Subject Code:	CS 3	Computer Science 3
Module Code:	5.0	Introduction to Cascading Style Sheets
Lesson Code:	5.6	Float, Position, width, height, top, bottom, left, right
Time Frame:		30 minutes

TARGET

Time Allocation: 1 min.

After completing this module, you are expected to:

- float multiple elements;
- float elements in a container;
- use floats to create columns.

HOOK

Time Allocation: 1 mins.

The previous learning module explains the basics of floating property. Let us look at more examples and explore additional floating behaviors. With CSS, it is possible to float multiple elements, float all the elements in a container element to create a multicolumn layout, as we will see in this module.

IGNITE

Time Allocation: 17 mins.

Floating multiple elements

It is perfectly fine to float multiple elements on a page or even within a single element. In fact, it is one way to turn a list of links into a horizontal menu, as we will see in a moment.

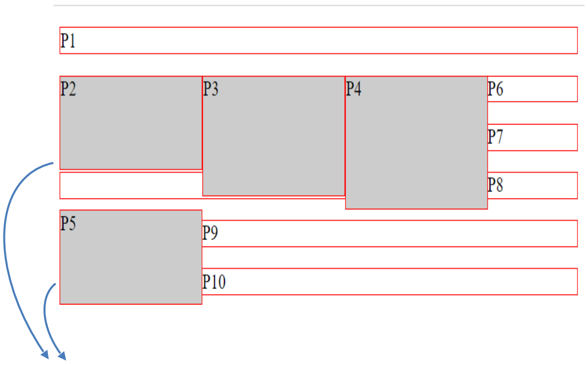
When you float multiple elements, there is a complex system of behind-the-scenes rendering rules that ensures floated elements do not overlap. You can consult the CSS specification *(https://www.w3.org/Style/CSS/specs.en.html)* for the details, but the upshot of it is that floated elements will be placed as far left or right (as specified) and as high up as space allows.

The figure below shows what happens when a series of sequential paragraphs are floated to the same side. The first three floats start stacking up from the left edge, *but when there isn't enough room for the fourth, it moves down and to the left until it bumps into something*—in this case, the edge of the browser window. In any case, in the event that one of the floats, for example, "P2," had been exceptionally long, it would have bumped the edge of the long floats.

The Source:

```
<p>P1</p>
<p class="float2">P2</p>
<p class="float3">P3</p>
<p class="float4">P4</p>
<p class="float5">P5</p>
<p>P6</p>
<p>P7</p>
<p>P8</p>
<p>P9</p>
<p>P10</p>
```

Result:



Elements floated to the same side line up. If there is not enough room, subsequent elements move down.

That is the underlying behavior, but let us apply it to something more practical, like a navigation menu. It makes sense semantically to mark-up navigation as an unordered list, as shown here. We have omitted real URL values in the ***a*** elements to simply the markup.

The style sheets:

```
p.float2 {
                float: left;
                width: 150px;
                height: 70px;
                margin: 0px;
                background:
     #CCC;
                }
p {border: 1px solid red;}

p.float3 {
                float: left;
                width: 150px;
                height: 90px;
                margin: 0px;
                background:
     #CCC;
                }
p{border: 1px solid red;}

p.float4 {
                float: left;
                width: 150px;
                height:
     100px;
                margin: 0px;
                background:
     #CCC;
               }
p{border: 1px solid red;}

p.float5 {
                float: left;
                width: 150px;
                height: 70px;
                margin: 0px;
                background:
     #CCC;
```

```
<ul>
     <li><a href = "#">Home</a></li>
     <li><a href = "#">News</a></li>
     <li><a href = "#">Contact</a></li>
     <li><a href = "#">About</a></li>
</ul>
```

There are various approaches to converting it to a horizontal bar but the primary steps in our floating example are as follows.

1. Turn off the bullets, and set padding and margins to zero.

```
ul {
      list-style-type: none;
      margin: 0;
      padding: 0;
   }
```

2. Float each list item to the left so they line up, taking advantage of the multiple float behavior described earlier.

```
 li {
     float: left;
    }
```

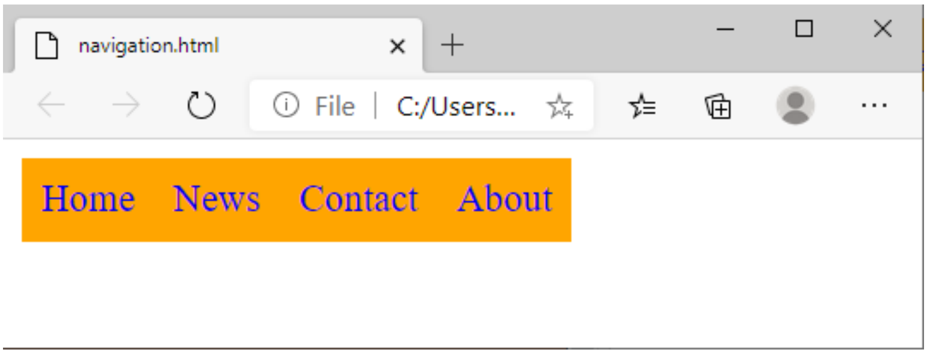
3. Make the anchor elements in the list items (**a**) display as block elements so you can set the dimensions, padding, margins, and other visual styles. You could set the styles for the other link states as well (such as **a:hover**), but we'll keep this example short.

```
li a {
       display: block;
       padding: 8px;
       background-color: orange;
      text-decoration: none;
            }
```

Note: The *li a* is a descendant selector in CSS that matches all <a> elements that are inside a <li> element.

4. Clear the element that comes after the menu in the document so it starts below the menu.

This will produce the following result:

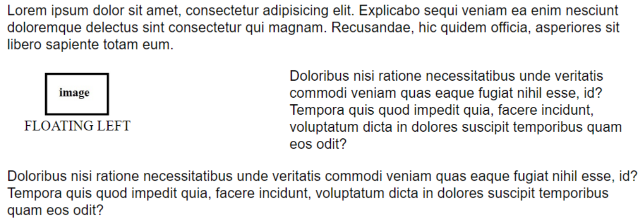


Home News Contact About

Using floats to create columns

So far, we have floated small parts of a page, but as mentioned earlier, you can also float whole sections of the page to create columns. In fact, that is the way the pros do it! There are a couple of solutions, and they mostly come down to a matter of preference.

Float one div and add a wide margin on the side of the text element that wraps around it.



Lorem ipsum dolor sit amet, consectetur adipisicing elit. Explicabo sequi veniam ea enim nesciunt doloremque delectus sint consectetur qui magnam. Recusandae, hic quidem officia, asperiores sit libero sapiente totam eum.

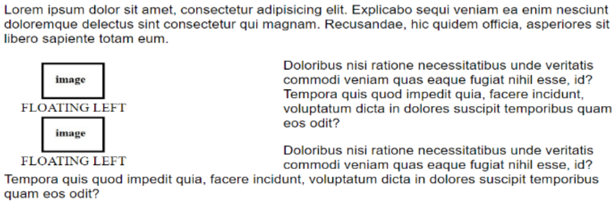
image

FLOATING LEFT

Doloribus nisi ratione necessitatibus unde veritatis commodi veniam quas eaque fugiat nihil esse, id? Tempora quis quod impedit quia, facere incidunt, voluptatum dicta in dolores suscipit temporibus quam eos odit?

Doloribus nisi ratione necessitatibus unde veritatis commodi veniam quas eaque fugiat nihil esse, id? Tempora quis quod impedit quia, facere incidunt, voluptatum dicta in dolores suscipit temporibus quam eos odit?

Float both divs to the left or right.

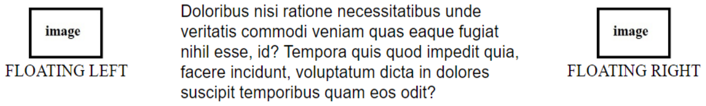


Lorem ipsum dolor sit amet, consectetur adipisicing elit. Explicabo sequi veniam ea enim nesciunt doloremque delectus sint consectetur qui magnam. Recusandae, hic quidem officia, asperiores sit libero sapiente totam eum.

Doloribus nisi ratione necessitatibus unde veritatis commodi veniam quas eaque fugiat nihil esse, id? Tempora quis quod impedit quia, facere incidunt, voluptatum dicta in dolores suscipit temporibus quam eos odit?

Doloribus nisi ratione necessitatibus unde veritatis commodi veniam quas eaque fugiat nihil esse, id? Tempora quis quod impedit quia, facere incidunt, voluptatum dicta in dolores suscipit temporibus quam eos odit?

Float one div to the left and the second div to the right (or vice versa).



Doloribus nisi ratione necessitatibus unde veritatis commodi veniam quas eaque fugiat nihil esse, id? Tempora quis quod impedit quia, facere incidunt, voluptatum dicta in dolores suscipit temporibus quam eos odit?

Let us take a look at a simple example: a two-column webpage with the content area on the left, navigation on the right, and a header and footer area to cap it off.

The source:

```
<div id="container">
      <div id="header">
<h1>Header</h1>
</div>
            <div id="content">
                                    <p> Lorem ipsum dolor sit amet,
                  consectetur adipiscing elit. Sed auctor placerat
                  metus, sit amet egestas dui rutrum sed. Aliquam eget
                  fringilla metus, vel pulvinar turpis. Mauris
                  consectetur purus convallis nibh consectetur, nec
                  mattis nisi euismod.</p>

                                    <p> Lorem ipsum dolor sit amet,
                  consectetur adipiscing elit. Sed auctor placerat
                  metus, sit amet egestas dui rutrum sed. Aliquam eget
                  fringilla metus, vel pulvinar turpis. Mauris
                  consectetur purus convallis nibh consectetur, nec
                  mattis nisi euismod.</p>
            </div>
      <div id="navigation">
            <a href="#">Nav Item</a>
            <a href="#">Nav Item</a>
            <a href="#">Nav Item</a>
            <a href="#">Nav Item</a>
      </div>
            <div id="footer">Footer</div>
```

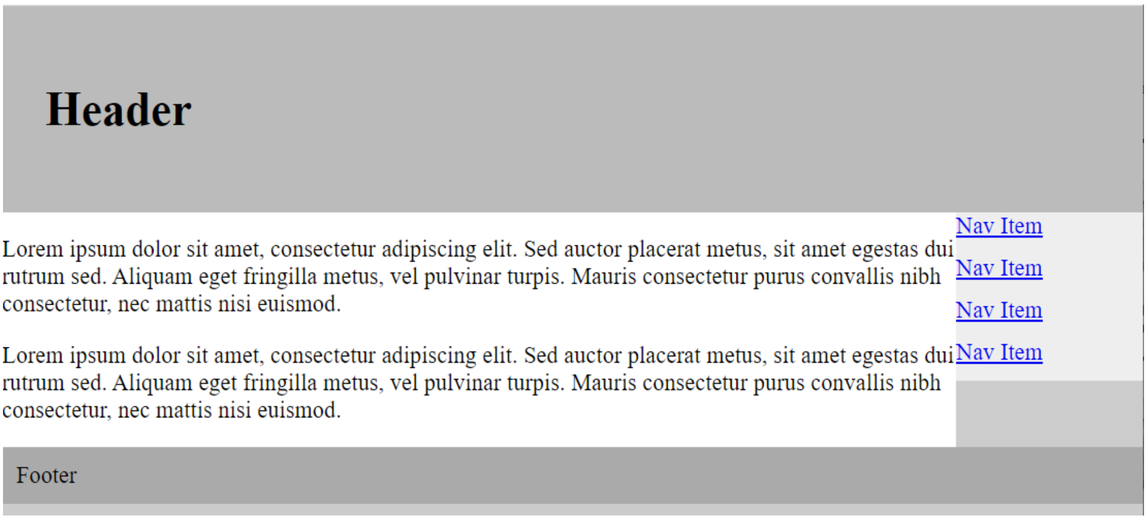
The style sheets:

```
<style>
      body {
            margin: 0;
            padding: 0;
            background: #ccc;
                     }
      #container {
                  width: 960px;
                  margin: 0 auto;
                    }
      #header {
            padding: 30px;
            background: #bbb;
                   }

      #content {
                  float: left;
                  width: 660px;
                  background: #fff;
                  }

      #navigation {
                  float: right;
                  width: 300px;
                  background: #eee;
            }
      #navigation a {
                  display: block;
                  margin: 0 0 10px 0;
            }
      #footer {
                  clear: both;
                  background: #aaa;
                  padding: 10px;
           }
</style>
```

This will produce the following result:



Header

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed auctor placerat metus, sit amet egestas dui rutrum sed. Aliquam eget fringilla metus, vel pulvinar turpis. Mauris consectetur purus convallis nibh consectetur, nec mattis nisi euismod.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed auctor placerat metus, sit amet egestas dui rutrum sed. Aliquam eget fringilla metus, vel pulvinar turpis. Mauris consectetur purus convallis nibh consectetur, nec mattis nisi euismod.

Nav Item
Nav Item
Nav Item
Nav Item

Footer

Three-column floats work basically the same way, there’s just more calculating to do.

Regardless of which method works best for your content or suits your fancy, there are a few things you need to keep in mind. First, every float needs to have a specified width. Thereafter, you need to be very careful that you have calculated the widths of each column correctly, factoring in padding, borders, and margins. If the total width of all the columns exceeds the available width of the browser or other containing element, you'll get what is known as "float drop". That is, the final floated column will run out of room and get bumped down below the column next to it.

The limitation to using floats for columns is that it is dependent on the order of the elements in the source document. The floated element must appear *before* the content that wraps around it, and your source may not always be ordered conveniently.

box-sizing Property

The **box-sizing** property defines how the width and height of an element are calculated: should they include padding and borders, or not.

The box property has worked like as shown below:

width + padding + border = actual visible/rendered width of an element's box
height + padding + border = actual visible/rendered height of an element's box

Means when you set the height and width, it appears a little bit bigger than the given size, when element border and padding are added to the element.

Let us take a look at a simple example:

The source:

```
     <p>
        <div id="example1">This div has a width of 300px. But the full
  width
            is 300px + 20px (left and right border) + 60px (left and
    right padding) = 380px!</div></p>
```

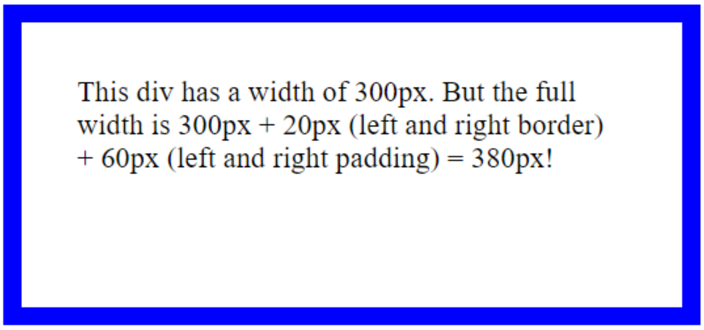
The style sheet:

```
<style>
#example1 {
      box-sizing: content-box;
      width: 300px;
      height: 100px;
      padding: 30px;
      border: 10px solid blue;
}
#example2 {
      box-sizing: border-box;
      width: 300px;
      height: 100px;
       padding: 30px;
      border: 10px solid blue;
}
</style>
```

This will produce the following result:

box-sizing: content-box (default):

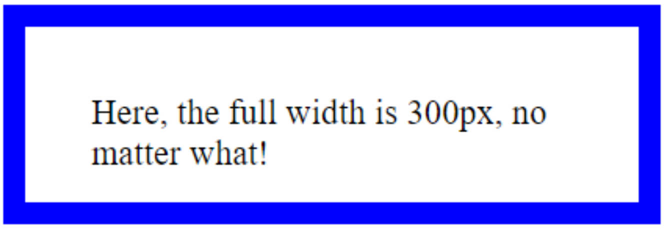
Width and height only apply to the content of the element:



This div has a width of 300px. But the full width is 300px + 20px (left and right border) + 60px (left and right padding) = 380px!

box-sizing: border-box:

Width and height apply to all parts of the element: content, padding and borders:



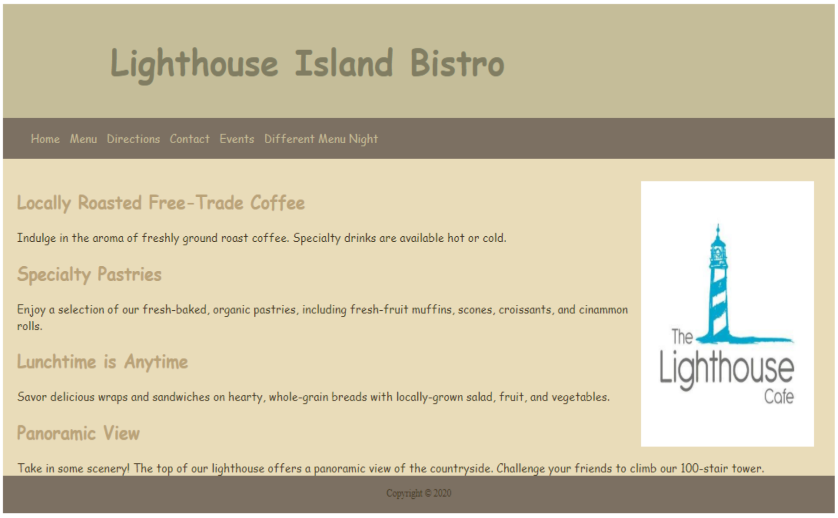
Here, the full width is 300px, no matter what!

NAVIGATE

Time Allocation: 10 mins.

It is your time to apply what you have learned.

Create a web page containing several layout components, including a header and navigational field similar to the image shown below:



Lighthouse Island Bistro

Locally Roasted Free-Trade Coffee

Indulge in the aroma of freshly ground roast coffee. Specialty drinks are available hot or cold.

Specialty Pastries

Enjoy a selection of our fresh-baked, organic pastries, including fresh-fruit muffins, scones, croissants, and cinammon rolls.

Lunchtime is Anytime

Savor delicious wraps and sandwiches on hearty, whole-grain breads with locally-grown salad, fruit, and vegetables.

Panoramic View

Take in some scenery! The top of our lighthouse offers a panoramic view of the countryside. Challenge your friends to climb our 100-stair tower.

Copyright © 2020

Modify the given html source code to use document-level stylesheets so that it is formatted nicely. The source code already includes a text content along with the Lighthouse Island Bistro image file where you can download from (*https://pbs.twimg.com/profile_images/446333085471088640/4ph_sh1D.jpeg*), so you don't need to type it in.

The Source:

```
<!doctype html>
<html>
<head>
<meta charset = "utf-8">
      <title>Lighthouse Island Bistro</title>
</head>
<body>
      <div id="wrapper">
            <div id="header">
                    <h1>Lighthouse&nbsp;Island&nbsp;Bistro</h1>
            </div>
            <div id="nav">
                    <ul>
                        <li><a href="index.html">Home</a></li>
                        <li><a href="rooms.html">Menu</a></li>
                        <li><a href="directions.html">Directions</a></li>
                        <li><a href="contact.html">Contact</a></li>
                        <li><a href="contact.html">Events</a></li>
                        <li><a href="contact.html">Different Menu Night </
                        a></li>
                   </ul>
            </div>
            <div class="content">
                        <img src="Bistro.jpeg" width="250" height="355"
                        id="floatright" alt="Lighthouse Island">
                        <h2>Locally Roasted Free-Trade Coffee</h2>
                        <p>Indulge  in  the  aroma  of  freshly  ground  roast
                        coffee. Specialty drinks are       available hot or
                        cold.</p>
                        <h2>Specialty Pastries</h2>
                        <p>Enjoy  a  selection  of  our  fresh-baked,  organic
                        pastries, including
                        fresh-fruit  muffins,  scones,  croissants,  and
                        cinammon rolls.</p>
                        <h2>Lunchtime is Anytime</h2>
                        <p>Savor delicious wraps and sandwiches on hearty,
                        whole-grain breads with locally-grown salad, fruit,
                        and vegetables. </p>
                        <h2>Panoramic View</h2>
                        <p>Take in some scenery! The top of our lighthouse
                        offers  a  panoramic  view  of      the  countryside.
                        Challenge  your  friends  to  climb  our  100-stair
                        tower.</p>
            </div>
            <div id="footer">Copyright &copy; 2020</div>
      </div>
</body>
</html>
```

Your page doesn't need to be exactly the same as shown. If your measurements are different or your margins are a bit off or your fonts are of different size, you will still receive full credit. But the overall placement of items should be the same. Make sure you have the following features:

- The body element selector: background (#453823) and Nova Slim, cursive, or the default sans-serif font face

- The wrapper id: centered, width set to 80% of the browser viewport; minimum width of 850px display text in a color (#453823) and display a background color (#7C7062) and margin is auto.
- The rightcolumn id: margin left is 155px; display a background color (#453823) and display text in a color (#000000).
- The header id: background color (#C5BD99); text color (#817D62); font size is 150% and padding 10px 10px 10px 155px and Nova Slim, cursive or the default sans-serif font face.
- The h2 element selector: display text in a color (#BAA378) and Nova Slim, cursive, or the default sans-serif font face.
- The class content: Configure a padding 20px 20px 0 20px; and display a background color (E9DCB9).
- The floatright id: 10px margin; float right.
- The footer id: 80% font size; centered text; clear right; 20 pixels of padding on the bottom.
- The #nav ul selector: eliminate the list markers.
- The #nav li selector: display inline; Nova Slim, cursive, or the default san-serif font face.
- The #nav a selector: no underline; 10 pixels padding right and display text color (#C5BD99).
- Save your file as **Section_FullName_Float_Layout.html** and send your final solution to ______________________.

Note: This is a non-graded activity. But you can still submit your output to check whether you have learned something from this module.

KNOT

Time Allocation: 1 min.

In summary,

- Floating is an extremely powerful technique that you will likely end up using quite a bit - it quickly becomes essential when you want to do more detailed layouts. However, it does come with a small price of added complexity. We have already shown you a couple of the "extraordinary" behaviors you will run into when using floats and there are more of them out there.
- The limitation to using floats for columns is that it is dependent on the order of the elements in the source document. The floated element must appear *before* the content that wraps around it, and your source may not always be ordered conveniently.
- Box sizing property is used to change the height and width of an element.

REFERENCES

Terry Felke-Morris(2015). Page Layout. In Terry Felke-Morris (Seventh Edition), Web Development and Design Foundations with HTML5 (267-312). England, Pearson

Jennifer Niederst Robbins (2012). Floating and Positioning. In Jennifer Niederst Robbins (4th Edition), Learning Web Design A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics (341-371). Canada, O'Reilly Media, Inc.

Noah Stokes (2011, March 08). CSS Float 101. Retrieve from https://alistapart.com/article/css-floats-101/

Patrick Young. HTML & Cascading Style Sheets. (n.d.) Retrieve from https://emunix.emich.edu/~evett/E-Commerce/assign1/HTML_CSS.html

Build Horizontal Navigation Bar with Floating List Items in CS3 (n.d.). Retrieve from https://www.tutorialspoint.com/build-horizontal-navigation-bar-with-floating-list-items-in-css

CSS3-Box Sizing (n.d.). Retrieve from https://www.tutorialspoint.com/css/css3_box_sizing.htm

Prepared by:	Reviewed by:
LEO R. MACALALAD SST-III PSHS-CALABARZON	ALINE TERESA MENDOZA SST V PSHS- Main