

WDD 130

[Home](#)[W1](#)[W2](#)[W3](#)[W4](#)[W5](#)[W6](#)[W7](#)[I-Learn](#)

CSS Layout: Position

Overview

Elements are positioned on a document using the [normal flow](#) by default. Sometimes, the layout calls for placing items by absolute or fixed positions on the document. This should be used sparingly and typically not for the entire document layout, only specific areas because of unintended outcomes.

"The position CSS property sets how an element is positioned in a document. The top, right, bottom, and left properties determine the final location of positioned elements." - [MDN](#)

Here is an example of using the **position** property and having text overlaying an image in the same container:



Figure 1: Example of Text Overlaying an Image in the Same Container

Prepare

- Reference: [CSS: Position](#) - MDN
Be sure to read the **absolute** and **fixed** sections under the [Values](#) section of the document.
- **Images:** Working with images can be challenging because they have a set width and height that must be maintained wherever it is used on the page or else the image will appear pixelated and/or distorted. Normally, a general rule of thumb is to set only the width or the height, not both in CSS in order to maintain the image's aspect ratio. If the image containers are set, then setting the width to

100% is a common practice to ensure the image fills the container. Here is an example of a CSS rule that sets the width to 100% and the height to auto:

```
img {  
  width: 100%;  
  height: auto;  
}
```

The **aspect ratio**, which is the width by height measurement is often shown as **2:1** where the width is twice the height.

In the example above from the Church's website, the intrinsic dimensions of the image may not fit the actual container dimensions by just using an auto height. In these cases, the object-fit property can be used to set the image to fill the container. Here is an example of the CSS rule:


```
img { /* be careful as this rule applies to all images */  
  width: 100%;  
  height: 100%;  
  object-fit: cover;  
}
```

Activity Instructions

Step 1: File Setup

1. Create a folder named "**week03**" in your wdd130 directory.
2. Create a folder named "**styles**" within the **week03** folder.
3. Create a file named "**overlay.html**" in the **week03** folder.
4. Add a CSS file named **overlay.css** in the **week03/styles** folder.

Step 2: HTML

1. Give the **overlay.html** a basic, valid html structure.
2. Be sure to **link** the CSS file in your HTML document **head**.
3. Copy the HTML from this  [CodePen - Position Absolute: Overlay Example](#) into the **overaly.html** **body**.

Step 3: CSS

1. Copy the CSS from the same CodePen into **overlay.css** file.
2. Move the **section class="stats"** display box from the upper right to the lower left of the image.

▼ Check Your Understanding

```
...  
bottom: 25px;  
left: 15px;  
...
```

3. Remove the **position: relative;** property from the container **div** (this div has a class of **herodiv** assigned to it).

What happens?

▼ Answer

The information **section** overlay moves to the bottom left of the document instead of the bottom left of the **div.herodiv** container. This is because of the principles of using position, the absolute positioned item is feeding off of its ancestors that HAVE a position property. None have that property now so it moves to the default document position. Still absolute but based upon the document now.

4. Also note the footer using a **position: fixed** declaration.

Submission

1. Test your page in your local browser using Live Server.
2. Commit and push your work to your wdd130 GitHub Pages repository.

<https://githubusername.github.io/wdd130/week03/overlay.html>

3. Share and discuss your work with your peers.

Copyright © Brigham Young University-Idaho | All rights reserved