

LAB 5

ADVANCED CSS: LAYOUT

What You Will Learn

- How to position and float elements outside of normal page flow
- How to construct multi-column layouts
- How to make a responsive layout
- How to use a CSS Framework to simplify page layout

Approximate Time

The exercises in this lab should take approximately 60 minutes to complete.

Fundamentals of Web Development

Randy Connolly and Ricardo Hoar

Textbook by Pearson
<http://www.funwebdev.com>

CREATING TABLES

PREPARING DIRECTORIES

- 1 If you haven't done so already, create a folder in your personal drive for all the labs for this book.
- 2 From the main labs folder (either downloaded from the textbook's web site using the code provided with the textbook or in a common location provided by your instructor), copy the folder titled lab05 to your course folder created in step one.

The `<table>` element in HTML represents information that exists in a two-dimensional grid. Tables can be used to display calendars, financial data, pricing tables, and many other types of data. Just like a real-world table, an HTML table can contain any type of data: not just numbers, but text, images, forms, even other tables.

Exercise 5.1 — DOCUMENT FLOW

- 1 Open, examine, and test lab05-exercise01.html.
- 2 Resize the browser window and experiment by making the width smaller and larger.
- 3 Modify the following style in lab05-exercise01.css and test.

```
h1 span {
  color: #8B0000;
  font-size: 1.5em;
  display: block;
}
```

This changes the `` element in the main heading from an inline element to a block-level element.

- 4 Add the following style and test.

```
li {
  display: inline;
}
```

This changes the `` elements from block-level to inline (they will thus show up on a single line now).

- 5 Add the following style and test.

```
div img {
  display: block;
}
```

POSITIONING

Exercise 5.2 — RELATIVE POSITIONING

- 1 Open, examine, and test lab05-exercise02.html.
- 2 Modify the following style in lab05-exercise02.css and test.

```
figure {  
  background-color: #EDEDDE;  
  border: 1pt solid #A8A8A8;  
  padding: 5px;  
  width: 150px;  
  top: 150px;  
  left: 200px;  
}
```

- 3 Modify the following style and test.

```
figure {  
  background-color: #EDEDDE;  
  border: 1pt solid #A8A8A8;  
  padding: 5px;  
  width: 150px;  
  top: 150px;  
  left: 200px;  
  position: relative;  
}
```

As you can see in Figure 5.1, the original space for the positioned <figure> element is preserved, as is the rest of the document's flow. As a consequence, the repositioned element now overlaps other content.

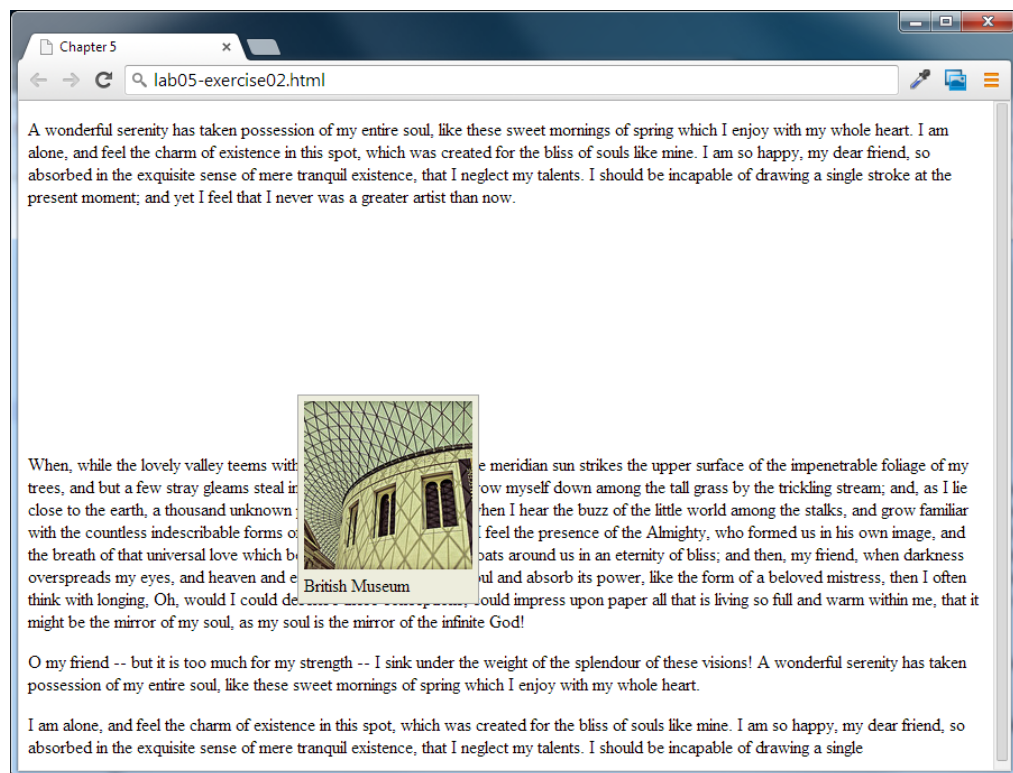


Figure 5.1 – Exercise 5.2 complete

Exercise 5.3 — ABSOLUTE POSITIONING

- 1 Open, examine, and test lab05-exercise03.html.
- 2 Modify the following style in lab05-exercise03.css and test.

```
figure {
    background-color: #EDEDDE;
    border: 1pt solid #A8A8A8;
    padding: 5px;
    width: 150px;
    top: 150px;
    left: 200px;
    position: absolute;
}
```

With absolute positioning, space is not left for the moved element, as it is no longer in the normal flow.

- 3 Modify the following style and test.

```
figcaption {
    background-color: #EDEDDE;
    padding: 5px;
    top: 150px;
```

```
    left: 200px;
    position: absolute;
}
```

A moved element via absolute position is actually positioned relative to its nearest positioned ancestor container (that is, a block-level element whose position is fixed, relative, or absolute). In this example, the `<figcaption>` is absolutely positioned; it is moved 150 px down and 200 px to the left of its nearest positioned ancestor, which happens to be its parent (the `<figure>` element).

Exercise 5.4 — STACKING USING Z-INDEX

- 1 Open, examine, and test `lab05-exercise04.html`.

- 2 Modify the following style and test.

```
figure {
    background-color: #EDEDDE;
    border: 1pt solid #A8A8A8;
    padding: 5px;
    width: 150px;
    top: 150px;
    left: 200px;
    position: absolute;
}

figcaption {
    font-size: 1.25em;
    background-color: yellow;
    padding: 5px;
    top: 90px;
    left: 140px;
    position: absolute;
}
```

Notice that the caption now overlaps both the underlying paragraph text and the figure image.

- 3 Modify the following styles (some CSS omitted here for clarity) and test.

```
figure {
    ...
    position: absolute;
    z-index: 5;
}

figcaption {
    ...
    position: absolute;
    z-index: 1;
}
```

Note that this did not move the `<figure>` on top of the `<figcaption>` as one might expect. This is due to the nesting of the caption within the figure.

- 4 Modify the following styles and test.

```
figure {
  ...
  position: absolute;
  z-index: 1;
}

figcaption {
  ...
  position: absolute;
  z-index: -1;
}
```

The <figure> zindex could be any value equal to or above 0 as long as the <figcaption> is below 0.

- 5 Modify the following styles and test.

```
figure {
  ...
  position: absolute;
  z-index: -1;
}

figcaption {
  ...
  position: absolute;
  z-index: 1;
}
```

If the <figure> zindex is given a value less than 0, then any of its positioned descendants change as well. Thus both the <figure> and <figcaption> move underneath the body text.

FLOATING

Exercise 5.5 — FLOATING ELEMENTS

- 1 Open, examine, and test lab05-exercise05.html.

- 2 Modify the following styles and test.

```
figure {
  border: 1pt solid #A8A8A8;
  background-color: #EDEDDE;
  padding: 5px;
  margin: 10px;
  width: 150px;
  float: left;
}
```

Notice that a floated block-level element must also have a width specified.

- 3 Modify the following styles and test.

```
figure {
  border: 1pt solid #A8A8A8;
  background-color: #EDEDDE;
  padding: 5px;
  margin: 10px;
  width: 150px;
  float: right;
}
```

Exercise 5.6 — FLOATING IN A CONTAINER

- 1 Open, examine, and test lab05-exercise06.html.

Notice the empty margin space on the left.

- 2 Modify the following styles and test.

```
figure {
  border: 1pt solid #262626;
  background-color: #c1c1c1;
  padding: 5px;
  width: 150px;
  margin: 10px;
  float: left;
}
```

Notice that a floated item moves to the left or right of its container. In this case the container for the floated <figure> is the <article> element.

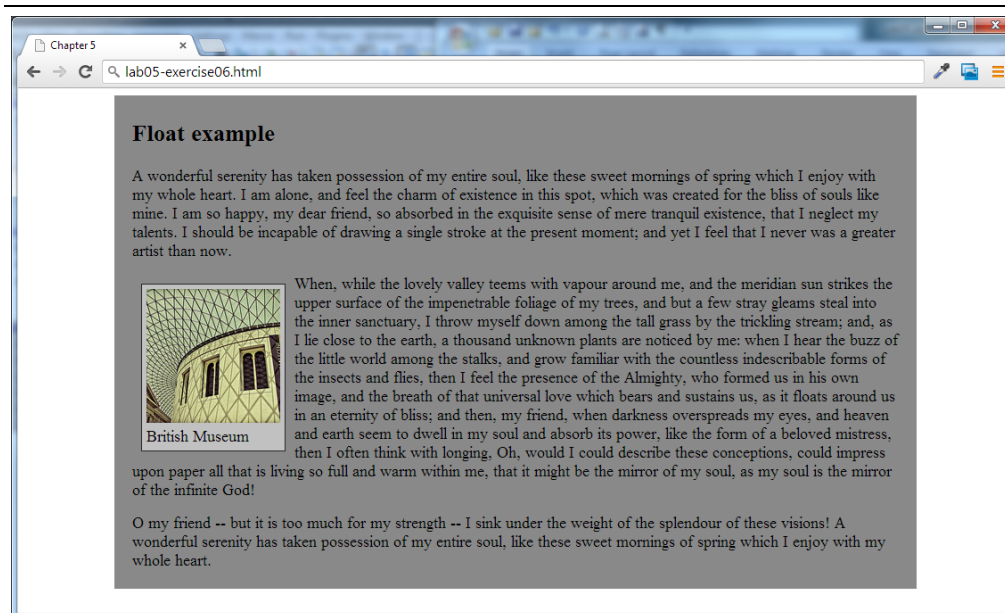


Figure 5.2 – Exercise 5.6 complete

Exercise 5.7 — FLOATING MULTIPLE ITEMS

- 1 Open, examine, and test lab05-exercise07.html.
Notice that each <figure> is a block-level element and hence on its own line.

- 2 Modify the following style and test.

```
figure {
  border: 1pt solid #262626;
  background-color: #c1c1c1;
  padding: 5px;
  width: 150px;
  margin: 10px;
  float: left;
}
```

When you float multiple items that are in proximity, each floated item in the container will be nestled up beside the previously floated item. All other content in the containing block (including other floated elements) will flow around all the floated elements.

- 3 Experiment by changing the width of the browser window.
This can result in some pretty messy layouts as the browser window increases or decreases in size (that is, as the containing block resizes). The solution is to use the clear property (next exercise).

Exercise 5.8 — FLOATING AND CLEARING

- 1 Open, examine, and test lab05-exercise08.html.
- 2 Add the following style and test.

```
.first { clear: left; }
```

- 3 Modify the third <figure> element as follows and test:

```
<figure class="first">
  
  <figcaption>British Museum</figcaption>
</figure>
```

- 4 Modify the <p> element after the figures as follows and test:

```
<p class="first">When, while the lovely valley ...
```

Exercise 5.9 — USING POSITIONING

- 1 Open, examine, and test lab05-exercise09.html.
In this exercise, we are going to position the caption on top of the image.
- 2 Modify the following style and test.


```
figcaption {
  width: 100px;
  background-color: black;
  color: white;
  opacity: 0.6;
  width: 140px;
  height: 20px;
  padding: 5px;
  position: absolute;
  top: 130px;
  left: 10px;
}
```

Since absolute positioning is positioning in relation to the closest positioned ancestor, we will also need to position the `<figure>`.

- 3 Modify the following style and test.

```
figure {
  border: 1pt solid #262626;
  background-color: #c1c1c1;
  padding: 10px;
  width: 200px;
  margin: 10px;
  position: relative;
}
```

Notice that we are not moving the figure. Instead by setting its position to relative, we are creating a positioning context for the absolute positioning in the previous step.

- 4 Add the following markup and test.

```
<figure>
  
  <figcaption>British Museum</figcaption>
  
</figure>
```

- 5 Add the following style and test.

```
.overlaid {
  position: absolute;
  top: 10px;
  left: 10px;
}
```

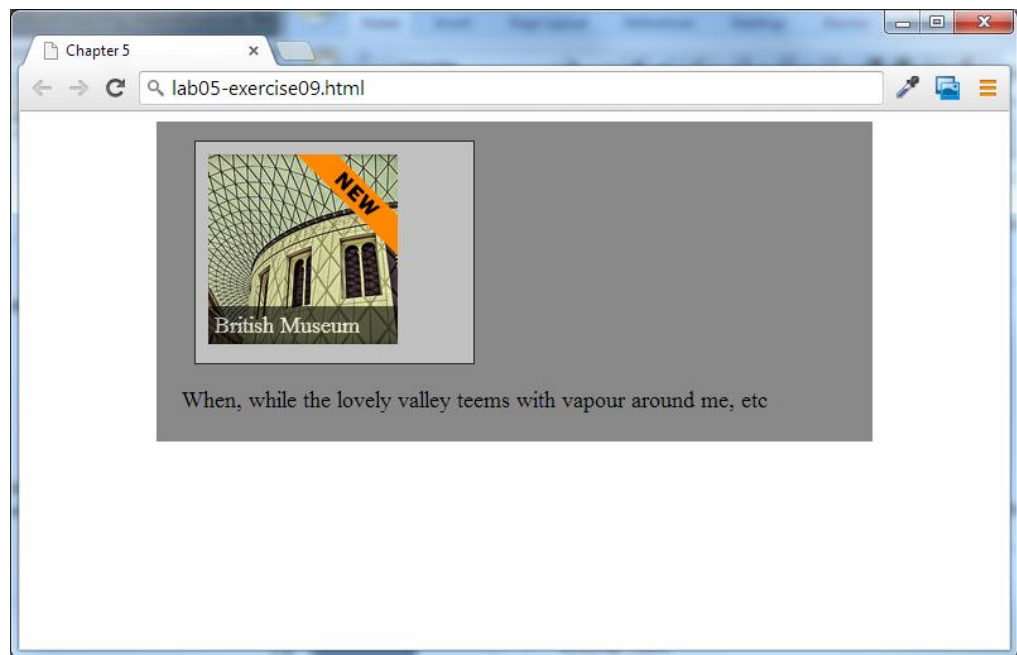


Figure 5.3 – Exercise 5.9 complete

LAYOUTS

The exercises so far in this lab showed two different ways to move items out of the normal top-down flow, namely, by using positioning and by using floats. They are the raw techniques that you can use to create more complex layouts.

Exercise 5.10 — TWO-COLUMN LAYOUT

1 Open, examine, and test lab05-exercise10.html.

2 Modify the following styles and test.

```
div#main {
  padding: 0.5em;
  margin-left: 12em;
}
nav {
  color: white;
  background-color: #A9A9A9;
  padding: 0.5em;
  width: 10em;
}
```

3 Modify the following style and test.

```
nav {
```

```
    color: white;
    background-color: #A9A9A9;
    padding: 0.5em;
    width: 10em;
    float: left;
}
```

The previous exercise created a multi-column layout using floats. You can also create a multi-column layout using positioning, as shown in the next exercise.

Exercise 5.11 — THREE-COLUMN LAYOUT

- 1 Open, examine, and test lab05-exercise11.html.
- 2 Modify the following styles and test.

```
nav {
    color: white;
    background-color: #A9A9A9;
    padding: 0.5em;
    width: 10em;
}
div#main {
    padding: 0.5em;
    margin-left: 12em;
    margin-right: 12em;
}
aside {
    color: white;
    background-color: #D3C6BA;
    padding: 0.5em;
    width: 10em;
}
```

- 3 Modify the following styles and test.

```
nav {
    color: white;
    background-color: #A9A9A9;
    padding: 0.5em;
    width: 10em;
    position: absolute;
    left: 0;
    top: 0;
}

aside {
    color: white;
    background-color: #D3C6BA;
    padding: 0.5em;
    width: 10em;
    position: absolute;
    right: 0;
    top: 0;
}
```

Can you figure out what is wrong (and the solution)? Remember that absolute positioning is relative to the last positioned element. There is no positioned element, so the measurements are relative to the browser window.

- 4 Add the following styles and test.

```
div#container {
    position: relative;
}
```

RESPONSIVE DESIGN

Exercise 5.12 — MEDIA QUERIES

- 1 Open, examine, and test lab05-exercise12.html.
- 2 Modify the following markup and test by progressively reducing the width of the browser window .

```
<link rel="stylesheet" href="lab05-exercise12-desktop.css"
      media="screen and (min-width:481px)" />
```

Once the window shrinks below 481px wide, this stylesheet will no longer be in effect (and hence all CSS styling will disappear).

- 3 Add the following additional media query.

```
<head lang="en">
  <meta charset="utf-8">

  <title>Chapter 5</title>
  <link rel="stylesheet" href="lab05-exercise12-mobile.css"
        media="screen and (max-width:480px)" />
  <link rel="stylesheet" href="lab05-exercise12-desktop.css"
        media="screen and (min-width:481px)" />
</head>
```

We are adding an additional style sheet for mobile browser sizes.

- 4 Edit lab05-exercise12-mobile.css (this is the mobile style sheet) as follows and test.

```
main {
    width: 100%;
    margin-left: auto;
    margin-right: auto;
}

nav {
    color: white;
    background-color: #A9A9A9;
    padding: 0.5em;
    width: 100%;
    /* be sure to delete the float as well */
}
```

```
div#main {
  padding: 0.5em;
  margin-left: 0;
}
```

```
figure img {
  max-width: 100%;
}
```

Notice that when the browser window shrinks to mobile size, the two-column layout is replaced with a single column. This requires removing the floats and the margins.

Exercise 5.13 — SETTING THE VIEWPORT

- 1 Continue this exercise by using your files from Exercise 12.

- 2 Add the following viewport.

```
<meta name="viewport" content="width=device-width" />
```

- 3 Add the following markup to the <nav> element.

```
<nav>
  <h3>Navigation</h3>
  <ul>
    <li><a href="#">Australia</a></li>
    <li><a href="#">Belgium</a></li>
    <li><a href="#">Canada</a></li>
    <li><a href="#">France</a></li>
    <li><a href="#">United Kingdom</a></li>
    <li><a href="#">United States</a></li>
  </ul>
  <select>
    <option>Australia</option>
    <option>Belgium</option>
    <option>Canada</option>
    <option>France</option>
    <option>United Kingdom</option>
    <option>United States</option>
  </select>
</nav>
```

We are going to use CSS to use the <select> element for mobile sizes, and the for larger sizes.

- 4 Edit lab05-exercise12-desktop.css (this is the desktop style sheet) as follows by adding the following style.

```
nav select {
  display: none;
}
```

- 5 Edit lab05-exercise12-mobile.css (this is the mobile style sheet) as follows by adding the following styles and test in browser. Be sure to test using small and larger browser widths.

```

nav ul      { display: none; }
nav select {
  display: inline-block;
  width: 90%;
}

```

GRID SYSTEMS

Exercise 5.14 — USING BOOTSTRAP

- 1 Create a new file named lab05-exercise14.html, add the following markup, and then test.

```

<head lang="en">
  <meta charset="utf-8">
  <title>Chapter 5</title>
  <link href="bootstrap-3.0.0/dist/css/bootstrap.css" rel="stylesheet">
</head>
<body>
  <div class="container">
    <div class="row">
      <div class="col-md-2">
        left column
      </div>
      <div class="col-md-7">
        main content
      </div>
      <div class="col-md-3">
        right column
      </div>
    </div>
  </div>
</body>

```

Notice that Bootstrap handles the entire layout without needing to bother with floats and positioning.

- 2 Add the following markup after the container and test.

```

<div class="container">
  <nav class="navbar navbar-default" role="navigation">
    <a class="navbar-brand" href="#">Logo</a>
    <ul class="nav navbar-nav">
      <li class="active"><a href="#">Home</a></li>
      <li><a href="#">Products</a></li>
      <li><a href="#">Search</a></li>
      <li><a href="#">About Us</a></li>
    </ul>
  </nav>

```

Bootstrap also comes with many additional classes that can simplify the process of quickly creating a design. For more information, see <http://getbootstrap.com/>.