

## EXERCISE THREE

# CSS LAYOUT & POSITIONING

In this exercise, you will finally be positioning your elements on the page and creating your layout. You will float certain elements and will move things out of the natural flow of the document by using position.

1. **Clear your margins and padding** at the beginning of your CSS document to remove the space around the body which is a browser default.

```
* {margin: 0; padding: 0;}
```

2. **Center the website** relative to the browser by putting everything in the body into a **div id** with the name **wrapper**. Set the width, and margin.

```
#wrapper {  
    width: 80%;  
    margin: auto;  
}
```

3. Include a **header**, **nav**, and **footer** and set their width, margins, and padding using percentages.

4. Add other **div** elements to further organize content and structure page. You may want to add more content to work with. I recommend creating columns for organization. When using columns add up their widths to equal 100% = (12 columns). For example:

css code:

```
.columns {float:left;}  
.col-1 {width: 8.33%;}  
.col-2 {width: 16.66%;}  
.col-3 {width: 25%;}  
.col-4 {width: 33.33%;}  
.col-5 {width: 41.66%;}  
.col-6 {width: 50%;}  
.col-7 {width: 58.33%;}  
.col-8 {width: 66.66%;}  
.col-9 {width: 75%;}
```

```
.col-10 {width: 83.33%;}  
.col-11 {width: 91.66%;}  
.col-12 {width: 100%;}
```

html code:

```
<div class="columns col-3">  
</div>  
<div class="columns col-9">  
</div>
```

4. Use **ids** to style unique elements: `<div id="container"></div>`

5. Use **classes** to style repeated elements with the same style:

```
<div class="columns"></div>
```

or multiple classes:

```
<div class="columns col-3"></div>
```

6. Use **css selectors** to apply formatting to various html elements that are named.

The **style requirements** include:

**display: inline** (navigation links for example)

**float: left or right** Position elements or columns using float. Make sure you indicate the width and height of your boxes, set margins, and use clear if you need to keep boxes from flowing next to a floating element.

**width** (only use %)

**height** (only use %)

**padding** (only use %)

**margin** (only use %)

**overflow** (set an element's css to overflow: auto to let the content dictate the height of the box)

**position: absolute or fixed** (Be sure to set the parent container to **relative**, then specify how far from the top, bottom, left, or right the box is positioned either from the browser or relative to its current position. Use positioning to remove elements from their normal flow.)

```
#container {
    width: 80%;
    margin: auto;
    position: relative;
}

nav {
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    background-color: green;
}
```

**z-index** (adjust the stacking order of the elements where you used the position property (*absolute*, *relative* or *fixed*). To push elements behind use a number lower than zero and to bring to the top use a number higher than zero.)