

Write your name below and indicate your role,

Project Manager (PM), Recorder (R)

Name \_\_\_\_\_ Role \_\_\_\_\_

Name \_\_\_\_\_ Role \_\_\_\_\_


## Positioning

### Your Tasks

- ☐ Indicate the default positioning of an element
- ☐ Apply the *position: relative* property
- ☐ Apply the *position: absolute* property
- ☐ Apply the *position: fixed* property
- ☐ Have Ms. Pluska check off the above tasks
- ☐ Apply the *z-index* property
- ☐ Apply the *display: inline-block* property
- ☐ Apply the *float* property
- ☐ Receive credit for the group portion of this lab


### ☐ Indicate the default positioning of an element

Consider the box elements shown below,

Output	
	
Index.html	Styles.css
<pre>&lt;div class = "gray"&gt;&lt;/div&gt; &lt;div class = "yellow"&gt;&lt;/div&gt; &lt;div class = "purple"&gt;&lt;/div&gt;</pre>	<pre>.gray {     background-color: gray;     height: 5em; } .yellow {     background-color: yellow;     height: 5em; } .purple {     background-color: purple;     height: 5em; }</pre>

Block-level elements like these boxes create a *block* the full width of their parent elements, and they prevent other elements from appearing in the same horizontal space.

Now consider the same box elements with a specified width,

Output	
	
Index.html	Styles.css
<pre> &lt;div class = "gray"&gt;&lt;/div&gt; &lt;div class = "yellow"&gt;&lt;/div&gt; &lt;div class = "purple"&gt;&lt;/div&gt; </pre>	<pre> .gray {     background-color: gray;     height: 5em;     width: 50%; } .yellow {     background-color: yellow;     height: 5em;     width: 50%; } .purple {     background-color: purple;     height: 5em;     width: 50%; } </pre>

In the example above the block-level elements also appear on the left side of the browser. This is the default position for block-level elements.

The default position of an element can be changed by setting its *position* property. The *position* property can take one of four values:

1. static- the default value (it does not need to be specified)
2. relative
3. absolute
4. fixed

Draw a picture of the output that would result when the css is applied to the html elements shown	
Index.html	Styles.css
<pre>&lt;div class = "one"&gt;One&lt;/div&gt; &lt;div class = "two"&gt;Two&lt;/div&gt; &lt;div class = "three"&gt;Three&lt;/div&gt;</pre>	<pre>.one {     width: 50%;     margin: 0 auto;     border: thin solid black; } .two {     position: static;     text-align: center;     width: 25%;     border: thin solid black; } .three {     text-align: right;     border: thin solid black; }</pre>

## □ Apply the *position: relative* property

One way to modify the default position of an element is by setting its position property to relative. This value allows you to position an element *relative* to its default static position on the web page.

```
.box-bottom {
    background-color: DeepSkyBlue;
    position: relative;
}
```

Although the code in the example above instructs the browser to expect a relative positioning of the div, it does not specify where the div should be positioned on the page.

In the example below, the `<div class="yellow">` will be moved down 20 pixels and to the right 50 pixels from its default static position. The image below displays the new position of the box. The dotted line represents where the statically positioned (default) box was positioned.


	
Index.html	Styles.css
<pre>&lt;div class = "gray"&gt;&lt;/div&gt; &lt;div class = "yellow"&gt;&lt;/div&gt;</pre>	<pre>.gray {   background-color: gray;   height: 5em; } .yellow {   background-color: yellow;   height: 5em;   position: relative;   top: 5em;   left: 10em; }</pre>

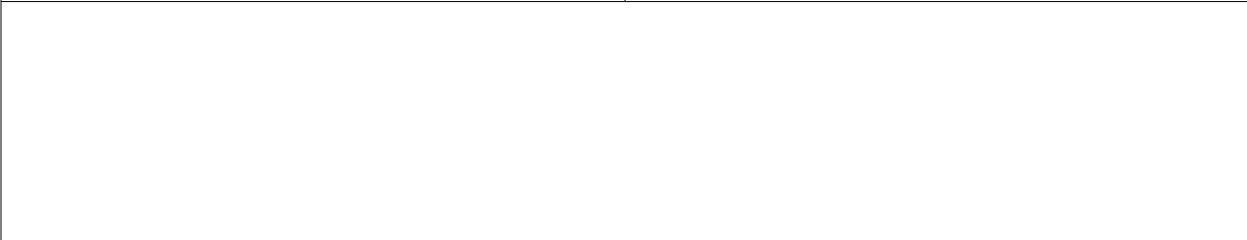
Units for offset properties can be specified in pixels, ems, or percentages.

Draw a picture of the output that would result when the css is applied to the html elements shown	
Index.html	Styles.css
<pre>&lt;div class = "one"&gt;One&lt;/div&gt; &lt;div class = "two"&gt;Two&lt;/div&gt;</pre>	<pre>.one {   width: 50%;   margin: 0 auto;   border: thin solid black; } .two {   position: relative;   text-align: center;   width: 25%;   left: 50%;   top: 10em;   border: thin solid black; }</pre>

## □ Apply the *position: absolute* property

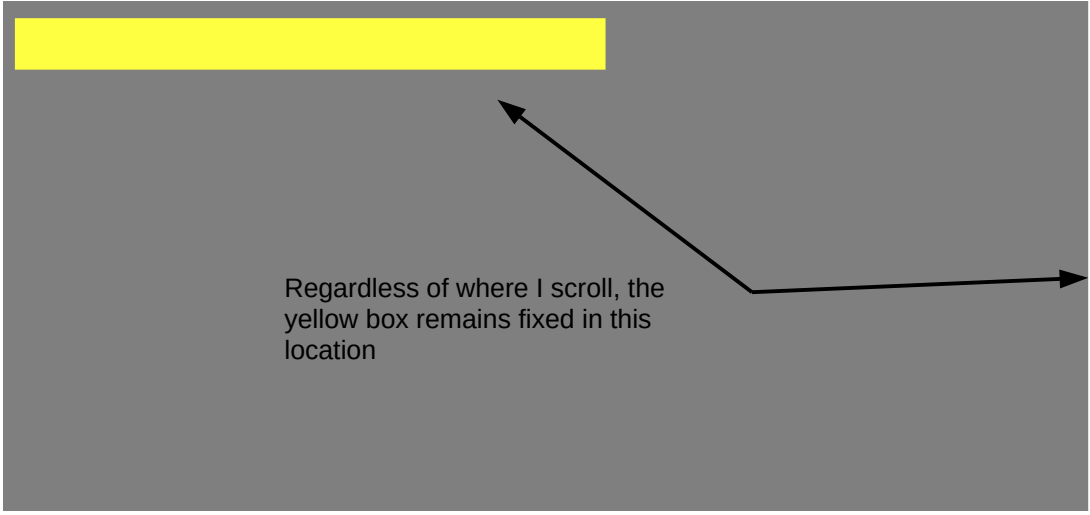
Another way of modifying the position of an element is by setting its position to *absolute*. When an element's position is set to *absolute* all other elements on the page will ignore the element and act like it is not present on the page.

Output	
	
Index.html	Styles.css
<pre>&lt;div class = "purple"&gt;&lt;/div&gt; &lt;div class = "gray"&gt;&lt;/div&gt; &lt;div class = "yellow"&gt;&lt;/div&gt;</pre>	<pre>.purple{     height: 5em;     background-color: purple; } .gray {     background-color: gray;     height: 5em; } .yellow {     background-color: yellow;     position: absolute;     top: 5em;     left: 10em;     height:5em;     width:50%; }</pre>

Draw a picture of the output that would result when the css is applied to the html elements shown	
Index.html	Styles.css
<pre>&lt;div class = "one"&gt;One&lt;/div&gt; &lt;div class = "two"&gt;Two&lt;/div&gt; &lt;div class = "three"&gt;Three&lt;/div&gt;</pre>	<pre>.one {     width: 50%;     margin: 0 auto;     border: thin solid black; } .two {     position: relative;     text-align: center;     width: 25%;     left: 50%;     border: thin solid black; } .three{     position: absolute;     top: 4em;     left: 25%;     height:5em;     width:50%;     border: thin solid black; }</pre>
	

## □ Apply the *position: fixed* property

When an element's position is set to *absolute*, as in the last exercise, the element will scroll with the rest of the document when a user scrolls. We can fix an element to a specific position on the page (regardless of user scrolling) by setting its position to *fixed*.

Output	
 <p>Regardless of where I scroll, the yellow box remains fixed in this location</p>	
Index.html	Styles.css
<pre>&lt;div class = "gray"&gt;&lt;/div&gt; &lt;div class = "yellow"&gt;&lt;/div&gt;</pre>	<pre>.gray{     height: 100em;     background-color: gray; } .yellow {     background-color: yellow;     position: fixed;     top: 5em;     left: 10em;     height:5em;     width:50%; }</pre>

Draw a picture of the output that would result when the css is applied to the html elements shown and indicate the behavior..

Index.html	Styles.css
<pre>&lt;div class = "one"&gt;One&lt;/div&gt; &lt;div class = "two"&gt;Two&lt;/div&gt;</pre>	<pre>.one {     position: fixed;     text-align: center;     width: 25%;     left: 50%;     border: thin solid black; } .two {     width: 50%;     margin: 0 auto;     border: thin solid black;     height:100em;     border: thin solid black; }</pre>

## □ Have Ms. Pluska check off the above tasks



Before you continue have Ms. Pluska check off the above tasks


Do not continue until you have Ms. Pluska's (or her designated TA's) signature \_\_\_\_\_

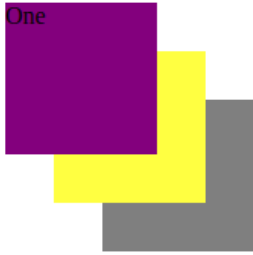
## □ Apply the *z-index* property

When boxes on a web page have a combination of different positions, the boxes (and therefore, their content) can overlap with each other, making the content difficult to read or consume.

The *z-index* property controls how far “back” or how far “forward” an element should appear on the web page when elements overlap. This can be thought of the *depth* of elements, with deeper elements appearing behind shallower elements.

In the example below, we set the *.box-top* position to relative and the *z-index* to 2. We changed position to relative, because the *z-index* property does not work on static elements. The *z-index* of 2 moves the *.box-top* element forward, because it is greater than the *.box-bottom z-index*, 1.

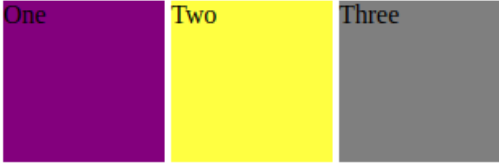
Output	
	
Index.html	Styles.css
<pre>&lt;div class = "box-top"&gt;&lt;/div&gt; &lt;div class = "box-bottom"&gt;&lt;/div&gt;</pre>	<pre>.box-top {   background-color: Aquamarine;   position: relative;   z-index: 2;   height:100px;   width:100px; } .box-bottom {   background-color: DeepSkyBlue;   position: absolute;   top: 20px;   left: 50px;   z-index: 1;   height:100px;   width:100px; }</pre>

Write css rules to style the boxes as shown	
<b>Output</b>	
	
<b>Index.html</b>	<b>Styles.css</b>
<pre>&lt;div class = "one"&gt;One&lt;/div&gt; &lt;div class = "two"&gt;Two&lt;/div&gt; &lt;div class = "three"&gt;Three&lt;/div&gt;</pre>	

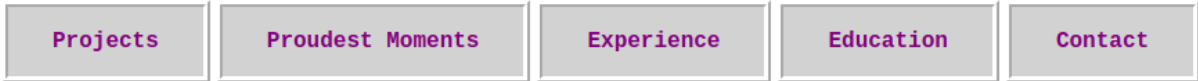
### □ Apply the *display: inline-block* property

The default display for html elements is vertically. The display: inline-block property forces elements to appear next to each other. Consider the following example,



Output	
	
Index.html	Styles.css
<pre>&lt;div class = "one"&gt;One&lt;/div&gt; &lt;div class = "two"&gt;Two&lt;/div&gt; &lt;div class = "three"&gt;Three&lt;/div&gt;</pre>	<pre>.one {   background-color: purple;   width:100px;   height:100px;   display: inline-block; } .two {   background-color: yellow;   width:100px;   height:100px;   display: inline-block; } .three {   background-color: gray;   width:100px;   height:100px;   display: inline-block; }</pre>

Write a css rule to style the navigation bar shown. Use the *display: inline-block* property to align the elements horizontally

Output	
	
Index.html	Styles.css
<pre>&lt;ul class="nav"&gt;   &lt;li&gt;Projects&lt;/li&gt;   &lt;li&gt;Proudest Moments&lt;/li&gt;   &lt;li&gt;Experience&lt;/li&gt;   &lt;li&gt;Education&lt;/li&gt;   &lt;li&gt;Contact&lt;/li&gt; &lt;/ul&gt;</pre>	

## □ Apply the *float* property

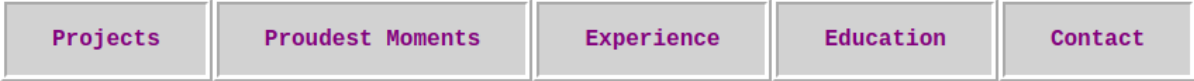
So far, you've learned how to specify the exact position of an element using offset properties. If you're simply interested in moving an element as far left or as far right as possible on the page, you can use the *float* property.

The *float* property can be set to one of two values:

1. *left* - this value will move, or float, elements as far left as possible.
2. *right* - this value will move elements as far right as possible.

Floated elements must have a width specified, as in the example above. Otherwise, the element will assume the full width of its containing element, and changing the float value will not yield any visible results.

	
Index.html	Styles.css
<pre>&lt;div class = "one"&gt;One&lt;/div&gt; &lt;div class = "two"&gt;Two&lt;/div&gt; &lt;div class = "three"&gt;Three&lt;/div&gt;</pre>	<pre>.one {   background-color: purple;   width:100px;   height:100px;   float: right; } .two {   background-color: yellow;   width:100px;   height:100px;   float: right; } .three {   background-color: gray;   width:100px;   height:100px;   float: right; }</pre>

Write a css rule to style the navigation bar shown. Use the <i>float</i> property to align the elements horizontally. Use the <i>list-style-type</i> property to remove the default bullets	
<b>Output</b>	
	
<b>Index.html</b>	<b>Styles.css</b>
<pre>&lt;ul class="nav"&gt;   &lt;li&gt;Projects&lt;/li&gt;   &lt;li&gt;Proudest Moments&lt;/li&gt;   &lt;li&gt;Experience&lt;/li&gt;   &lt;li&gt;Education&lt;/li&gt;   &lt;li&gt;Contact&lt;/li&gt; &lt;/ul&gt;</pre>	

## ❑ Receive Credit for the group portion of this lab



- Indicate the names of all group members.
- Have Ms. Pluska check your Box Model tasks
- Submit your lab to the needs to be graded folder to receive credit for the group portion of this lab.
- Do not submit your lab until you have Ms. Pluska's (or her designated TA's) signature

\_\_\_\_\_