# ACS-1809-051
Web Design and Development

Chapter 6

A note on the use of these slides:

These slides has been adopted and/or modified from the original for the use in this course. The author of the text have make these slides available to all (faculty, students, readers) and they obviously represent a *lot* of work on their part.

In return for use, please:

- If slides are being used (e.g., in a class) that the source be mentioned (after all, the author like people to use our book!)
- If any slides are being posted on a www site, note that they are adapted from (or perhaps identical to) the author original slides, and note their copyright of this material.

© McGraw-Hill Osborne Media 2013, HTML. A beginners guide, 5th Edition
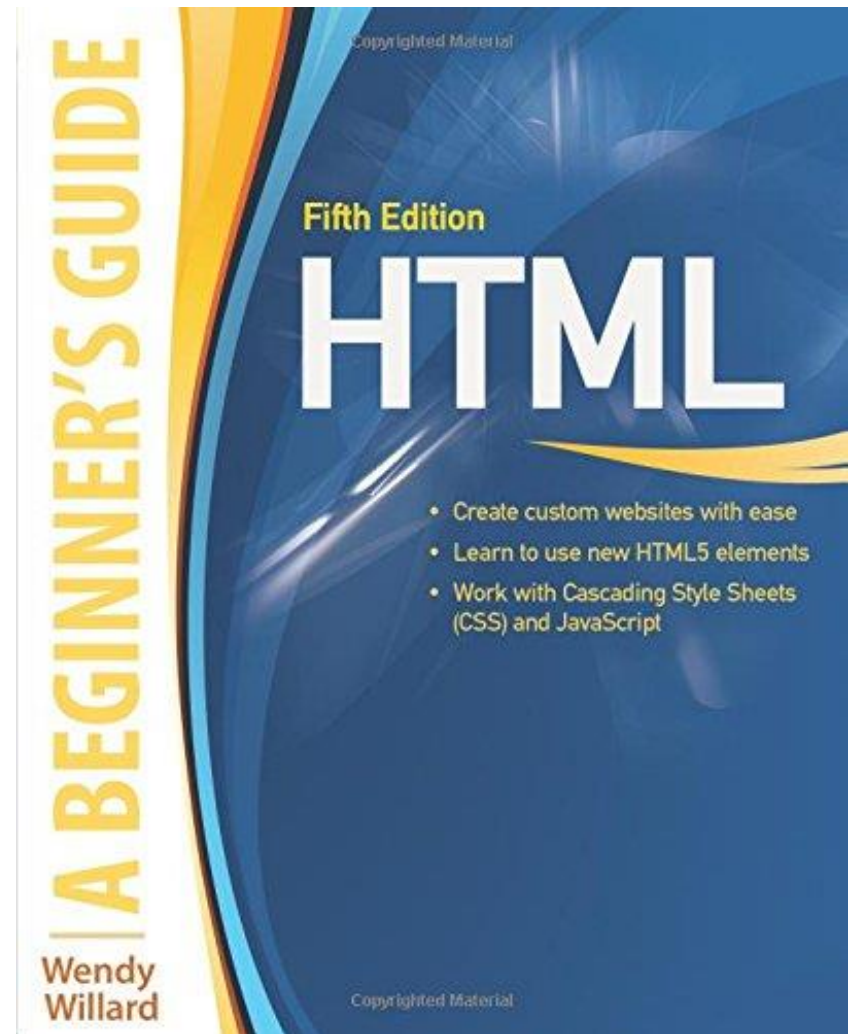
Back to CSS Again

- Understand the concept and uses of Styles Sheets for Page Layout
- Create a Single-Column, centered, Fluid Page Layout
- Create a Multicolumn Fluid Page Layout
- Layer Content Within a Layout

DISCOVER · ACHIEVE · BELONG

Why CSS (AGAIN!)?

- The ultimate purpose
  - To separate the content of a web site from its design (style, look, …)
    - Content is the king
    - Anything else should be icing on the cake
  - This would make maintaining a website a lot easier
  - Leaving the style (look) of a website to CSS completely also greatly benefit redesign the website
    - Check http://www.csszengarden.com/

Example

- Creating a single-column centered, fluid page layout

```css
body {
  margin: 0px;
  padding: 0px;
}
#content {
  margin: 50px;
  padding: 20px;
  color: black;
  background-color: #CCC;
  border: 1px dashed black;
}
```

DISCOVER · ACHIEVE · BELONG

Example

```
p {
    font-size: 12pt;
    line-height: 20pt;
    font-family: verdana, arial, helvetica, sans-serif;
    margin: 0px 0px 12px 0px;
 }
h3 {
    font-size: 20pt;
    font-family: wingding;
    font-style: italic
}
```

Example

- Creating a multi-column fluid page layout

```
body {
    margin: 10px 10px 0px 10px;
    padding: 0px;
    }
#header {
    height: 50px;
    background-color: #CCC;
    padding: 5px;
    }
#navigation {
    position: absolute;
    left: 10px;
    top: 70px;
    width: 150px;
    border: 1px solid #000;
    padding: 5px;
    }
```

DISCOVER · ACHIEVE · BELONG

Example

```css
#extra {
    position: absolute;
    right: 10px;
    top: 70px;
    width: 150px;
    border: 1px solid #396;
    padding: 5px;
    }
#content {
    margin-left: 162px;
    margin-right: 162px;
    padding: 5px;
    color: yellow;
    background-color: #666;
    }
#footer {
    border-top: 2px solid #000;
    padding: 5px;
    }
```

DISCOVER · ACHIEVE · BELONG

Example

- Something new here?
  - Position
    - static
      - Like "no position" – let the browser handle the position of the element
    - relative
      - Positions the element relative to its original position

        Position: relative; bottom: 50px;
    - absolute
      - Precisely position the element on the page
        » Not following the normal (static) flow

DISCOVER · ACHIEVE · BELONG

Layered Content

- Layered content within a layout
  - with the help of position property

        The style declaration of two divisions:

        #apple {

            position: absolute;

            left: 20px;

            top: 20px;

            width: 200px;

            padding: 10px;

            text-align: center;

            background-color: #ccc;

            border: 1px dashed #000;

            }

DISCOVER · ACHIEVE · BELONG

Layered Content

```
#orange {

        position: relative;

        width: 200px;

        padding: 10px;

        text-align: center;

        background-color: #333;

        color: #fff;

        border: 3px solid #999;

        }
```

Layered Content

– Remember we mentioned the absolutely positioned elements are removed from the normal page layout flow from the browser?

- So the division of "orange" is positioned right to the upper left corner as it's the first element handled in the page layout flow

– By default the relatively positioned element is placed above the absolutely positioned element

- So "orange" is above "apple"

DISCOVER · ACHIEVE · BELONG

Layered Content

– What if we want to place "apple" above "orange"

    • Then we can use the ***z-index*** property

      – The element with the highest z-index value is the one on top

In #apple {

    position: absolute;

    …

    border: 1px dashed #000;

    }

We add "z-index: 2;",

and add "z-index: 1;" to "orange".