

CSS and Styling

TJ Dev Club



How do we get our website from this...

What We Do

Mentoring, Projects, Community

TJ Dev Club meets Wednesdays 8B. Developers of all skill levels are welcome to learn, build, and plug into a supportive developer community.

1. 👤

Mentoring

Collaborate with fellow students and club officers through small groups, guided tutorials, and lectures from experts. Mentor others to help them succeed and earn service hours.

2. </>

Projects

Build for competitions or for yourself. Ship at least one project this year. We'll help you win HackTJ.

3. 💬

Community

Share your work at frequent competitions, make friends, and stay connected on Discord. Our strong community is like no other.



...to this?

What We Do

Mentoring, Projects, Community

TJ Dev Club meets Wednesdays 8B. Developers of all skill levels are welcome to learn, build, and plug into a supportive developer community.



Mentoring

Collaborate with fellow students and club officers through small groups, guided tutorials, and lectures from experts. Mentor others to help them succeed and earn service hours.



Projects

Build for competitions or for yourself. Ship at least one project this year. We'll help you win HackTJ.



Community

Share your work at frequent competitions, make friends, and stay connected on Discord. Our strong community is like no other.



Let's begin with some HTML.

We need some HTML to style. That is, we are going to add color to it, change sizes, fonts, etc.

Open up your HTML file from last week.

If you don't have it, you can use [this template](#)



Find the `<head>` tag, and make sure you put your cursor between the `<head>` and `</head>` .

```
<!DOCTYPE html>
<html>
  <head>
    <title>My website</title>

    <!-- Make sure you are going to work between the <head> tags! -->
  </head>

  <!-- Body code omitted -->
</html>
```



The `<style>` tag

We gotta tell the browser the instructions on how to style the website.

This is going to be done in a new language called **Cascading Style Sheets**, or **CSS** for short.

We put our CSS between `<style>` and `</style>` tags, which **tells the browser we are entering the styling language zone**



The <style> tag

In **between the head tags**, write a new set of `<style></style>` tags.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      /* Get ready to work in here! */
    </style>
  </head>
</html>
```

Notice that in CSS, our comments are surrounded by `/* */` instead of `<!-- -->`.

Remember that comments are notes for us and other humans and are ignored by the computer.



Milestone

Check with the officer and make sure you've loaded your HTML correctly and put your tags in the right place.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- <style> tags between <head> tags -->
    <style>
      /* CSS between <style> tags */
    </style>
  </head>
</html>
```



Thinking like the computer

Let's consider our first goal: we want to change the background color to something other than default white.

Think about what information the computer needs or what we need to tell the computer in order for it to achieve that goal.

Obviously we need to tell it what color we want the background to be set to, but *first* we need to select the element that we want to change the background of.



Thinking like the computer

There are so many elements on this page (the h1 element, the two p elements, the image element, the div element, the body element, etc.)

We need to choose one to style.



Changing background color

In our style tag, let's select the `<body></body>` element by writing in "body".

We then put a set of curly braces `{ }`. We will put our instructions on how to style the page inside those braces later.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {
      }
    </style>
  </head>
</html>
```



Changing background color

Now, let's actually put an instruction inside the curly braces. Each instruction is called a *declaration*. It follows the format: `PROPERTY: VALUE;`

Let's set the `background-color` attribute to `blue`.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {
        background-color: blue;
      }
    </style>
  </head>
</html>
```



Milestone

Check with the officer and make sure the background is blue.

```
<!doctype html>
<html>
  <head>
    <title>My website</title>
    <style>
      body {
        background-color: blue;
      }
    </style>
  </head>
```



Changing the font color

It's hard to read the black text with this blue background.

Try on your own to set the `h1` tag's font color to white, using the `color` property.

You can either set the value to `white`, or specify a color in *hexadecimal* format. In this case, pure white is `#ffffff`.

You can get the hex code from a color picker, such as the [Google color picker](#).

Check in with your officer once you have done this.



Changing the font color

You should have:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {
        background-color: blue;
      }

      h1 {
        color: #ffffff;
      }
    </style>
  </head>
</html>
```



The `*` selector

What if we want to change the font color for all elements? It would be tedious or even impossible to select each element individually.

Instead, we can select all elements using `*`, instead of the name of an element.

Try to change the font color of all elements to white right now.



Changing the font color

You should have:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {
        background-color: blue;
      }

      * {
        color: #ffffff;
      }
    </style>
  </head>
</html>
```

We removed the `h1` styling because it is no longer necessary.



The meaning of "Cascading"

Add back a declaration block for `h1` elements. Set the font color to red.

```
body {  
  background-color: blue;  
}  
  
* {  
  color: #ffffff;  
}  
  
h1 {  
  color: red;  
}
```

We now have two declarations that changes the font color for `h1` elements: the one selected by `*`, and the one selected by `h1`.

Which one is used?



The meaning of "Cascading"

Notice that now the h1 tag is now red, not white.

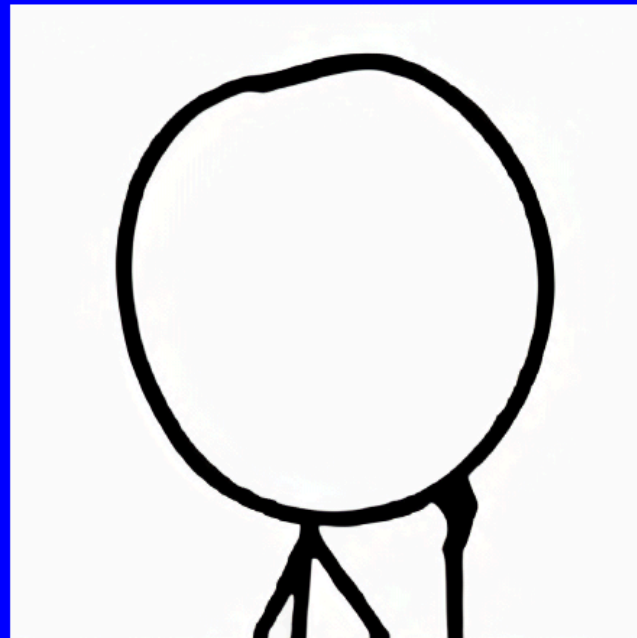
The most specific declaration is used for styling.

This is what "cascading" means: authority starts from the most broad declaration and "cascades" down to the most specific.

My website!

Hi! This is where I write about it in my blog.

Picture of me:



class and id

The "Picture of me:" text should be special. Let's change the color of only that text to yellow.

Let's try to do it by selecting the `p` tags, and coloring them green. Do you see the problem?

```
p {  
  color: yellow;  
}
```



class and id

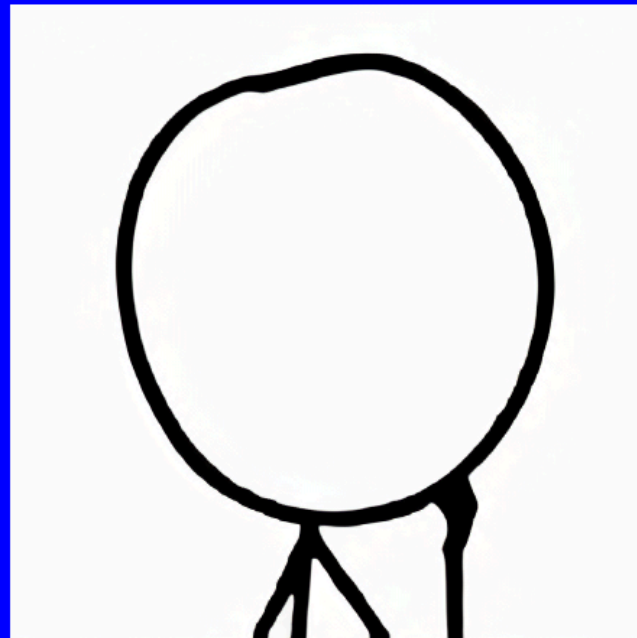
Ignoring the readability problem, notice that ALL `p` elements have green text.

We need some way to specify which element we are selecting, some *identifier*.

My website!

Hi! This is where I write about it in my blog.

Picture of me:



class and id

To do this, we need to go back into the HTML.

Find an element you want to style. Then, add the HTML attribute `id=""`. Put a relevant and useful identifier between the quotes. Make sure not to use spaces.

```
<body>
  <div>
    <p id="picture-caption">Picture of me:</p>
  </div>
</body>
```



class and id

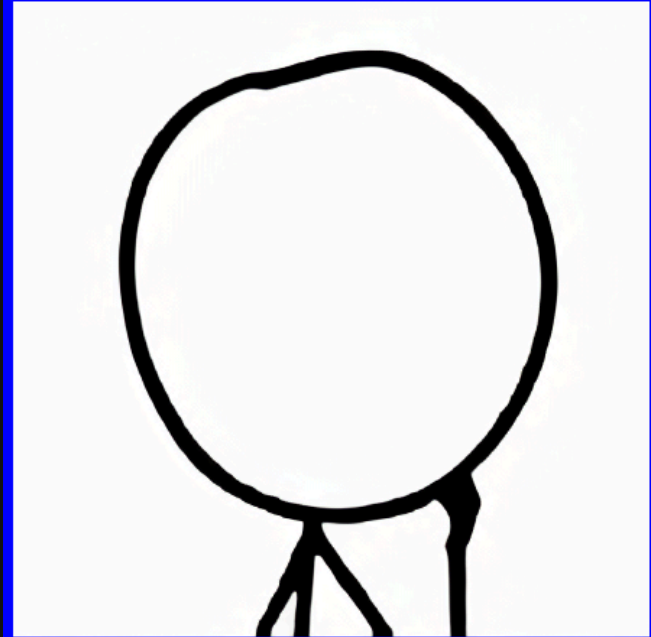
Notice now only the "Picture of me" text is green.

Check in with your officer and make sure only the selected element with the id is green.

My website!

Hi! This is where I write about it in my blog.

Picture of me:



class and id

What if we want to style multiple elements the same way?

We can't repeat the id attribute; it's unique to only one element:

```
<!-- THIS DOES NOT WORK!!! -->  
<p id="green-color">Some body text</p>  
<p id="green-color">Some more body text</p>
```



class and id

Instead, replace `id` with `class` :

```
<p class="green-color">Some body text</p>  
<p class="green-color">Some more body text</p>
```

and then...



class and id

In our CSS selectors, use `.` instead of `#`.

```
.green-color {  
  color: green;  
}
```

Remember: Nothing selects element type, period is class, hashtag is id



Cool CSS properties

We will now just go over some cool CSS properties to make your site look modern.

You can find more on Google, or at [MDN Web Docs](#)



Border

Add some borders to ground your divs and make it clear that things are segmented.

```
<style>
  div {
    border-style: solid; /* Solid border; other options include none, dotted, double, inset */
    border-width: 2px; /* Set border thickness to 2px */
    border-color: #999999; /* Set border to a nice gray color */
  }
</style>

<div>Hello.</div>
```



border-radius

Nothing says "modern" more than rounded corners.

```
<style>
  div {
    border-radius: 7px;
    border-style: solid;
    border-width: 2px;
    border-color: #999999;
  }
</style>

<div>Hello.</div>
```



box-shadow

Add a nice shadow to make things look floating. Usually you can just use a [box shadow CSS generator](#) to do most of the work for you; just copy the box-shadow declaration.

```
<style>
  div {
    box-shadow: 0px 0px 8px 0px □ rgba(0, 0, 0, 0.82);
  }
</style>

<div>Hello.</div>
<br />
<div>Hi.</div>
```



font-family

Changes the font to something else.

WARNING: These will only work if the font is installed on the user's computer! We will talk about how to add fonts in webpages in a future lesson.

```
<style>
  .trebuchet {
    font-family:
      "Trebuchet MS", sans-serif; /* Use Open Sans if it exists, otherwise, use the default "sans-serif" font. */
  }
</style>

<div>This is the default font.</div>

<div class="trebuchet">This is Trebuchet MS.</div>
```



Mini-project

Make your site or the template site look more modern.

Ideas: Add dark mode, make the colors nice, add dividers, add shadows, round corners

Next week: Sizing elements, how to layout and position our elements.



`</style>`

