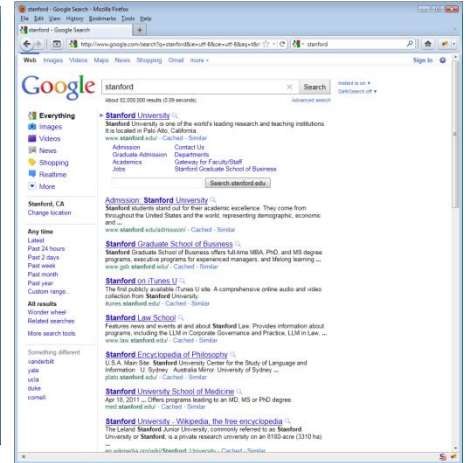# Float-based Layout

Dr. Patrick Young, Stanford University

CS193C

Most webpages contain columns. Consider for example these three webpages:

While a variety of techniques can be used to create columns on a webpage, one of the most commonly used is taking advantage of the CSS float property. That's how we'll learn how to create columns.

Float Review

As you'll recall, the CSS float property places an image on the left or right side of a webpage and floats the text alongside the image. Here is a simple webpage using float:
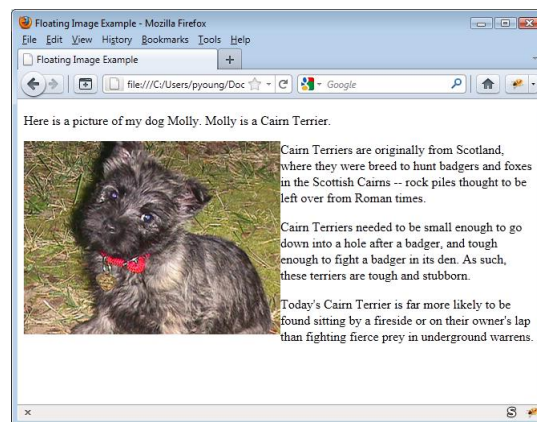
The `<img>` was set with the `id="puppy"` and here a style rule we used to float this image:

```
#puppy {float: left}
```

While floating images is very common, we can also float other items, such as the blockquote shown here:
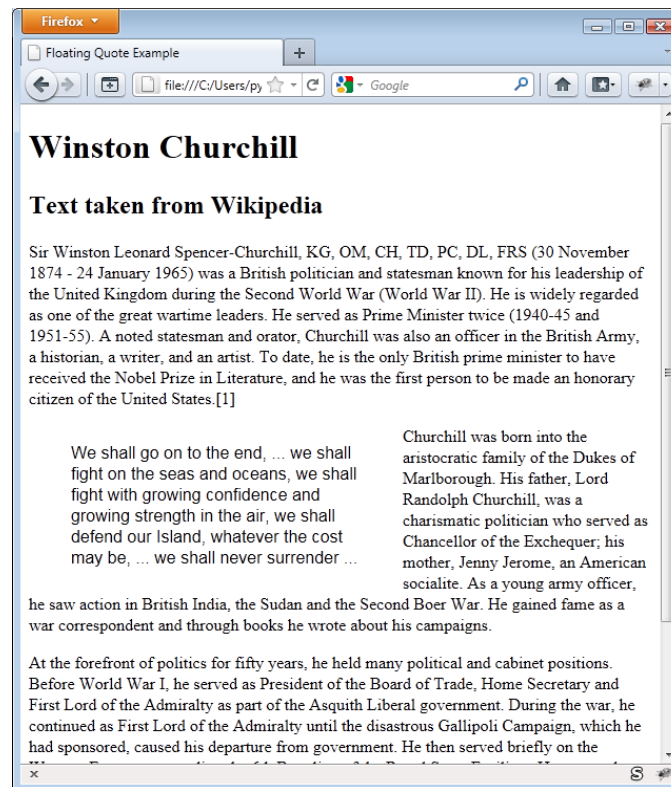
Here is our HTML:

```
...
<p>Sir Winston Leonard Spencer-Churchill, KG, OM, CH, TD, PC,
DL, FRS ...</p>

<blockquote id="ww2Quote">We shall go on to the end, ... we
shall fight on the seas and oceans, we shall fight with
growing confidence and growing strength in the air, we shall
defend our Island, whatever the cost may be, ... we shall
never surrender ...</blockquote>

<p>Churchill was born ....</p>
...
```
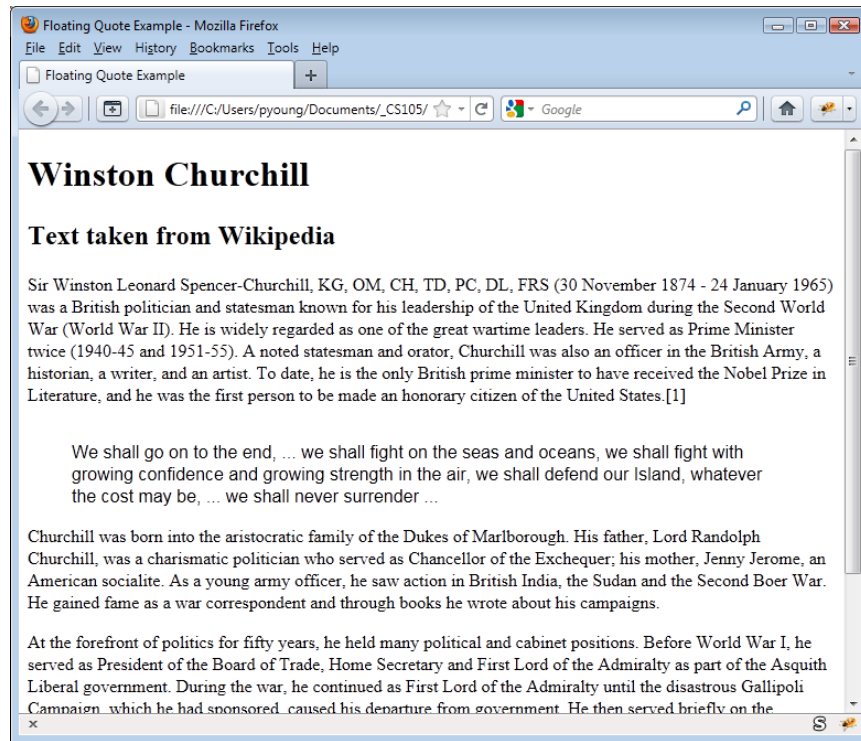
And our corresponding CSS

```
#ww2Quote {
    float: left;
    width: 275px;
    font-family: sans-serif;
}
```

One important difference between floating an image and floating a blockquote, images have a natural width, blockquotes can expand out to fill as much space as is available to them. Without a width, our blockquote will look like this:

The quote is floated to the left, but the quote has expanded to take up the entire width of the webpage, squeezing out any space for text to flow alongside.

Floating Multiple Items

What happens if we create a second image or second blockquote and float it as well? The web browser will float the two items next to each other. Here we've duplicated the blockquote and created a second style rule, which is the same as the first style rule. Now our style rules are:

```
#ww2Quote {float: left;
          width: 275px;
          font-family: sans-serif;
}
#secondQuote {float: left;
          width: 275px;
          font-family: sans-serif;
}
```

Here's what our two blockquotes look like in the web browser:
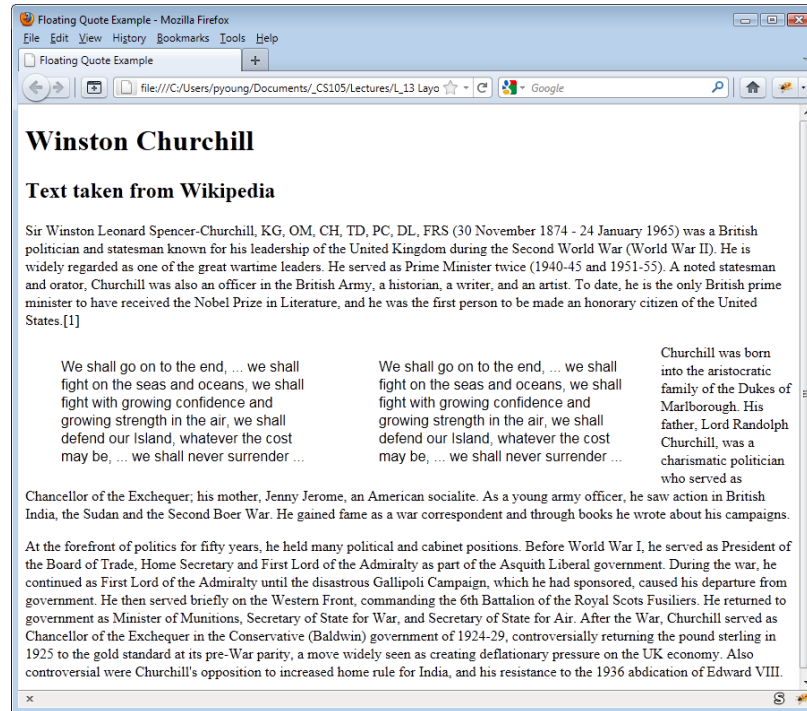
As you can see our two quotes end up side-by-side. If we remove the rest of the text of the website, we discover we essentially have a basic two column layout:
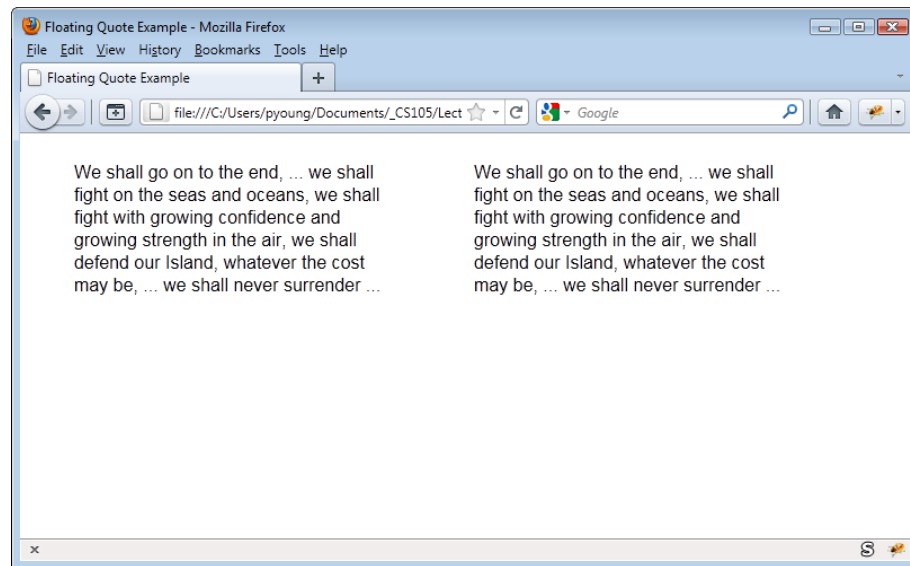
We just need to make a few changes and we'll have a basic two-column webpage. First, we switch from using the blockquote element and instead use a more generic div element. While blockquote has a default margin, div does not, so we'll add a bit of a margin. With a div, we can place whatever text we want into the div, including headings and paragraphs. Here's what this might look like in the web browser:
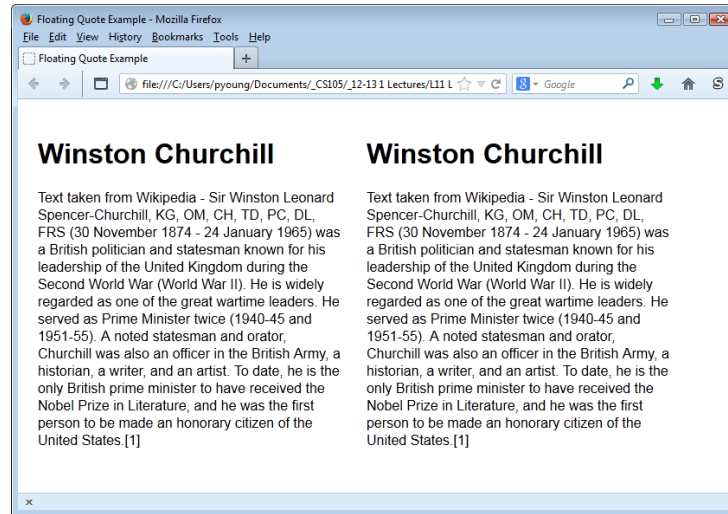
Here's our revised HTML and CSS:

```html
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8" />
<title>Floating Quote Example</title>
<style>
#firstDiv {float: left;
           width: 350px;
           font-family: sans-serif;
           margin: 15px;
}
#secondDiv {float: left;
           width: 350px;
           font-family: sans-serif;
           margin: 15px;
}
</style>
</head>
<body>

<div id="firstDiv">
<h1>Winston Churchill</h1>

<p> Text taken from Wikipedia - Sir Winston Leonard Spencer-
Churchill, KG, OM, CH, TD, PC, DL, FRS (30 November 1874 - 24
January 1965) was a British politician …</p>
</div>

<div id="secondDiv">
<h1>Winston Churchill</h1>

<p> Text taken from Wikipedia - Sir Winston Leonard Spencer-
Churchill, KG, OM, CH, TD, PC, DL, FRS (30 November 1874 - 24
January 1965) was a British politician …</p>
</div>

</body>
</html>
```

Creating Classes

If you look at our style sheet, you can see that we have two rules which are exactly the same:

```
<style>
#firstDiv {float: left;
           width: 350px;
           font-family: sans-serif;
           margin: 15px;
}
#secondDiv {float: left;
           width: 350px;
           font-family: sans-serif;
           margin: 15px;
}
</style>
```

So it might make sense to use a class here instead of using IDs like this:

```
 <style>
.column {float: left;
         width: 350px;
         font-family: sans-serif;
         margin: 15px;
}
</style>
```

However, this would only make sense if all our columns were 350 pixels. In fact as we'll see in a few moments, we'll want columns of different widths. So instead we'll break this in to two classes a column class for all columns and a mid-width class for columns of 350 pixels. Depending on your design you'll have to decide if the 15px margin goes with all columns or if you'll want different columns to have different margins. The same goes for the font-family. For our example I'll use the following style rules:

```
<style>
.column {float: left;
         font-family: sans-serif;
         margin: 15px;
}
.mid-width { width: 350px; }
</style>
```

You may recall that we can list multiple classes in an HTML tag's class attribute if we want more than one class rule to apply. So here's our revised file:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8" />
<title>Floating Quote Example</title>
<style>
.column {float: left;
         font-family: sans-serif;
         margin: 15px;
}
.mid-width { width: 350px; }
</style>

</style>
</head>
<body>

<div class="column mid-width">
<h1>Winston Churchill</h1>

<p> Text taken from Wikipedia - Sir Winston Leonard Spencer-
Churchill, …</p>
</div>

<div class="column mid-width">
<h1>Winston Churchill</h1>

<p> Text taken from Wikipedia - Sir Winston Leonard Spencer-
Churchill, …</p>
</div>

</body>
</html>
```
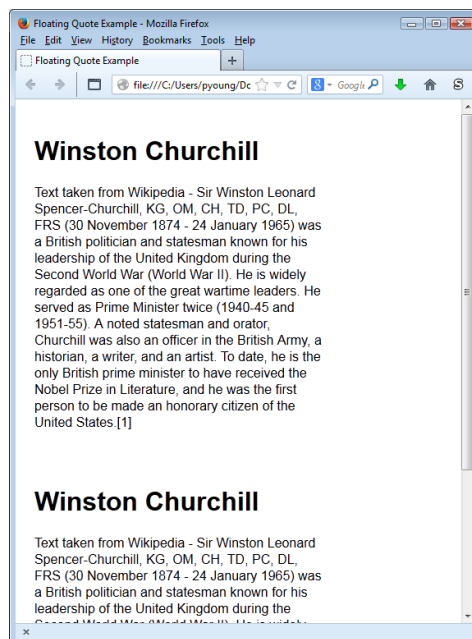
Width Issues

One problem that will arise is what happens if the web browser window gets to narrow for both our divs to appear side-by-side. When this happens the divs stop floating next to each other and instead appear one on top of the other like this:

This generally isn't what we want to happen. We can correct the problem by forcing the webpage to a specific width. The simplest way to do this is to set a width on the body itself. However, a more common technique is to place the entire webpage inside of a div. There's only one of these, so we'll use an ID for this. Here's our revised body with the `<div id="main">` added:

```
…
<body>

<div id="main">
<div class="column mid-width">
<h1>Winston Churchill</h1>

<p>Text taken from Wikipedia - Sir Winston Leonard Spencer-
Churchill, KG, OM, CH, TD, PC, DL, FRS (30 November 1874 - 24
January 1965) was a British politician …</p>
</div>

<div class="column mid-width">
<h1>Winston Churchill</h1>

<p>Text taken from Wikipedia - Sir Winston Leonard Spencer-
Churchill, KG, OM, CH, TD, PC, DL, FRS (30 November 1874 - 24
January 1965) was a British politician …</p>
</div>
</div>
</body>
…
```

We now give our surrounding div a width:

```
#main {width: 760px}
```

Make sure you make this div wide enough to fit the divs. If the surrounding div is too narrow, the divs won't be able to sit side by side, leading to our original situation in which one div was placed on top of the other. Don't forget the inner divs' width does not including margin, padding, or border. The surrounding div needs to be wide enough to account for not only the divs inside, but any margin, padding, and border you place around them.

Depending on what you want your webpage to look like you can make one addition to this style rule. If you set margin to auto, the web browser will automatically center your webpage content.

```
#main {
    width: 760px;
    margin: auto;
}
```

You can even put a border around the entire webpage contents like this:

```
#main {width: 760px;
    border: 1px solid black;
    margin: auto;
    overflow: hidden;
}
```

We'll discuss the overflow hidden added to this style rule in the next section.

Mixed Layouts

While columns are central to most websites, real websites don't just have a single set of columns. Instead they are composed of different sections, where each section is composed of columns:

We will now learn how to create these different sections. As the term *section* already has specific meaning in HTML, I will instead refer to these as panels.

We'll start out by trying to make this webpage layout with three columns in the upper panel and two columns in the lower panel:
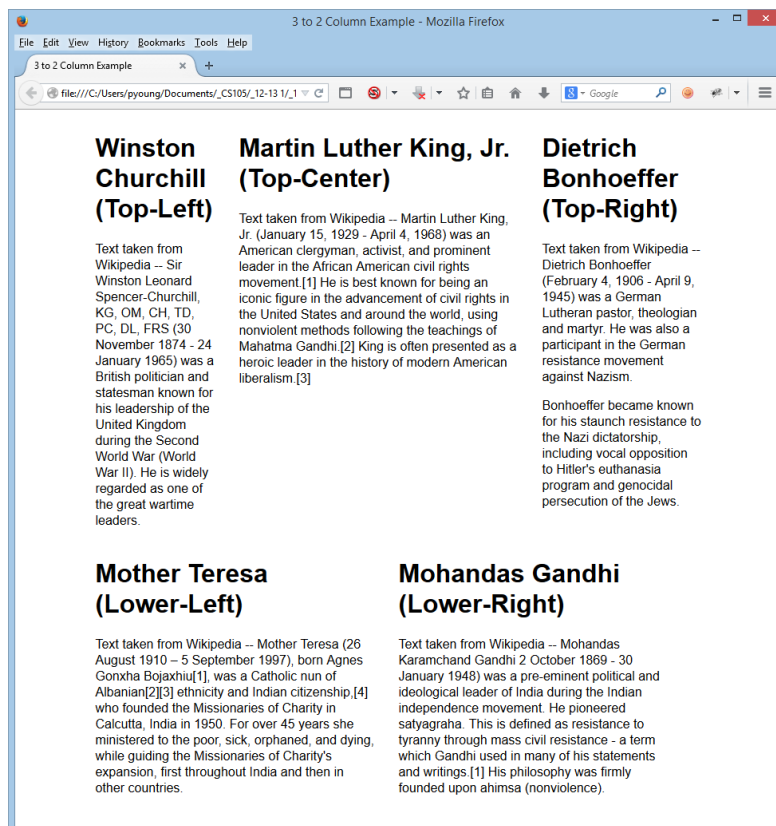
Div Elements

We'll use div elements to represent each of our panels. To see why this works consider how a web browser usually displays two block elements, whether those elements are p paragraph elements, blockquote elements, h1, …, h6 elements, or div elements.   The browser places the elements one on top of the other. For example, if my HTML source contains:

```
<p>Paragraph 1</p>
<p>Paragraph 2</p>
```

the web browser will display paragraph 1 followed by paragraph 2 like this:

Paragraph 1

Paragraph 2

This is essentially the same behavior we want for our 3-column to 2-column webpage. But instead of simply wanting one paragraph on top of the other element, we want an entire 3-column panel on top of a 2-column panel.

In order to get this effect we'll place the columns into a div. So if our HTML looks something like this:

```
<div>Stuff creating top three-column-wide panel</div>
<div>Stuff creating bottom two-column-wide panel</div>
```

The web browser will dutifully place the three-column div above the two-column div—which is just what we want.

Here's how we'll setup our divs:

```
<body>

<div id="main"> <!-- div for entire webpage contents -->

<div class="panel"> <!-- div for entire upper half of webpage -->
<div class="column very-narrow">
<h1>Winston Churchill (Top-Left)</h1>

<p>Text taken from Wikipedia -- Sir Winston Leonard Spencer-Churchill,
KG, OM, CH, TD, PC, DL, FRS ...</p>
</div>

<div  class="column wide">
<h1>Martin Luther King, Jr. (Top-Center)</h1>

<p>Text taken from Wikipedia -- Martin Luther King, Jr. (January 15,
1929 - April 4, 1968) was an American clergyman, ...</p>
</div>

<div class="column narrow">
<h1>Dietrich Bonhoeffer (Top-Right)</h1>

<p>Text taken from Wikipedia -- Dietrich Bonhoeffer (February 4, 1906
- April 9, 1945) was a German Lutheran pastor, theologian and martyr.
```

```
He was also a participant in the German resistance movement against
Nazism ...</p>
</div>
</div> <!-- end upper half panel -->

<div class="panel"> <!-- div for entire lower half of webpage -->
<div  class="column wide">
<h1>Mother Teresa (Lower-Left)</h1>

<p>Text taken from Wikipedia --  Mother Teresa (26 August 1910 - 5
September 1997), born Agnes Gonxha Bojaxhiu[1], was a Catholic nun of
Albanian[2][3] ethnicity and Indian citizenship,[4] ...</p>
</div>

<div  class="column wide"><h1>Mohandas Gandhi (Lower-Right)</h1>

<p>Text taken from Wikipedia -- Mohandas Karamchand Gandhi  2 October
1869 - 30 January 1948) was a pre-eminent political and ideological
leader of India during the Indian independence movement. ... </p>
</div>
</div> <!-- end lower half panel -->

</div> <!-- end div for entire webpage id="main" -->
</body>
```

We'll use the techniques we used in the last handout to create our columns—we'll float each of the inner divs representing the columns to the left and we'll set them to specific widths:

```
.column {float: left;
      font-family: sans-serif;
      margin: 0px 15px;
          /* 0px top and bottom, 15px left-and-right */
}

.very-narrow { width: 150px; }
.narrow { width: 200px; }
.wide { width: 350px; }

.panel {
   overflow: hidden;
}

#main {width: 790px;
   margin: auto;
}
```

The overflow hidden is important. The technical aspects of what this does is a bit complex. It forces the web browser to create what's called a "Block Formatting Context". The important point however, is that without setting this (and the creation of the new Context), items from the bottom panel will float up to the top panel if there is sufficient space for them. In this next screenshot, I've removed the overflow: hidden on from the panel style rule and shortened the top-right section on Dietrich Bonhoeffer leaving some empty space, and the web browser responds to by moving Mother Theresa's article up into that empty space:
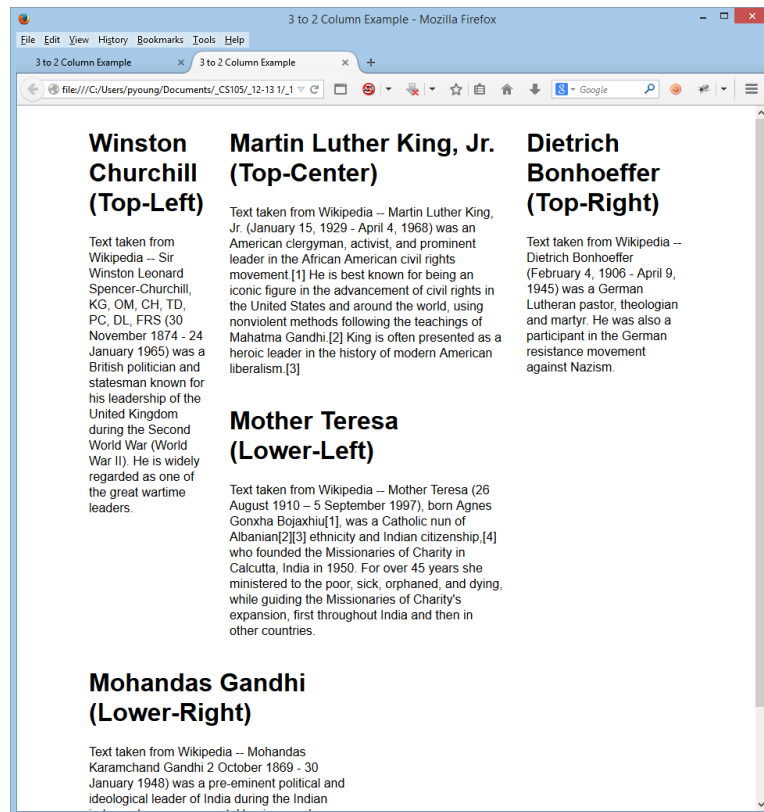
If you start seeing this type of behavior, double check and make sure you have overflow: hidden set.

More Complex Layouts

Suppose we want to create a layout which looks something like this (original text from Stanford News Service):
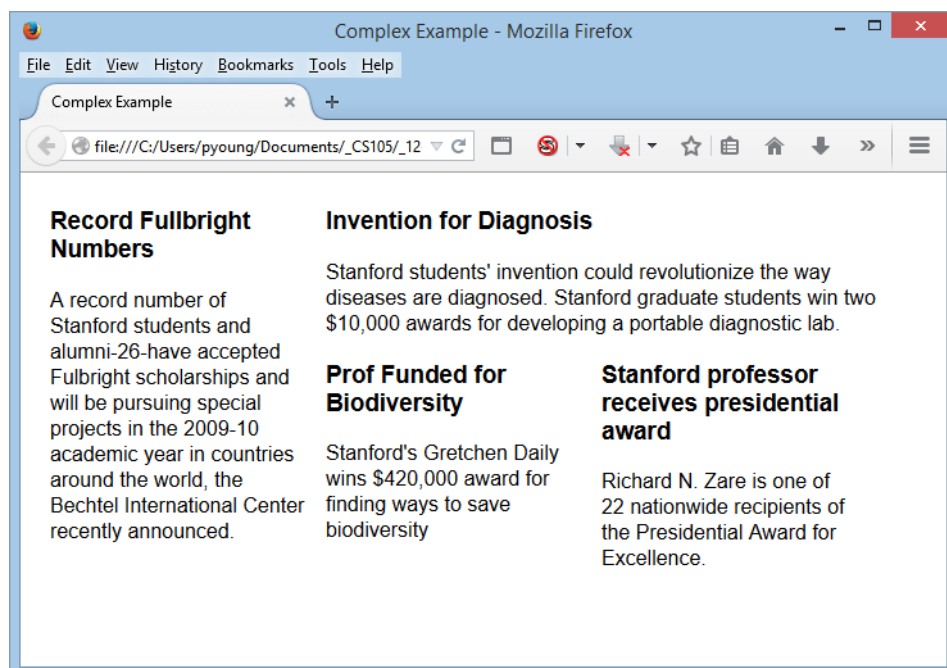
The way to think of this is that the main page consists of a main left-column and a main right-column so it's basically a two-column layout, which we know how to do.

The main right-column can be further subdivided. It consists of a one-column panel on top of a two-column panel. You can work this problem by solving the inner elements (the right-hand main column consisting of the "Invention for Diagnosis," "Prof Funded for Biodiversity," and "Stanford Professor receives presidential award") and then placing it in the context of a two-column layout.

Let's start out by laying out the right-hand side. We'll create a div containing all three stories, a top panel div for the top story "Invention for Diagnosis" a bottom panel div for to contain the two bottom-right stories. The bottom panel div will contain two divs for each of the columns "Prof Funded" and "Stanford professor."

```html
<div class="column wide"> <!-- Main Right Column -->

<div class="panel"> <!-- top right panel -->
<h3>Invention for Diagnosis</h3>
Stanford students' invention could revolutionize …
</div> <!-- end top right panel -->

<div class="panel"> <!-- bottom right panel -->
<div class="column standard-width">
<h3>Prof Funded for Biodiversity</h3>
Stanford's Gretchen Daily wins $420,000 award for finding
ways to save biodiversity</div>

<div class="column standard-width">
<h3>Stanford professor receives presidential award</h3>
Richard N. Zare is one of 22 nationwide recipients of the
Presidential Award for Excellence.</div>

</div> <!-- end bottom right panel -->

</div> <!-- end  Main Right Column  -->
```

Here are our style rules:

```css
.column {
   float: left;
   margin-right: 15px;
}

.panel {
   overflow: hidden;
}

.wide { width: 460px; }
.standard-width { width: 200px; }
```

We use our standard column and panel settings. We create a class to set the entire right column to 460 pixels and then a second 200 pixels wide for the two bottom columns. You'll want to experiment with margin settings. These can be a bit more complex because of the columns inside of columns. Suppose for example we used a left-margin of 15 pixels. In this case the bottom columns have both a margin from the main right column as well as their own margins so we would actually see a 30 pixel margin (15 pixels + 15 pixels), whereas the top panel does not have an inner column div and thus would only has a single margin 15 pixel margin. I've

masked this by eliminating the left-margin (the same issue will show up on the right, but will be far less obvious). Alternatively the top panel story could be contained within its own internal column, thus both top and bottom would be contained within inner and outer divs.

Now we place that whole thing in the context of the larger webpage:

```
<div id="main">

<div class="column standard-width"> <!-- Main Left Column -->
<h3>Record Fullbright Numbers</h3>
A record number of Stanford students and alumni-26-have
accepted Fulbright scholarships …
</div> <!-- end Main Left Column -->

<div class="column wide"> <!-- Main Right Column -->

… <!-- insert the right column contents we just wrote above -->

</div>
</div> <!-- end  Main Right Column  -->
</div> <!-- end main -->
```

Since the left and right main columns have class="column" they will automatically be floated, we just have to make sure their widths are right. In this case, I simply reuse the wide and standard-width classes I previously created.

Here's my entire HTML file including the CSS:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8" />
<title>Complex Example</title>
<style>
.column {
   float: left;
   margin-right: 15px;
}

.panel {
   overflow: hidden;
}

.wide { width: 460px; }
.standard-width { width: 200px; }

#main {
   width: 690px;
   font-family: sans-serif;
   margin-left: 15px;
}

</style>
</head>

<body>
<div id="main">
```

```
<div class="column standard-width"> <!-- Main Left Column -->
<h3>Record Fullbright Numbers</h3>
A record number of Stanford students and alumni-26-have
accepted Fulbright scholarships …
</div> <!-- end Main Left Column -->

<div class="column wide"> <!-- Main Right Column -->

<div class="panel"> <!-- top right panel -->
<h3>Invention for Diagnosis</h3>
Stanford students' invention could revolutionize …
</div> <!-- end top right panel -->

<div class="panel"> <!-- bottom right panel -->
<div class="column standard-width">
<h3>Prof Funded for Biodiversity</h3>
Stanford's Gretchen Daily wins $420,000 award for …
</div>
<div class="column standard-width">
<h3>Stanford professor receives presidential award</h3>
Richard N. Zare is one of 22 nationwide recipients …
</div>
</div> <!-- end bottom right panel -->

</div> <!-- end  Main Right Column  -->

</div> <!-- end main -->
</body>
</html>
```

Webpage Layout Summary

- Columns are a basic building block of most websites.

- In order to create a column, place the elements within the column into a div. Float the div element for each column to the left. Make sure each column has a specific width.

- Combine columns into panels which are themselves div elements. Each panel can have a different arrangements of columns. For example some panels may have three columns and others may contain only two columns.

- These panels will stack one on top of each other, as div elements will appear one on top of the next without any intervention on your part.

- Each panel containing floating columns must have `overflow: hidden` set.

- Place all your webpage elements into an overall containing div. Set a width on this div, otherwise your floating elements may collapse if the user narrows the web browser window.

- The overall containing div may also be used to put a border around your webpage. You may also center your webpage content by setting the CSS property margin: auto. Make sure you set overflow: hidden on this div as well as on the panel divs.