Exercise 2: Styling Web Pages with CSS

1 Introduction

Web pages are plain text documents formatted with HTML. Different HTML elements -- or tags -- format plain text to change their appearance. For instance, heading tags format their text to be much larger than their surrounding plain text. Paragraph tags add vertical spacing. List tags enumerate or bullet the line items they surround. Table tags can format data into rows and columns. Other than basic appearance and layout, HTML does not offer much else to make a Web page more appealing or distinguished from other Web pages.

To make Web pages stand out, we must use ***CSS*** (***Cascading Style Sheets***). CSS is a computer language that can describe the appearance of an existing HTML document. CSS can refer to different parts of a document and configure various appearance attributes such as foreground and background color, font, alignment, spacing, borders, paddings, etc. This assignment will give us a chance to learn about how to use CSS to style Web pages.

2 Learning objectives

- Styling Web content with Cascading Style Sheets (CSS)
- Laying out content with Flex
- Animating content with keyframes

3 Exercises

This section presents several CSS exercises to practice styling HTML documents. After you work through the exercises you will apply the skills to create ***Tuiter*** on your own. Create directory ***exercises/e2*** where you will practice several CSS exercises. Under the ***exercises/e2*** directory, create an HTML file called ***index.html***.

3.1 Styling a document's look and feel

3.1.1 Styling elements with the ***style*** attribute

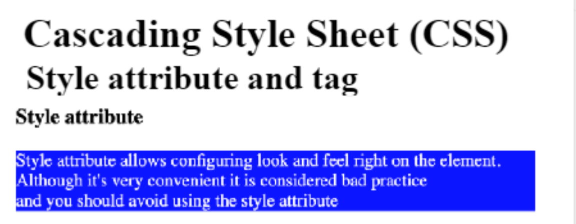
An HTML element's ***style*** attribute can configure the look and feel of the element by changing the values of its style properties as shown below

```
<element style="property1: value1; property2: value2">
  element body
</element>
```

The value of the style attribute is a string formatted with a language called ***CSS*** or ***Cascading Style Sheets*** and is the focus of this assignment. To practice using the ***style*** attribute, copy and paste the example below into ***exercises/e2/index.html***

index.html

```
<h1>Cascading Style Sheet (CSS)</h1>
<h2>Style attribute and tag</h2>
<h3>Style attribute</h3>
<p style="background-color: blue; color:
white">
  Style attribute allows configuring look
  and feel right on the element. Although
it's
  very convenient it is considered bad
practice
  and you should avoid using the style
attribute
</p>
```



Cascading Style Sheet (CSS)

Style attribute and tag

Style attribute

Style attribute allows configuring look and feel right on the element.
Although it's very convenient it is considered bad practice
and you should avoid using the style attribute

In the exercise above we styled the paragraph element with its ***style*** attribute. We changed the color of its background by setting the ***background-color*** property to ***blue*** and also changing the foreground color to white by setting the ***color*** property to ***white***. There are 100s of style attributes of which we'll cover the most relevant.

3.1.2 Styling documents with the ***style*** tag

A slightly better way to configure the look and feel of a Web page is to declare ***CSS rules*** in the ***<style>*** tag in the ***<header>*** and then refer to the tag by their ***selector***, e.g., name, ID, or class. It's better because you can control the style of a whole page from a central place instead of dealing with individual elements. To practice using the ***style*** tag, copy and paste the highlighted style tag below at the end of the ***head*** tag as shown.

```
<style>
   selector1 {
       property1:
value1;
       property2:
value2;
   }
</style>
```

index.html

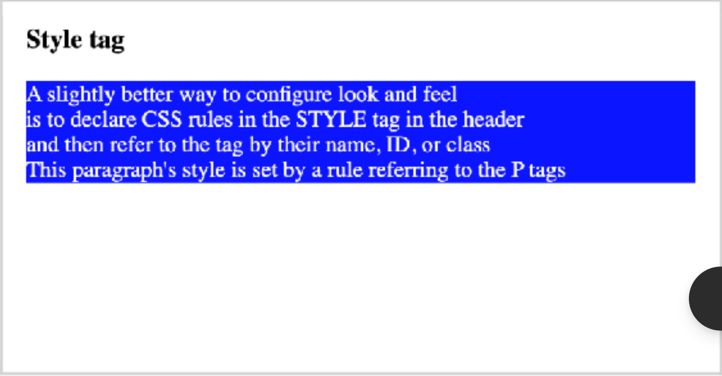
```
<head>
   <style>
       p {
           background-color:
blue;
           color: white;
       }
   </style>
</head>
```

```
<!-- new style tag
     selector "p" refers to all paragraphs
     sets all paragraph background color to
blue
     sets all paragraph foreground color to
white
-->
```

Then copy the following paragraph at the end of the ***index.html*** document. The paragraph should render with a blue background and white text as shown.

index.html

```
<h3>Style tag</h3>
<p>
A slightly better way to configure look and
feel
is to declare CSS rules in the STYLE tag in the
header and then refer to the tag by their
name,
ID, or class This paragraph's style is set by
a
rule referring to the P tags
</p>
```



Style tag

A slightly better way to configure look and feel
is to declare CSS rules in the STYLE tag in the header
and then refer to the tag by their name, ID, or class
This paragraph's style is set by a rule referring to the P tags

3.1.3 Linking CSS styling documents with the *link* tag

Although the style tag is slightly better than the style attributes, styling an entire website will entail editing many style tags in their HTML documents. Instead of changing styles inside many HTML documents, it is a ***best practice*** to do all styling configuration in separate CSS files and then link the files from the HTML document with the ***link*** tag. To practice using the ***link*** tag create a brand-new file called ***index.css*** in the same directory of the ***index.html*** document, and copy the following content.

index.css

```
p {
    background-color:          /*This is the same content currently in the
blue;                          style tag.*/
    color: white;              /*We'll use this file from now on for this
}                              assignment*/
```

Then, as shown below, comment out the style tag in ***index.html*** since we won't be using it anymore -- ever. Instead, use the ***link*** tag to load the ***index.css*** file created earlier.

index.html

```
<head>
    <!--style>                              <!--
        p {                                  Comment out, or remove, the style
            background-color:               tag.
blue;                                        We moved this content to the
            color: white;                   index.css file
        }                                   -->
    </style-->
    <link href="index.css"
          rel="stylesheet"/>                <!-- Instead load the index.css
</head>                                          file created earlier -->
```

Finally copy the following content to the end of ***index.html*** which should render a blue and white paragraph as shown.

index.html

```html
<h3>Link tag</h3>
<p>
   The best way to apply CSS rules is to declare
   them in a separate CSS file and load it with
   the LINK tag. Always use this method.
</p>
```

Link tag

The best way to apply CSS rules is
to declare them in a separate CSS file
and load it with the LINK tag. Always use this method.

3.2 Selecting content to style with selectors

3.2.1 Selecting content with *ID* selectors

The CSS rules in previous exercises styled all paragraphs at once by using the name of the tag ***p*** and then specifying the style property values. Instead of changing the look and feel of all the elements of the same name, e.g., ***p***, we can refer to a specific element by their ID, an attribute specifying a unique identifier. To practice using ID selectors, in ***index.css***, comment out the highlighted paragraph CSS rule as shown and add the two CSS rules referring to paragraphs with IDs ***id-selector-1*** and ***id-selector-2***.

index.css

```css
/*p {
  background-color:
blue;
  color: white;}*/
p#id-selector-1 {
  background-color:
red;
  color: white;
}
p#id-selector-2 {
  background-color:
yellow;
  color: black;
}
```

index.html

```html
<h3>ID selectors</h3>
<p id="id-selector-1">
  Instead of changing the
look and
  feel of all the
elements of the
  same name, e.g., P, we
can refer
  to a specific element
by its ID
</p>
<p id="id-selector-2">
  Here's another
paragraph using a
  different ID and a
different look
  and feel
</p>
```

Browser

ID selectors

Instead of changing the look and feel of all the elements of the same name, e.g., P, we can refer to a specific element by its ID

Here's another paragraph using a different ID and a different look and feel

3.2.2 Selecting content with *class* selectors

Instead of using IDs to refer to specific elements, you can use an element's ***class*** attribute instead, or a combination of both. Class selectors can be used just like ID selectors if you keep them unique, but can also be used to apply the same style to several elements, even if they are different types of elements. We will be using class selectors exclusively from now on. To practice using class selectors, copy the CSS rule below into ***index.css***, and the HTML at the end of ***index.html***.

index.css	***index.html***	**Browser**
`.class-selector { background-color: yellow; color: blue; }`	`<h3>Class selectors</h3> <p class="class-selector"> Instead of using IDs to refer to elements, you can use an element's CLASS attribute</p> <h4 class="class-selector"> This heading has same style as paragraph above</h4>`	**Class selectors** Instead of using IDs to refer to elements, you can use an element's CLASS attribute **This heading has same style as paragraph above**

The example above declares a selector that declares a style that transforms the background and foreground color. We can then use the selector to apply the transformation to several elements. The above example applies the style to two elements, the paragraph and the heading.

3.2.3 Selecting content based on the document structure

Selectors can be combined to refer to elements in particular places in the document. A set of selectors separated by a space can refer to elements in a hierarchy. For instance: ***.selector1 .selector2 { ... }*** refers to an element whose class is ***.selector2*** and is inside some ***descendant*** of another element whose class is ***.selector1***. If we use ">" instead to separate the classes, then we can establish a direct parent/child relationship. To practice selecting elements using a set of selectors, copy the following content in ***index.html***

index.html	
`<div class="selector-1"> <h3>Document structure selectors</h3> <div class="selector-2"> Selectors can be combined to refer elements in particular places in the document <p class="selector-3"> This paragraph's red background is referenced as <br/>.selector-2 .selector3<br/> meaning the descendant of some ancestor. <br/> <span class="selector-4">Whereas this span is a direct child of its parent</span> <br/> You can combine these relationships to create specific styles depending on the document structure`	`<!-- this is parent element with selector .selector-1 .selector-2 is a direct child of .selector-1 .selector-3 is a descendant of .selector-1 and a direct child of .selector-2 this is a descendant of .selector-1 and .selector-2 and a direct child of .selector-3 -->`

```
  </p>
 </div>
</div>
```

Let's now style elements .selector-3 and .selector-4. Copy the CSS below into ***index.css***.

index.css

```
.selector-1 .selector-3 {              /* refers to .selector-3 as a descendant of
   background-color: red;              .selector-1   */
   color: white;
}
.selector-2 > .selector-3 >
.selector-4 {
   background-color: yellow;
   color: blue;                        /* refers to .selector-4 as a direct child of
}                                      .selector-3 */
                                       /* which is a direct child of .selector-
                                       2                  */
```

3.3 Styling color

3.3.1 Styling the foreground color

Foreground colors can be configured using the CSS ***color*** property as follows

```
.some-css-selector {  /* selects some DOM element    */
   color: blue;       /* sets color property to blue */
}
```

Colors can be defined as follows

- As strings, e.g., white, red, blue, etc
- As hexadecimals, e.g., #ABCDEF
- As RGB, e.g., rgb(12, 34, 56)

Here are a couple of examples:

`.the-sun { color: rgb(255,255,0); }`	`.the-sky { color: blue; }`	`.ketchup { color: #FF0000; }`
🟨	🟦	🟥

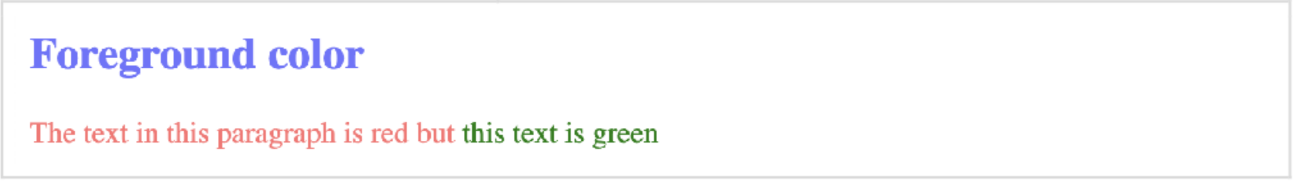
To practice working with foreground colors, copy the CSS rules below into ***index.css*** to declare several useful color classes. Copy the HTML code below into ***index.html***. Confirm it renders as shown.

index.css

```
.fg-color-black { color: black;
}
.fg-color-white { color: white;
}
.fg-color-blue { color:
#7070ff; }
.fg-color-red { color: #ff7070;
}
.fg-color-green { color: green;
}
```

index.html

```
<h2>Colors</h2>
<h3 class="fg-color-blue">Foreground color</h3>
<p class="fg-color-red">
  The text in this paragraph is red but
  <span class="fg-color-green">this text is
green</span>
</p>
```



```
Foreground color

The text in this paragraph is red but this text is green
```

3.3.2 Styling the background color

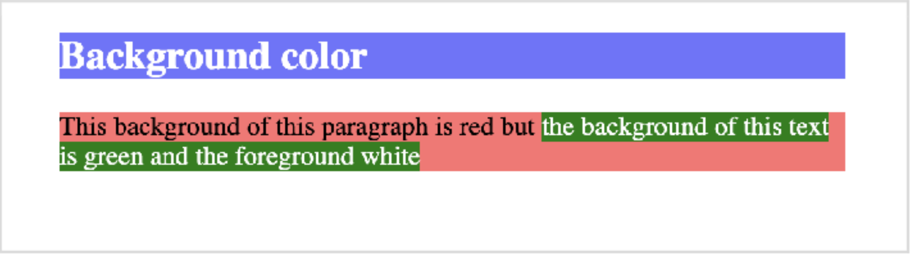
To practice working with background colors, copy the CSS rules shown below into ***index.css*** and the HTML code into ***index.html***. Confirm the browser renders as shown.

index.css

```
.bg-color-yellow {
   background-color:
#ffff07;
}
.bg-color-blue {
   background-color:
#7070ff;
}
.bg-color-red {
   background-color:
#ff7070;
}
.bg-color-green {
   background-color:
green;
}
```

index.html

```
<h3 class="bg-color-blue fg-color-white">Background
color</h3>
<p class="bg-color-red fg-color-black">
   This background of this paragraph is red but
   <span class="bg-color-green fg-color-white"> the
background
  of this text is green and the foreground white</span>
</p>
```



```
Background color

This background of this paragraph is red but the background of this text
is green and the foreground white
```

3.4 Styling the box model

3.4.1 Styling borders

Use CSS border properties to configure the look and feel of the border around content.
Here's a sample of the properties that can be configured.

`.some-selector {`	
`  border-width: 10px;`	`/* configure border with several properties*/`
`  border-style: solid|dotted|dashed|double;`	`/* border's width. Can also provide per border*/`
`  border-color: red │ blue ...;`	`/* the style of the border*/`
`}`	`/* the color of the border      */`

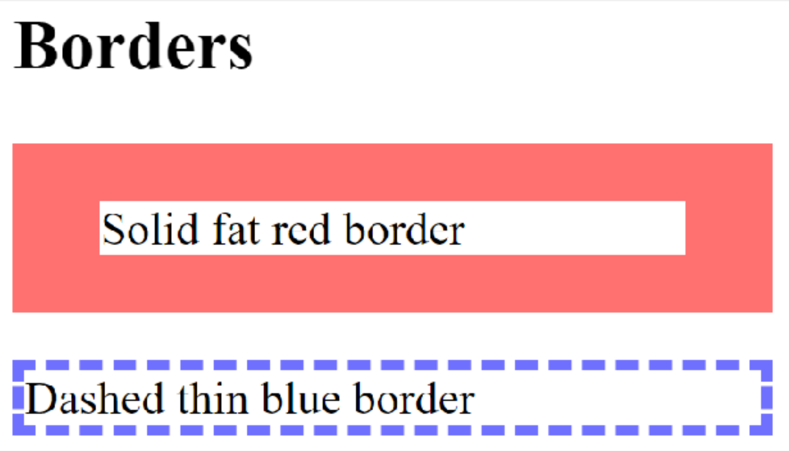
To practice styling borders, copy the CSS code below into ***index.css*** and the HTML code into ***index.html*** and confirm the browser renders as shown.

index.css

```
.border-fat {
   border-width: 20px
30px 20px 30px; }
.border-thin {
   border-width: 4px; }
.border-solid {
   border-style: solid; }
.border-dashed {
   border-style: dashed;
}
.border-yellow {
   border-color: #ffff07;
}
.border-red {
   border-color: #ff7070;
}
.border-blue {
   border-color: #7070ff;
}
```

index.html

```
<h2>Borders</h2>
<p class="border-fat border-red border-solid">
   Solid fat red border</p>
<p class="border-thin border-blue border-dashed">
   Dashed thin blue border</p>
```



3.4.2 Stying margins and paddings

You can also configure the spacing between elements. To practice with paddings, copy the CSS code below into ***index.css*** and the HTML into ***index.html***. Confirm browser renders as shown.

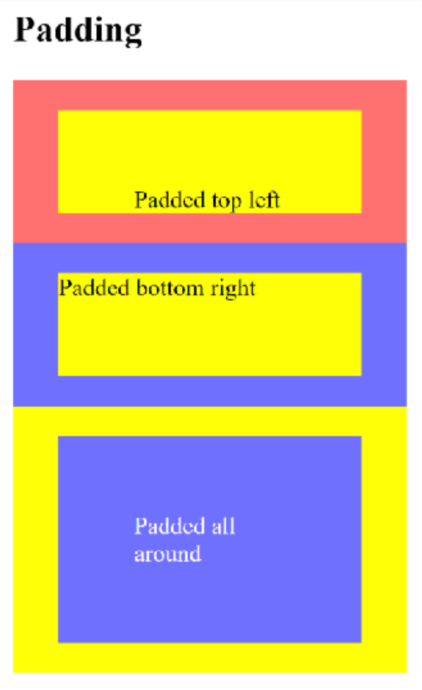
index.css

```
.padded-top-left
{
  padding-top:
50px;
  padding-left:
50px;
}
.padded-bottom-
right {
  padding-bottom:
50px;
  padding-right:
50px;
}
.padding-fat {
  padding: 50px;
}
```

index.html

```
<h2>Padding</h2>
<div class="padded-top-left
border-fat
            border-red border-
solid
            bg-color-yellow">
   Padded top left
</div>

<div class="padded-bottom-right
border-fat
            border-blue border-
solid
            bg-color-yellow">
   Padded bottom right
</div>
<div class="padding-fat border-
fat
            border-yellow
border-solid
```

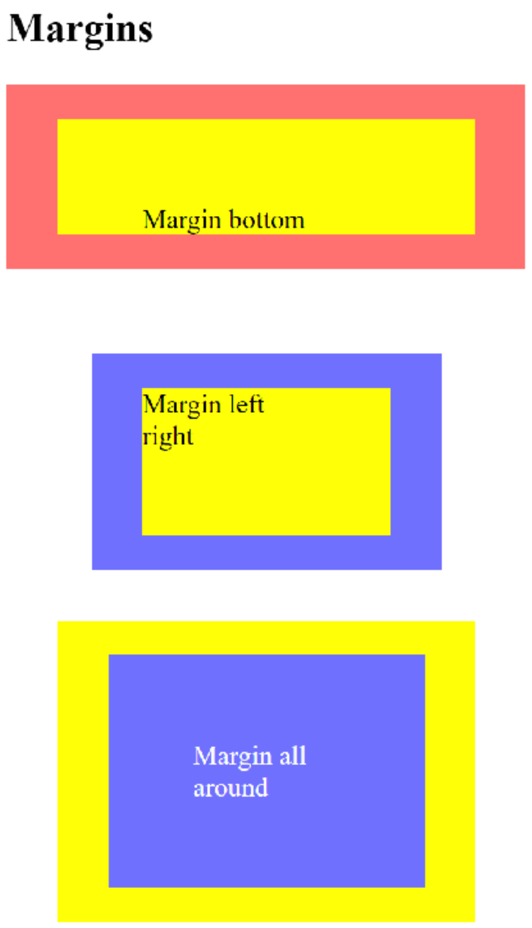
Browser



	`            bg-color-blue fg-`
	`color-white">`
	`   Padded all around`
	`</div>`	

To practice with margins, copy the CSS code below into ***index.css*** and the HTML into ***index.html***. Confirm browser renders as shown.

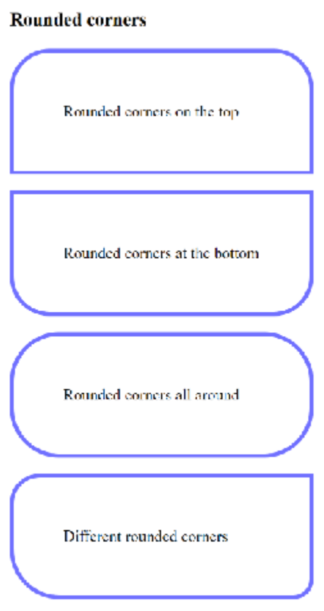
index.css	***index.html***	***Browser***

`.margin-bottom`
`{`
`  margin-`
`bottom: 50px;`
`}`
 	
`.margin-right-`
`left {`
`  margin-left:`
`50px;`
`  margin-`
`right: 50px;`
`}`
 	
`.margin-all-`
`around {`
`  margin:`
`30px;`
`}`
	`<h2>Margins</h2>`
	`<div class="margin-bottom`
	`            padded-top-`
	`left`
	`            border-fat`
	`border-red`
	`            border-solid`
	`            bg-color-`
	`yellow">`
	`   Margin bottom</div>`
	`<div class="margin-right-`
	`left`
	`            padded-bottom-`
	`right`
	`            border-fat`
	`border-blue`
	`            border-solid`
	`            bg-color-`
	`yellow">`
	`   Margin left right`
	`</div>`
	`<div class="margin-all-`
	`around`
	`            padding-fat`
	`border-fat`
	`            border-yellow`
	`            border-solid`
	`            bg-color-blue`
	`            fg-color-`
	`white">`
	`   Margin all around`
	`</div>`
		 **Margins**
		Margin bottom
		Margin left right
		Margin all around

3.4.3 Stying corners

You can configure the corners of element borders to be rounded. Either all of them at once or specific corners. You can do this by configuring a border's radius. To practice rounding some corners, copy the CSS and HTML below into ***index.css*** and ***index.html*** and confirm the browser renders as shown.

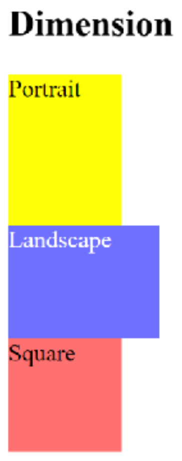
index.css	*index.html*	*Browser*

`.rounded-corners-bottom {`
`  border-bottom-left-radius: 40px;`
`  border-bottom-right-radius: 40px;`
`}`
 	
`.rounded-corners-all-around {`
`  border-radius: 50px;`
`}`
 	
`.rounded-corners-inline {`
`  border-radius: 30px 0px 20px 50px;`
`}`
	`<h3>Rounded corners</h3>`
	`<p class="rounded-corners-top border-thin border-blue border-solid padding-fat">`
	`  Rounded corners on the top`
	`</p>`
	`<p class="rounded-corners-bottom border-thin border-blue border-solid padding-fat">`
	`  Rounded corners at the bottom`
	`</p>`
	`<p class="rounded-corners-all-around border-thin border-blue border-solid padding-fat">`
	`  Rounded corners all around`
	`</p>`
	`<p class="rounded-corners-inline border-thin border-blue border-solid padding-fat">`
	`  Different rounded corners`
	`</p>`
		 **Rounded corners**
		Rounded corners on the top
		Rounded corners at the bottom
		Rounded corners all around
		Different rounded corners

3.5 Styling dimensions

You can configure an element's dimensions with ***width*** and ***height*** properties. To practice setting element's dimensions, copy the CSS and HTML below into ***index.css*** and ***index.html*** and confirm the browser renders as shown.

index.css	*index.html*	*Browser*

`.dimension-portrait {`
`    width: 75px;`
`    height: 100px;`
`}`
`.dimension-landscape {`
`    width: 100px;`
`    height: 75px;`
`}`
`.dimension-square {`
`    width: 75px;`
`    height: 75px;`
`}`
	`<h2>Dimension</h2>`
	`<div>`
	`   <div class="dimension-portrait bg-color-yellow">`
	`       Portrait`
	`   </div>`
	`   <div class="dimension-landscape bg-color-blue fg-color-white">`
	`       Landscape`
	`   </div>`
	`   <div class="dimension-square bg-color-red">`
	`       Square`
	`   </div>`
	`</div>`
		 **Dimension**
		Portrait
		Landscape
		Square

3.6 Styling position

You can configure an element's position with the ***position*** property. The property has many possible values, but we'll explore ***relative***, ***absolute***, and ***static***.

3.6.1 Styling relative position

Setting ***position*** property to ***relative*** allows moving the element relative to its original position. To practice setting element's relative position, copy the CSS and HTML below into ***index.css*** and ***index.html*** and confirm the browser renders as shown.

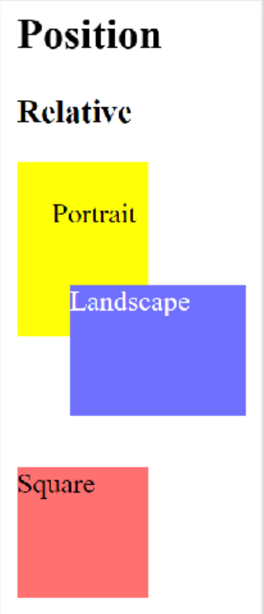
index.css

```
.pos-relative-nudge-up-
right {
   position: relative;
   bottom: 30px;
   left: 30px;
}
.pos-relative-nudge-
down-right {
   position: relative;
   top: 20px;
   left: 20px;
}
.pos-relative {
   position: relative;
}
```

index.html

```
<h2>Position</h2>
<h3>Relative</h3>
<div>
 <div class="bg-color-yellow
             dimension-portrait">
  <div class="pos-relative-nudge-
down-right">
       Portrait</div>
 </div>
 <div class="pos-relative-nudge-up-
right
             bg-color-blue fg-color-
white
             dimension-landscape">
       Landscape</div>
 <div class="bg-color-red dimension-
square">
       Square</div>
</div>
```

Browser



3.6.2 Styling absolute position

Setting ***position*** property to ***absolute*** allows moving the element relative to the position of its parent. To practice setting element's absolute position, copy the CSS and HTML below into ***index.css*** and ***index.html***. Notice several ***<br/>*** elements were added at the end of the example to make room for the next exercise.

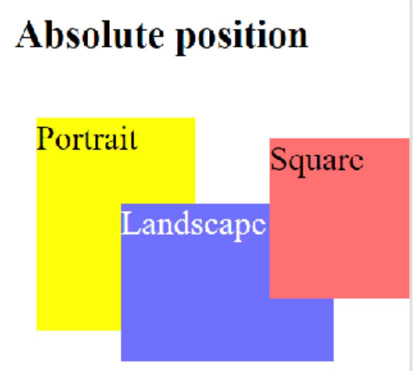
index.css

```
.pos-
absolute-10-
10 {
   position:
absolute;
   top:
10px;
   left:
10px;
}
.pos-
absolute-50-
50 {
   position:
absolute;
   top:
50px;
   left:
50px;
}
```

index.html

```
<h3>Absolute position</h3>
<div class="pos-relative">
   <div class="pos-absolute-10-10
    bg-color-yellow dimension-
portrait">
       Portrait
   </div>
   <div class="pos-absolute-50-50
        bg-color-blue fg-color-white
        dimension-landscape">
       Landscape   </div>
   <div class="pos-absolute-120-20
        bg-color-red dimension-
square">
       Square      </div>
</div><br/><br/><br/><br/><br/><br/><b
r/>
```

Browser

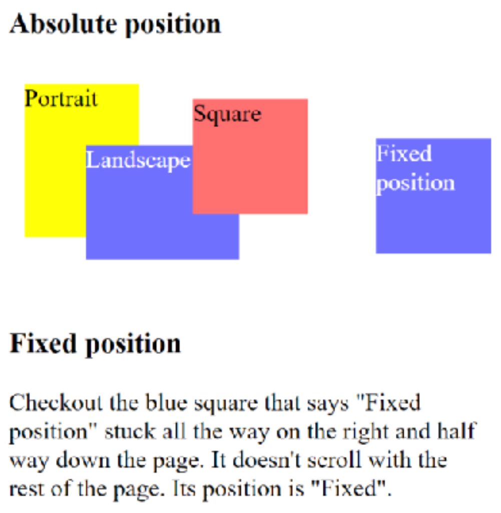


`.pos-absolute-120-20 { position: absolute; top: 20px; left: 120px; }`		

3.6.3 Styling fixed position

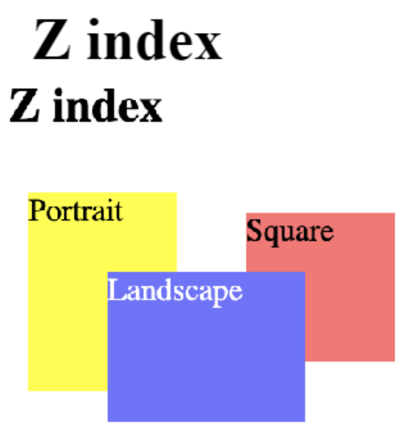
Setting ***position*** property to ***fixed*** allows setting the element relative to the window. That means that if you scroll the content of the page, the element will not scroll with it. To practice setting element's fixed position, copy the CSS and HTML below into ***index.css*** and ***index.html*** and confirm the browser renders as shown. Your display may be different depending on the actual size of the screen and scrolling.

index.css	***index.html***	***Browser***

`.pos-fixed { position: fixed; right: 0px; bottom: 50%; }`	`<h3>Fixed position</h3>`
	
	`Checkout the blue square that says "Fixed position" stuck all the way on the right and half way down the page. It doesn't scroll with the rest of the page. Its position is "Fixed".`
	
	`<div class="pos-fixed dimension-square bg-color-blue fg-color-white">`
	`Fixed position`
	`</div>`
		 **Absolute position** Portrait Square Landscape Fixed position **Fixed position** Checkout the blue square that says "Fixed position" stuck all the way on the right and half way down the page. It doesn't scroll with the rest of the page. Its position is "Fixed".

3.6.4 Styling z-index

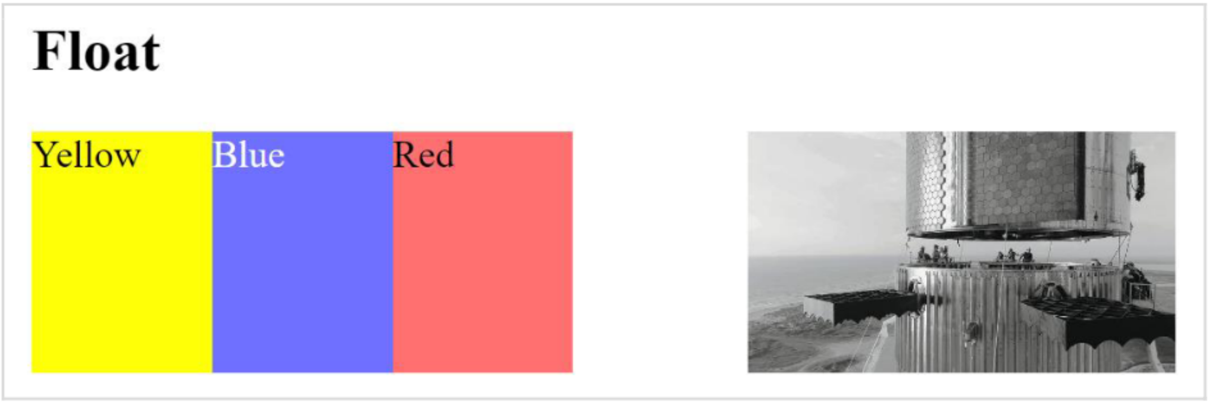
When the browser renders content declared in HTML documents, it calculates positions and dimensions so every element has a dedicated rectangle on the window. Typically, elements don't fall on top of each other. When you start moving elements with ***position***, then overlapping elements are possible. By default, elements are rendered in the order declared in HTML documents. Elements declared later render on top of elements declared earlier. The ***z-index*** CSS property overrides this behavior. Default value of ***z-index*** is ***auto***, which corresponds to 0. Increasing z-index can make elements render later, or on top of, others. To practice setting an element's ***z-index***, copy the CSS and HTML below into ***index.css*** and ***index.html***

index.css	*index.html*	*Browser*
`.zindex-bring-to-front { z-index: 10; }`	`<h2>Z index</h2> <div class="pos-relative"> <div class="pos-absolute-10-10 bg-color-yellow dimension-portrait"> Portrait</div> <div class="zindex-bring-to-front pos-absolute-50-50 bg-color-blue fg-color-white dimension-landscape"> Landscape</div> <div class="pos-absolute-120-20 bg-color-red dimension-square"> Square</div> </div><br/><br/><br/><br/><br/><br/><br/>`	

3.6.5 Floating content

HTML does not support laying out content horizontally. The CSS float property allows fixing that. To practice laying out content horizontally, copy the CSS and HTML below into ***index.css*** and ***index.html*** and confirm the browser renders as shown.

index.css	*index.html*
`.float-left { float: left; } .float-right { float: right; height: 100px; } .float-done { clear: both; }`	`<h2>Float</h2> <div> <div class="float-left dimension-portrait bg-color-yellow"> Yellow</div> <div class="float-left dimension-portrait bg-color-blue fg-color-white"> Blue</div> <div class="float-left dimension-portrait bg-color-red"> Red</div> <img class="float-right" src="https://www.staradvertiser.com/wp-content/uploads/2021/08/web1_Starship-gap2.jpg"/> <div class="float-done"></div> </div>`



3.6.6 Laying out content in a grid

Using float, we can implement a grid layout made up of rows and columns. To practice laying out content in a grid, copy the CSS and HTML below into ***index.css*** and ***index.html*** and confirm the browser renders as shown.

index.css

```css
.grid-row {
   clear: both;
}

.grid-col-half-page {
   width: 50%;
   float: left;
}
```

index.html

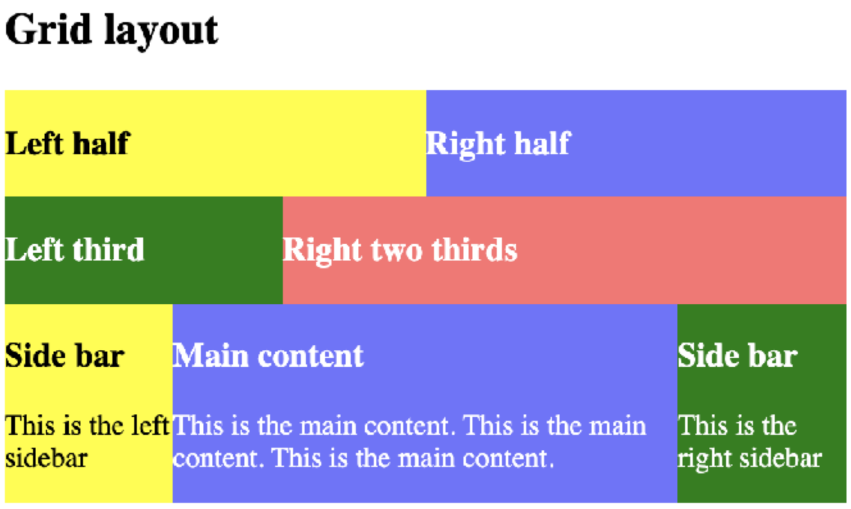
```html
<h2>Grid layout</h2>
<div class="grid-row">
  <div class="grid-col-half-page bg-color-yellow">
    <h3>Left half</h3>
  </div>
  <div class="grid-col-half-page bg-color-blue
              fg-color-white">
    <h3>Right half</h3>
  </div>
</div>
```

```css
.grid-col-third-page {
   width: 33%;
   float: left;
}

.grid-col-two-thirds-page
{
   width: 67%;
   float: left;
}

.grid-col-left-sidebar {
   width: 20%;
   float: left;
}

.grid-col-main-content {
   width: 60%;
   float: left;
}

.grid-col-right-sidebar {
```

```html
<div class="grid-row">
  <div class="grid-col-third-page bg-color-green
              fg-color-white">
    <h3>Left third</h3>
  </div>
  <div class="grid-col-two-thirds-page bg-color-red
              fg-color-white">
    <h3>Right two thirds</h3>
  </div>
</div>
<div class="grid-row">
  <div class="grid-col-left-sidebar bg-color-
yellow">
    <h3>Side bar</h3>
    <p>This is the left sidebar</p>
  </div>
  <div class="grid-col-main-content bg-color-blue
              fg-color-white">
    <h3>Main content</h3>
    <p>This is the main content. This is the main
content.
       This is the main content. </p>
  </div>
```

```
    width: 20%;
    float: left;
}
```

```
    <div class="grid-col-right-sidebar bg-color-green
                fg-color-white">
      <h3>Side bar</h3>
      <p>This is the right sidebar</p>
    </div>
</div>
```



Grid layout

Left half

Right half

Left third

Right two thirds

Side bar

This is the left sidebar

Main content

This is the main content. This is the main content. This is the main content.

Side bar

This is the right sidebar