

Block and Inline Elements

Block vs. Inline

Block elements always start on a new line and take up the full width available. They also have a top and bottom margin. Examples of block elements are `div`, `p`, `article`, `section`, `h1-h6`, `ol`, `ul`, `li`, `form`, `main`, and `nav`.

Inline elements do not start on a new line and only take up as much width as necessary. Examples of inline elements are `img`, `a`, `span`, and `button`.

Example 1

In this example, we will look at the block and inline elements on our page.

Let's copy the following code into an HTML file. (If you've completed other learning modules you may already have these files. You can just add to them.)

```
<!DOCTYPE html>
<html>
<head>
  <title>BYUI</title>
  <link rel="stylesheet" href="styles/styles.css">
</head>
<body>
  <header>
    <nav>
      <a id="active" href="byui.html">Home</a>
      <a href="https://www.churchofjesuschrist.org">Church of Jesus Christ</a>
      <a href="https://www.churchofjesuschrist.org/temples">Temples</a>
      <a href="https://www.churchofjesuschrist.org/study/scriptures">Scriptures</a>
    </nav>
  </header>
  <main>
    <h1>Brigham Young University Idaho</h1>
    
    <p>BYU-Idaho is a comfortable place to learn and grow as a disciple of Jesus Christ because students, faculty, and employees share a commitment to live the gospel.</p>
  </main>
  <footer>
    <p>My Web Page 20XX &copy;</p>
  </footer>
</body>
</html>
```

Let's create a CSS file. Create a folder at the same level as our HTML file and call it styles. Within that folder, let's create a styles.css file. Open that styles.css in the editor. (Again, if you already have this CSS file you can just add to it.)

Remember if you already have CSS rules targeting an element that you will be asked to give a new rule to, it's best to replace or combine your declarations with the rule you already have in CSS. This makes our code clean and efficient.

Our page has some elements that are inline and some that are block. The `img` and `a` tags are inline elements. That is why the nav links line up one after another on one line. If we had two images with smaller width next to each other they would also line up next to each other as well. Let's add another image and give them a small width so you can see that. Copy and paste the `img` tag so you have two of the same image.

```
<main>
  <h1>Brigham Young University Idaho</h1>
  
  
  <p>BYU-Idaho is a comfortable place to learn and grow as a disciple of Jesus Christ because students, faculty, and employees share a commitment to live the gospel.</p>
</main>
```

Let's give them both a width of 30%. You may have to replace a width value you already had.

```
img {  
  width: 30%;  
}
```

Notice that both images are on the same line because they have room. Now that we've seen that images are inline, take the extra image back out of your HTML.

```
<main>  
  <h1>Brigham Young University Idaho</h1>  
    
  <p>BYU-Idaho is a comfortable place to learn and grow as a disciple of Jesus Christ because students, faculty, and employees share a commitment to live the gospel.</p>  
</main>
```

And change your image width to 80%.

```
img {  
  width: 80%;  
}
```

Many of the other elements like nav, h1, and p are block-level elements. This means that they take up all the room they can. Let's put a border around every elements so we can see this easily.

```
* {  
  border: 2px solid blue;  
}
```

The nav and h1 or example take up all the room available from left to right. The paragraphs also take up all the room they are given. If they have a width given to them, then they take up as much space within that width as they can. Notice the image and a elements that are inline have a tight border around them.

Example 2

In this example, we will be centering block and inline elements on our page.

Let's delete any body centering from past weeks so we can practice centering different elements on our page.

Delete the following if you have it in your styles.css file:

```
body {  
  text-align: center;  
}
```

We just deleted the centering we had for the entire body that many of the elements inherited. Let's center some elements individually now.

To center content inside their container, we can use text-align: center. We can center our h1 with text-align: center and it will center within it's container because there is space left over after the text and before the end of the element.

You may have already had the h1 rule with a color property and that's fine just leave it and add the text-align: center to the h1 rule set.

```
h1 {  
  text-align: center;  
}
```

We can do the same centering with our nav and footer because they are also block level elements.

Again, that rule set may have other declarations that you can just leave and add the text-align: center to them.

```
nav, footer {
  text-align: center;
}
```

If we try to center the `img` element the same way, with `text-align: center`, it won't work because there is no extra space inside it's container. Text-align center will center the content of its container. With images we want to center the container itself. To give it some extra space, let's have it take the whole amount available and change its display property to `display: block`. Now to center it we can center the container itself with `margin: 0 auto`. Remember, however, if the image itself takes up 100% of the space it won't appear to center either. So it's a good idea to give it a width and use `margin: 0 auto` together.

```
img {
  width: 80%;
  display: block;
  margin: 0 auto;
}
```

Now the image should be centered. The zero in `margin: 0 auto`, refers to the top and bottom margin and the auto will automatically place an equal amount of space on the right and left to center it.

Let's center the paragraph container, that is inside main, in the same way. Give it a width of 80% and give it a `margin: 0 auto`.

```
main p {
  width: 80%;
  margin: 0 auto;
}
```

We didn't have to put `display: block` because `p` elements are already block-level elements.

If we want to also center the content of the container, we could add a `text-align: center` as well and the paragraph would center inside the container as well.

```
main p {
  width: 80%;
  margin: 0 auto;
  text-align: center;
}
```

I like it left aligned inside the container, so I won't add the `text-align: center`. But you can leave it if you like.

If I wanted to add some more margin to the top and bottom of my paragraph I can increase the zero and add a pixel size.

```
main p {
  width: 80%;
  margin: 50px auto;
  text-align: center;
}
```

Now we can see how to center containers and content of containers. And the elements inside our main section are centered nicely.

Let's take off the borders if you haven't already. Delete the following rule-set:

```
* {
  border: 2px solid blue;
}
```

So there we have a few different ways to center items on your page keeping in mind the difference between block and inline elements.