Table of Contents

1. This Sushi Card series uses Trinket for writing the website code. If you do not have an account on Trinket, it is recommended that you create one. It is free: all you need is an email address. Go to dojo.soy/html2-trinket to sign up.

2. To work through the examples in the Sushi Cards, **Remix** this trinket: dojo.soy/html2-website-start. Then click on your username and go to "My Trinkets". You will see the remixed project there. Open it and change the name if you like, then click "Save".

3. The Sushi Cards mainly use examples from the above trinket, but if you prefer you can work with a website that you already made, for example the one that you made in the Beginner HTML and CSS series. To make a copy of your own website in Trinket, open it and click the "Duplicate" button. Change the name if you like, and then click "Save".

4. Just above the tabs in the code panel, click the icon and select either "Click to Run" or "Autorun".
   - *Click to Run*: Recommended. Any time you have made changes to your code, click the button to see the results.
   - *Autorun*: Your website will be automatically updated every time you make a change to the code. It doesn't always wait for you to stop typing, so this usually results in a lot of flashing of the page while you type! It also thinks there's errors when you haven't finished typing yet, which can be confusing.

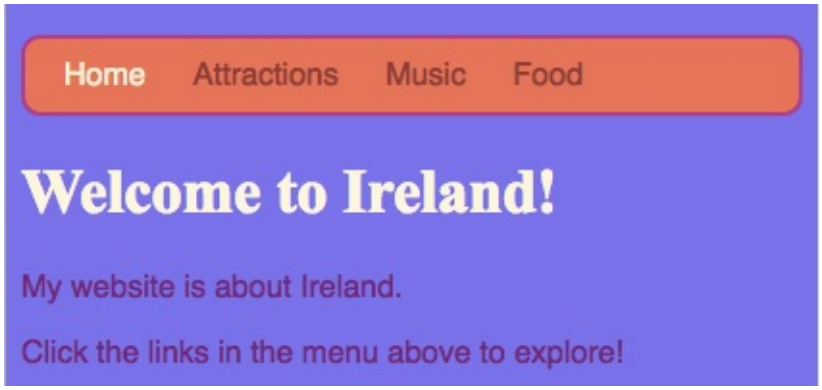
1 Take a look at your **stylesheet**. That's the file that has **.css** in the name.

2 Inside the CSS rules for **body**, set the background colour to

```
background-color: #7B68EE;
```

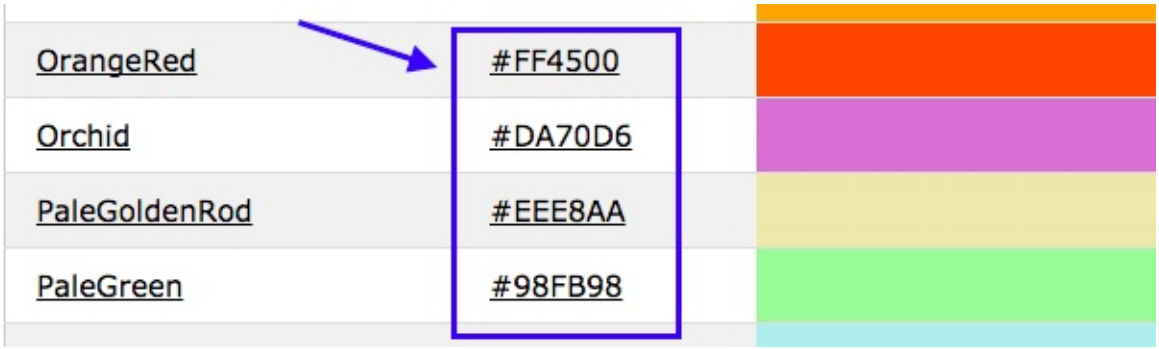
- If you are using a Mac, the **#** is typed by holding the **alt** key and the number **3** key together.

3 When you click Run, your website should have a purple background.



As you have seen, you can type in many different colour names as words and the browser will recognise them. But a more common way to set colours is to use a code like the one above, called a **HEX** code.

4 Try it yourself: Go to dojo.soy/html2-colors and choose a colour for your website. But instead of typing the name of the colour, type in the **HEX** code.



5 HEX codes allow you to mix any colour, even if it's not on any list of colours. Try making up your own colour code

- The code must start with a **#**. This tells the browser that it is a HEX code instead of a colour name.
- The rest of the code is made up of six characters. They can be any number from **0-9** and any letter from **A-F**.

6 How does it work? Every colour is made by mixing three colors: **Red**, **Green**, and **Blue**. You may have seen this written down as **RGB**. Each of these colours is represented by two of the six digits in your code. The higher the number, the more of the colour there is.

- What's going on with those letters?! **Hexadecimal** is a special way of counting that uses the letters A-F as **extra digits** higher than 9. This makes the numbers shorter to write. Colours go from **0** up to **255**, or in Hexadecimal: from **00** up to **FF**. If you wrote the colour *#9ADC32* using plain RGB numbers, it would be *rgb(154,205,50)*.

7 You don't need to worry too much about learning to count with Hexadecimal numbers, but do experiment to get used to a few colours! Here are the three most basic colours. Try setting your background to each of them in turn.

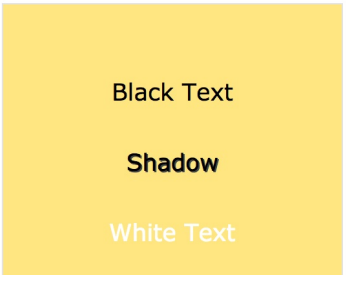
- *#FF0000* is red
- *#00FF00* is green
- *#0000FF* is blue

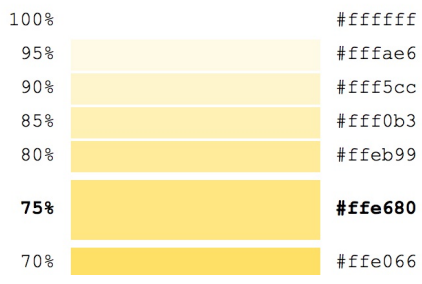
What do you think will happen if you use **less** of a colour? Change *FF* to something smaller like *88* or *33* in each of the above colours and see.

8 Let's try some mixing. What do you think you will get if you mix **red** and **blue**? You do it like this: *#FF00FF*. How about **red** and **green**: *#FFFF00*? Try it out!

- Try mixing with more of one colour and less of the other, for example *#FF8800*

9 There are plenty of colour picking tools that help you get the HEX code for any colour you want.





Go to dojo.soy/html2-color-picker and try out the colour picker to choose some HEX colour codes to use for the rest of the styles on your website.

10 Fun exercise: Take your six-digit date of birth and put a **#** symbol in front to make your own unique personal colour code! Try it out in your stylesheet. What colour did you get?

1 So far you've used **headings** and **paragraphs** to make your **content** look tidy and easy to read. Let's make it even more organised by grouping things together.

- **Content** is all the "stuff" on your web page, like information and pictures

2 Go to the attractions.html file (or one of your own pages if you're not using the example Trinket project). Near the top, just *underneath* the opening `<main>` tag, type the following on a new line:

```
<main>
  <article>
```

Delete the closing tag that Trinket automatically adds in for you.

3 At the bottom of the file, just *above* the closing `</main>` tag, add a new line and close the **article** element:

```
</article>
  </main>
```

4 Think of the **article** element as a **container** for a piece of content, in this case a set of information about attractions in my home country, Ireland. If you have different bits of content that aren't related, you should put each one into its own **article** element instead of putting one set of the tags around the whole lot.

- Remember when you created a menu and then put it in between `<nav> </nav>` tags? That's another example of a type of container. What you are doing is telling the browser that everything in between these tags belongs together. Like organising things in boxes and shelves in your home!

```
<article>
  <h1>My favourite places to see in Ireland</h1>
    <h2>The Cliffs of Moher</h2>
    <p>
    The Cliffs of Moher are found in County Clare, where I am from. Look how cool they are!</p>
    <img src="cliffs.JPG" alt="The Cliffs of Moher" height="200px" />
    <h2>Achill Island</h2>
    <p>This is a large island off the coast of County Mayo. It has a wild and
    beautiful landscape of mountains, bogs and cliffs.
    </p>
    <img src="achill.JPG" width="200px" />
</article>
```

5 Now look at the content in your **article**: can you break it up into chunks or sections? Another HTML element called **section** lets you do exactly this! I've put the information about each different attraction in between its own set of `<section> </section>` tags:

```
<article>
  <h1>My favourite places to see in Ireland</h1>
  <section>
    <h2>The Cliffs of Moher</h2>
    <p>
    The Cliffs of Moher are found in County Clare, where I am from. Look how cool they are!</p>
    <img src="cliffs.JPG" alt="The Cliffs of Moher" height="200px" />
  </section>
  <section>
    <h2>Achill Island</h2>
    <p>This is a large island off the coast of County Mayo. It has a wild and
    beautiful landscape of mountains, bogs and cliffs.
    </p>
    <img src="achill.JPG" width="200px" />
  </section>
</article>
```

6 Once your content has been organised into **articles** and **sections** you can create CSS rules in the stylesheet to control how the different bits look! Here's some example CSS styling. See if you can understand it and then make some of your own.

```
section {
  border-top-style: solid;
  border-top-width: 2px;
  border-top-color: #F5FFFA;
  padding-bottom: 10px;
}

article {
  border-radius: 10px;
  background-color: #48D1CC;
  padding: 10px;
}
```

7 Have a go at organising all of your HTML files in this way. In my example I've added an **article** with a bunch of **sections** onto the Food page. On the next card you'll design a different theme for each page that's organised into **articles** and **sections**!

1 Let's design a theme for the Attractions page that's different from the other pages. Go to your stylesheet file and add the following. Be sure to include the dot in front!

```
.pupleBackground {
    background-color: #9f80ff;
}
```

The dot makes this into a **class selector**. With a CSS **class** you can apply a set of CSS rules to any element on your website by adding an **attribute** to the element.

- You could pick the colour to match the background of your page!

2 Now go to *attractions.html* (or the page you're working on if you're using your own project) and add the following **attribute** to the **article** tag:

```
<article class="purpleBackground">
```

You should see the background change around the text on the page.

3 Add the following class to your stylesheet:

```
.greenBackground {
    background-color: #48D1CC;
}
```

Back in the **html** file, add the **attribute** `class="greenBackground"` to each of the **section** tags, just like you did for the **article** element. Click Run and look at the Attractions page. Compare it to the other pages that have **section** elements.

4 When you use a CSS **element selector** such as **section**, the style rules apply to *all* the elements of that type on your website. With CSS **classes** you're able to just change *some* of them. You can add more than one class to an element too. Let's make another CSS **class** to give the sections some margin and padding. In *styles.css*, add the following code:

```
.sectionSpacing {
  padding: 10px;
  margin-top: 20px;
}
```

5 In the **html** page you were working on, add the new class to each of the sections like this:

```
<section class="greenBackground sectionSpacing">
```

You can add as many CSS classes to an element as you like. Just write the names of all the classes you want to use inside the **class attribute**, and separate them with spaces.

6 How about re-vamping the Food page? (If you're using your own website, you'll need another page with some **sections** on it) Add the following classes to your stylesheet:

```
.yellowBackground {
  background-color: #FFFFCC;
  color: #A52A2A;
}

.solidRoundBorders {
  border-style: solid;
  border-width: 2px;
  border-color: #F5FFFA;
  border-radius: 10px;
}
```

Go to the *food.html* file (or your own file) and add the new classes as well as the `.sectionSpacing` class to each section:

```
<section class="yellowBackground sectionSpacing solidRoundBorders">
```

7 Notice how you can mix and match the classes in your element's class attribute? Using CSS classes lets you **reuse** the same set of style rules on lots of elements. You can apply them to any element that has the properties you've set. Go to *index.html* and add the *solidRoundBorders* class to the picture of Tito.

```
<img class="solidRoundBorders" src="tito.png" alt="Tito the dog" width="100px" />
```

8 Why not experiment with some more CSS classes to make the other pages have their own themes? You can name a CSS class anything you like. It's common to give it a name that describes what it does or what it's for!

9 Extra challenge: use CSS **classes** to define a few different picture sizes for the website, for example `.smallPictures` and `.mediumPictures`. Then remove the **width** attribute from each of your **img** elements and add the appropriate class instead.

1 Let's jazz up the homepage a bit! Go to *index.html* and add the following **attribute** to the first paragraph tag:

```
<p id="myCoolText">
    My website is about Ireland.
</p>
```

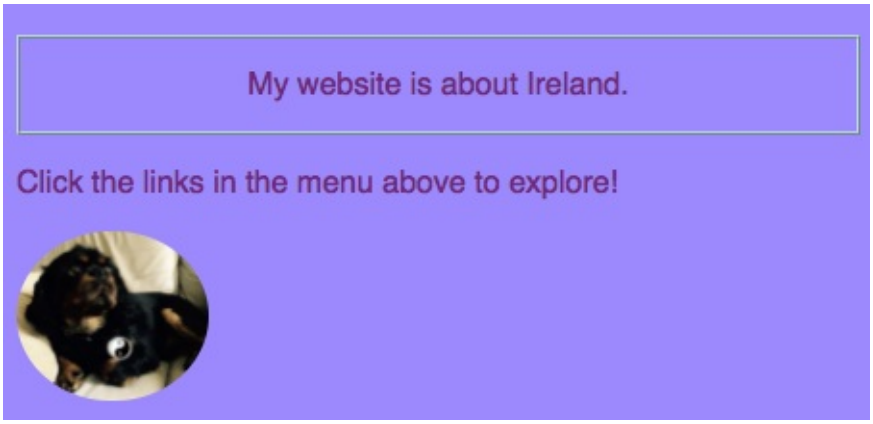
The **id** is a name you give to **identify** this particular element. No two elements on a page should ever have the same **id**.

2 Now go to your stylesheet and add the following code

```
#myCoolText {
    color: #000080;
    border: 2px ridge #add8e6;
    padding: 15px;
    text-align: center;
}
```

Here you're using an **ID selector**. You can tell because of the **#** in front of it.

- The name should *exactly* match the name you put in the element's **id** attribute



3 By giving an element an **id** and using an **ID selector** in your stylesheet, you can make *unique* CSS rules that just apply to that one specific thing on your website. Let's do one for the **body** of the homepage. Go to *index.html* and add an **id** to the **body** tag and one the **article** tag.

```
<body id="frontPage">
```

```
<article id="frontPageArticle">
```

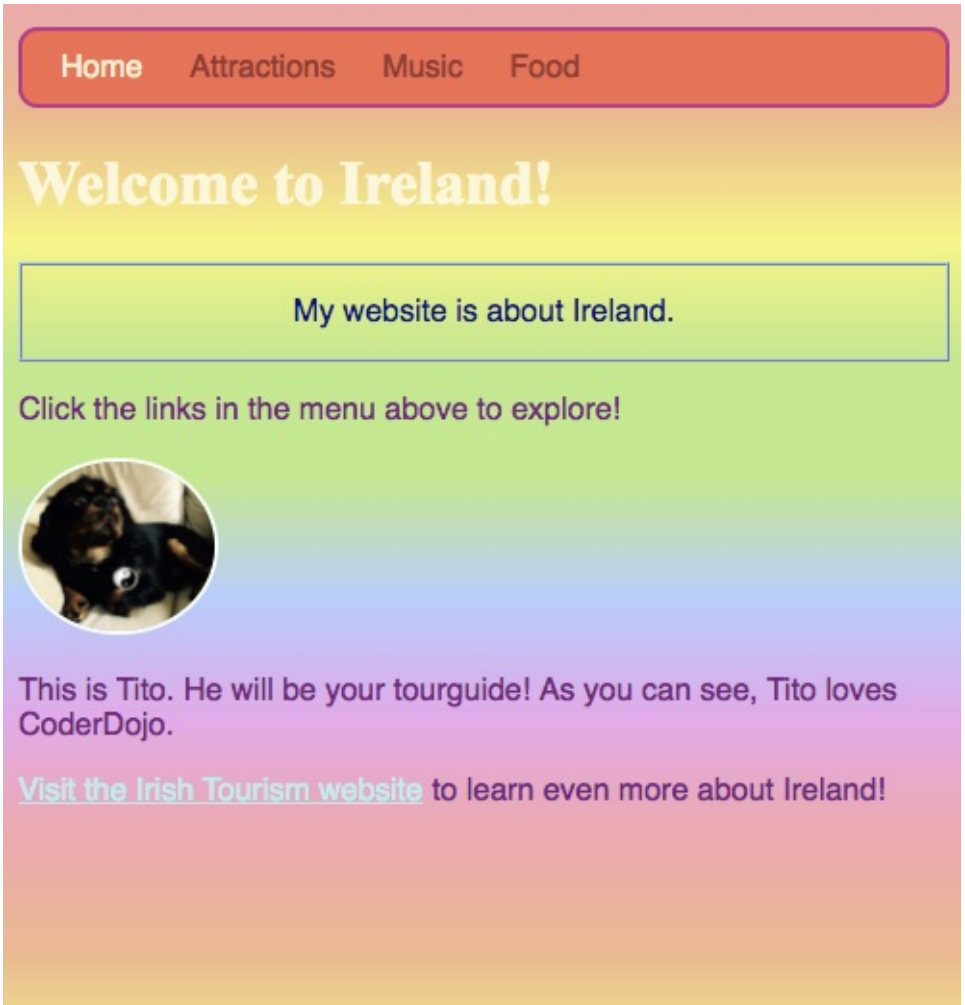
4 In the stylesheet, add CSS rules for the body of the homepage.

```
#frontPage {
    background: #48D1CC;
    background: linear-gradient(#fea3aa, #f8b88b, #faf884, #baed91, #baed91, #b2cefe, #f2a2e8, #fea3aa);
}
#frontPageArticle {
    background: none;
}
```

You just used a **gradient**! You can make lots of different effects with gradients. If you want to learn more, go to html2-css-gradients.

- The first **background** property sets a default colour for browsers that don't support gradients.

5 Try giving another element an **id** and styling that element using the id selector with **#** as above. How about making one picture have a **border-radius** of *100%* so that it's fully rounded? Any other pictures on the website will stay the same as they are.



1 Here's a neat trick that will make trying out different styles quicker and easier. Before you start, make sure to click Save on your project. Then on your web page (the actual page, not the code) right click on some text and select the option **Inspect** from the menu that appears.

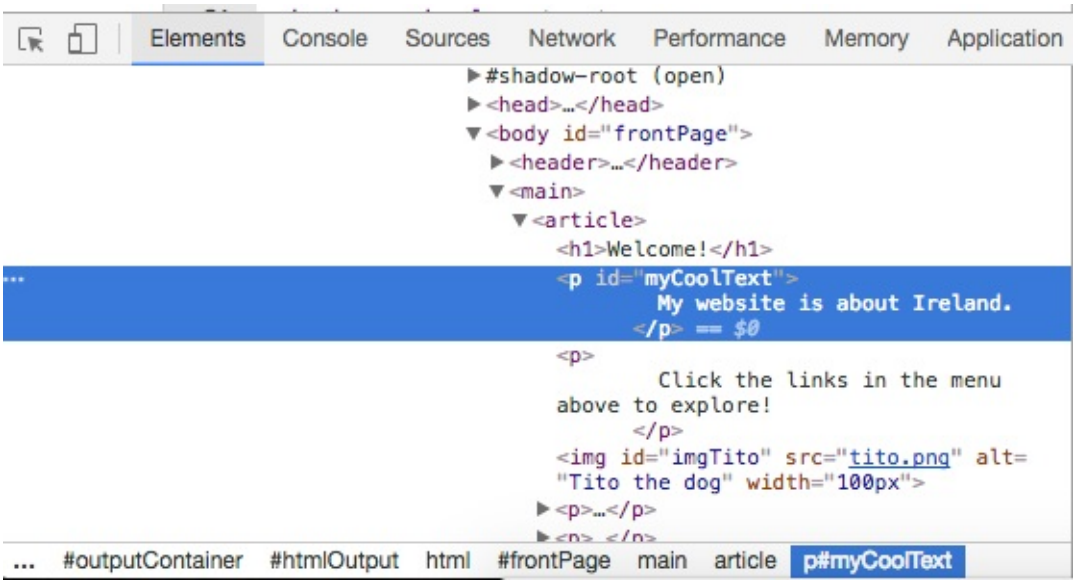
2 A whole new pane will appear in your web browser with lots of tabs and code: the **Developer Tools**, or **Dev Tools** for short. Here you can see the code for the thing you clicked on, as well as the code for the whole page!

3 At the top right of this new pane, click on the three dots next to the **x** icon. In the menu that appears, look at the top item, the one that says "Dock side". Choose the **bottom** option.

- When you're familiar with the tools, you can choose to move the panel wherever you prefer.

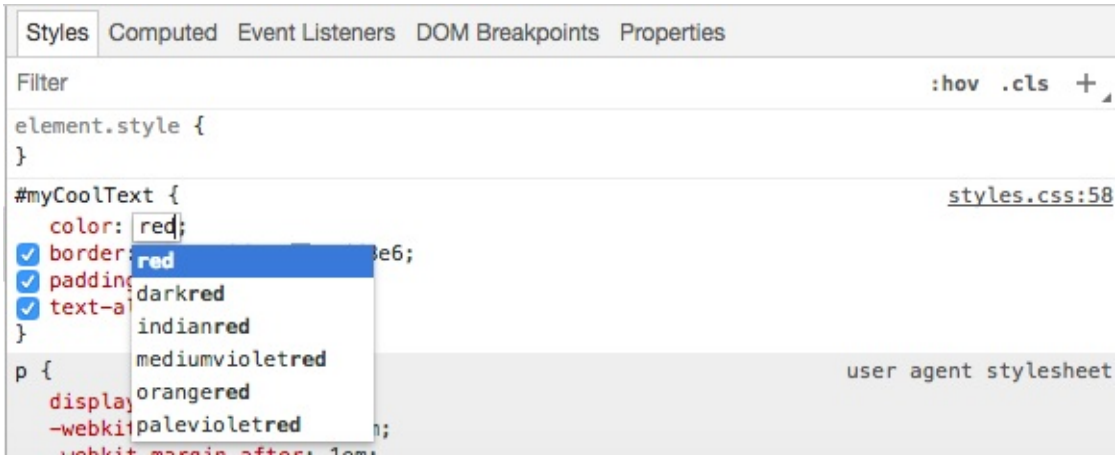
4 Press the **Esc** key on your keyboard once or twice until the Dev Tools is split into two main panels rather than a super-crowded three (you don't need to see the Console right now). The panels you will be using are **Elements** on the left and **Styles** on the right. You can drag the border in between them to change their sizes. You can also drag the top border of the Dev Tools to make it bigger or smaller.

5 In the top left corner of the Dev Tools, click the tiny square icon with the arrow. This lets you select any element on the page to see the code. Click on the new text on your webpage, "This is a paragraph with its own special style!". The **HTML** code will be highlighted in the **Elements** panel of the Dev Tools. It should look pretty much the same as how you typed it in your html file!



6 In the **Styles** tab on the right panel of the Dev Tools you will see the **CSS** code for the selected element. Notice how there are a few sets of CSS rules. One is the block you created specifically for that paragraph: *#myCoolText*.

7 Here's the fun part! Click on the colour name or the HEX code next to the **color** property of *#myCoolText*. Now you can type in a different value. Try *red*. If you like you can choose one of the suggestions that appear while you type. Watch the text on your webpage change colour straight away!



- You can also click the coloured square instead of the text: this lets you use a colour picker tool.

8 How is this special? Surely it's the same as changing your code?! To see why it's different, click Run on your project... or even Refresh the page. Notice how the text went back to its original colour?

- When you change code using the Dev Tools you are **temporarily** changing what it looks like **in your browser**. You aren't actually changing the files in your website. This is handy because it means you can **test** out loads of different values and **instantly** see what they look like, *without touching your code files*.

9 Click in the space after the colour. A new line starts, where you can type more CSS. Enter the following:

```
background-color: LightBlue;
```

Try adding or changing some more styles, like the border. If you want to delete a style, untick the little blue box beside it, you'll see it get crossed out.

10 Have a scroll through the **Elements** panel and see if you can locate other elements you recognise from your own code. They should become highlighted on your webpage as you move over the code with your mouse. When you click an element, its style rules will be shown on the right.

- Click on the tiny grey triangles to expand elements that have more code inside them.

11 You can use the Developer Tools to check out the code on other websites too! Why not visit a few websites, right click and Inspect. See if you understand any of the code you see. You can even make temporary changes! It's not going to change the website itself: Everything you do here is **local** (in your browser), which means only you see it.

- Remember, all these changes are **temporary**. When you reload the page or re-run your code, it will all reset.

1 Up until now you've been using **pixels** to set the size of things, e.g. `10px`. This is called an **absolute** measurement. It means you set an exact size and it doesn't change. Another way to set the size of things is using **relative** measurements. That means how big elements are in relation to each other. So when one thing changes size, everything else will automatically change as well to keep the same **proportions**.

2 Go to index.html and find the **img** element with the picture of Tito. Delete the width attribute if `width="100px"` if it's there, and give the element an **id** of *imgTito*.

```
<img id="imgTito" src="tito.png" alt="Tito the dog" />
```

3 In your CSS file find the `#imgTito` code block and add the `width` property below:

```
#imgTito {
  border-radius: 100%;
  width: 50%;
}
```

50% (50 percent) is **half**. Try a few different percentages and see if you can work out what it's doing.

4 When you're using **relative** measurements it's important to know what the **parent** of your element is. The **parent** is the thing that your element is inside, and that's what the measurement will be in relation to. For example, **parent** of the image above is the **article** element because the **img** element is in between the `<article></article>` tags.

5 Now in your html file, put the image inside a **section** element with **id** *titoSection*. Include some text in the section too.

```
  <section id="titoSection">
  <img id="imgTito" src="tito.png" alt="Tito the dog" />
  <p>
    This is Tito. He will be your tourguide! As you can see, Tito loves CoderDojo.
  </p>
</section>
```

6 Add the following code to your stylesheet, so you can see what's going on more clearly

```
#titoSection {
 width: 200px;
 background-color: white;
}
```

- If you don't give **section** elements a width, they fill up all the space available, which is usually a good thing!

7 You should notice that the picture is much smaller now. That is because it is taking up 50% of the width of the **section** element instead of the **article** element (which is roughly the width of the page).

8 Another **relative** unit of measurement is `em`, which is related to the size of your text. **Cut** the heading `<h1>Welcome to Ireland!</h1>` and **paste** it on a new line *above* the opening `<article>` tag, so that it's outside of the `article` element. Add a class **attribute** to the heading as well.

```
<main>
  <h1 class="funsize">Welcome to Ireland!</h1>
  <article id="frontPageArticle">
```

Then, in your CSS file, add the following code. There is a border so you can see the spacing more easily.

```
.funsize {
  border: 2px solid #FFFFFF;
  padding: 5px;
  font-size: small;
}
```

9 Set the font-size to different values for example *smaller*, *small*, *medium*, *larger*, *xx-large*. Can you see that the **padding** (space between the border and the text) doesn't change?

10 With the padding staying the same, the space starts to look small as the text gets bigger. Let's give it a **relative** value instead of the **absolute** pixel value. Change it to

```
padding: 1em;
```

11 Now experiment again with different values for **font-size**. You should see the spacing change to match the font size this time.

- Setting the size of anything to *1em* makes it the same size as the text; setting it to *2em* makes it twice the size of the text, and so on (specifically, *1em* is the width of the letter *m*!).

12 You can use **em** values for anything you can set the size of, not just padding. Experiment with using it instead of **px** on borders, or instead of **%** on your image!

13 Once you're done experimenting, delete the **section** tags you added around the image and also delete the *#titoSection* CSS block! Set the size of the Tito picture back to *100px* as well.

1 Go to the bottom of your CSS file and add the following code

```
@keyframes myFirstAnimation {
    from {
        width: 100px;
    }
    to {
        width: 300px;
    }
}
```

This code creates an animation called "myFirstAnimation" that you can add to any element on your website. What do you think it does?

2 Find your CSS rules for the ID *#imgTito* and add the following three properties:

```
animation-name: myFirstAnimation;
animation-duration: 2s;
animation-iteration-count: 1;
```

3 Run your code to see what happens! Try different values for **animation-iteration-count** to see what it does.

4 Let's try another animation! Add the following code to the end of your CSS file:

```
@keyframes glowPulse {
    0% {
        color: DeepSkyBlue;
    }
    50% {
        color: LightGreen;
    }
    100% {
        color: DeepSkyBlue;
    }
}
```

5 Now find the *#myCoolText* CSS rules from earlier and add in the animation code:

```
#myCoolText {
        animation-name: glowPulse;
        animation-duration: 1.5s;
        animation-iteration-count: 1;
}
```

6 When you use **percentages** instead of **from** and **to**, you're able to set in between values as well as just start and end values. You can set as many in between values as you like using different percentages. Try adding in more colours to the glowing sequence above, for example at *25%* and *75%*!

7 Change the value of **animation-iteration-count** to *infinite*. What do you think will happen? You can also play with different values for **animation-duration** to speed up or slow down your animation.

8 One final trick! Add this animation code:

```
    @keyframes slide {
    0% {
        background-position-x: 0;
    }
    100% {
        background-position-x: 600vw;
    }
    }
```

Now find the *#frontPage* CSS rules from earlier and change them to:

```
    #frontPage {
        background: repeating-linear-gradient(-45deg, red 0%, yellow 7.14%, lime 14.28%, cyan 21.42%, cyan 28.56%, blue 35.7%,
magenta 42.84%, red 50%);
        background-size: 600vw 600vw;
        animation: slide 10s infinite linear forwards;
    }
```

Don't worry about understanding *all* of it... just sit back and enjoy!!

9 Try making your own new animation using the **@keyframes** keyword and a name that you choose. How about animating your nav menu? To learn about more things you can do with animation, visit dojo.soy/html2-css-animation. Have fun!

1 You can make your website more **interactive** by making cool stuff happen when you hover over things with the mouse! Find your CSS rules for the **img** elements, or create some if you don't have any. Add in a border, and then add a new block of rules right underneath:

```
img {
  border: 2px solid White;
}
img:hover {
  border: 2px dashed Navy;
}
```

You've just used a special type of CSS block called a **pseudo-class**. The `:hover` bit is the **pseudo-class**. It's a bit like a **class**, but it comes built in: you can add `:hover` style rules to any **element**, **class** or **id** selector in your stylesheet without needing to add anything extra in your HTML code.

2 What do you think will happen? Check what pages on your website have pictures on them (add a picture if there aren't any!), then run your code and move the mouse over a picture to find out!

3 Let's use this new `:hover` **pseudo-class** together with a **CSS class** to make links glow when you hover over them! Add a link to your web page and include an **attribute** to specify the **class** name. Remember, links are defined using the **a** tag.

```
<p>
  Visit the <a class="niceLinks" href="https://en.wikipedia.org/wiki/Ireland">Wikipedia page</a> to learn even more about Ireland!
</p>
```

4 Add the following code to your stylesheet, then run your code to see your lovely links in action.

```
.niceLinks {
  text-decoration: none;
  color: magenta;
}
.niceLinks:hover {
  color: deepskyblue;
}
```

Why not add the **attribute** `class="niceLinks"` to all of the links in your menu bar as well?

5 You can combine all of these tricks with animations too! Find the *#imgTito* CSS block in your stylesheet again (if you are working with your own project you can use any picture and *id*). Add the following code to your stylesheet file:

```
#imgTito {
  border-radius: 100%;
  width: 100px;
}
#imgTito:hover {
  animation-name: rollOver;
  animation-duration: 1s;
  animation-iteration-count: 1;
}
@keyframes rollOver {
  0% {
    transform: rotate(0deg);
  }
  100% {
    transform: rotate(-360deg);
  }
}
```

Can you guess what will happen?

6 Run your code and shout "Roll over!" as you move the mouse over the picture!

7 Can you use the *glowPulse* animation from the previous card to make the *niceLinks* keep changing colours when the mouse is hovering over them? (Hint: use a value of *infinite* for the **animation-iteration-count**)

Claim your badge on the CoderDojo website!

Why?
Have a record of your achievement you can share with your friends and family!

How?
Visit **dojo.soy/htmli-badge** and follow the instructions on the page.