31      src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/luna.JPG' /></div>
32    <div><img
33      src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/morphoScales.png' /></div>
```

```

23     <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/moth.JPG' /></div>
24     </body>
25 </html>

```

This looks a lot better. Let's add some padding to get some borders around our images! [Also, let's add a title:](#)

```

1  <!doctype html>
2  <html>
3    <head>
4      <style>
5        div { float: left; padding: 20px; }
6        img { width: 500px; height: 500px }
7        .border{ border: 2px solid black; margin: 5px}
8
9      </style>
10   </head>
11   <body>
12     <h1> My Photo Blog: </h1>
13     <hr />
14     <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/batwing.JPG' /></div>
15     <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/christmasTarantula.JPG' /></div>
16     <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/cicada.JPG' /></div>
17     <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/desertHairy.JPG' /></div>
18     <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/giantFlower.JPG' /></div>
19     <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/glassWing.JPG' /></div>
20     <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/greenJay.png' /></div>

```

```
21     <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/leafwing.JPG' /></div>
22     <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/lubber.JPG' /></div>
23     <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/luna.JPG' /></div>
24     <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/morphoScales.png' /></div>
25     <div><img
src='https://raw.githubusercontent.com/LaneMatthewJ/lanemattthewj.github.io/master/img/moth.JPG' /></div>
26 </body>
27 </html>
28
```

FlexBox!

[Flex box does a lot of this for us!](#) Let's take a look!

- [Simple Layouts](#)
- [Gridlist](#)
- [MORE](#)