

WHAT WE WILL COVER TODAY

- **HTML** – the primary language that is used to create Web pages. We will be using the latest version which is XHTML
- **Markup** – a set of simple tags that suggest the structure of a document. The first part of the morning we will be looking at the various tags you will need to use in order to build a Web page.
- **CSS** – a language that lets you control how a Web page looks. We will spend a majority of our time looking at CSS. We will use it to style page elements such as text and use it to position elements on the Web page.

WHAT TOOLS DO I NEED TO GET STARTED?

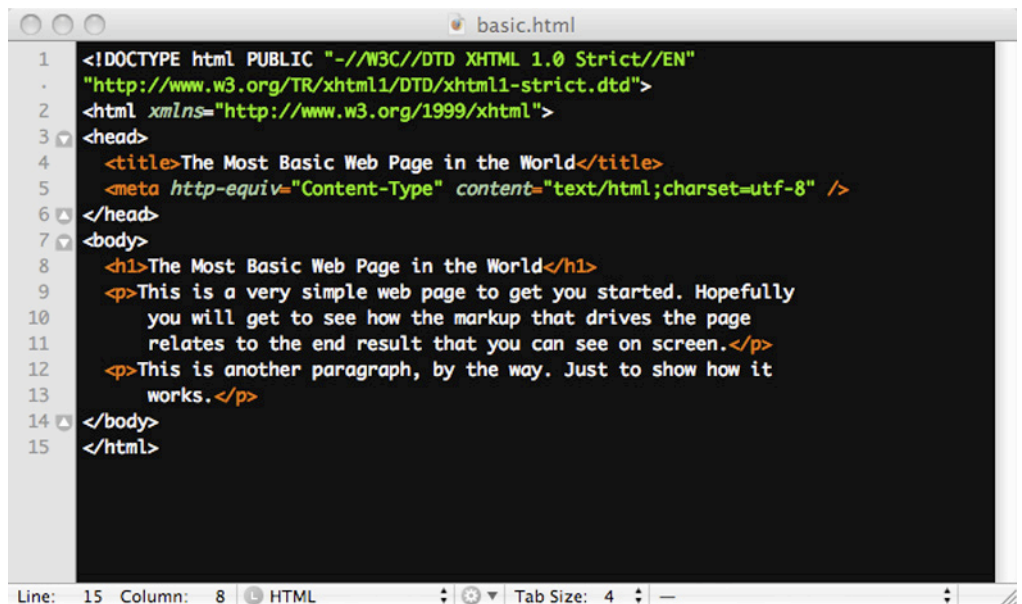
- Text Editor - TextEdit or Notepad
- Browser – Firefox is a great one to test in
- Image Editor – iPhoto and Picassa are good first options, GIMP is another one that I have heard good things about

GETTING STARTED WITH HTML

Elements are the basic building blocks of HTML and tell the Web browser what a particular item in a page is:

- Paragraph
- Heading
- Quotation
- So on

Let's look at a basic page in HTML



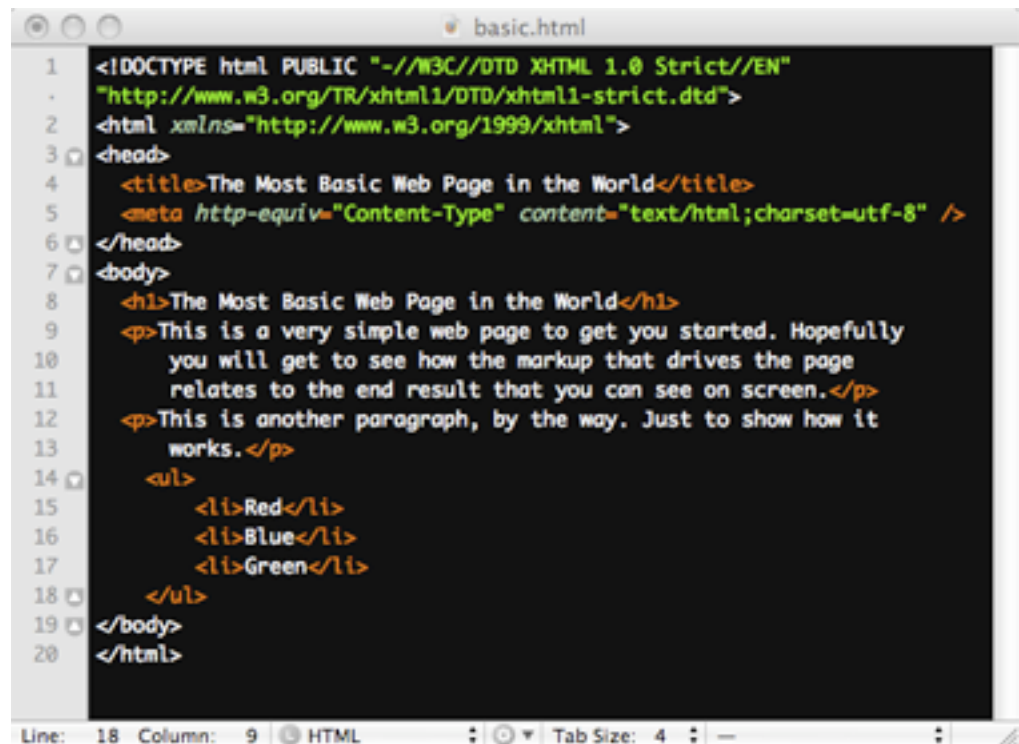
```
basic.html
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4   <title>The Most Basic Web Page in the World</title>
5   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6 </head>
7 <body>
8   <h1>The Most Basic Web Page in the World</h1>
9   <p>This is a very simple web page to get you started. Hopefully
10  you will get to see how the markup that drives the page
11  relates to the end result that you can see on screen.</p>
12  <p>This is another paragraph, by the way. Just to show how it
13  works.</p>
14 </body>
15 </html>
```

Line: 15 Column: 8 HTML Tab Size: 4

ITEMS YOU MUST HAVE TO BUILD A WEB PAGE

The basic items you must have in order to build a Web page: (tags tell the browser what element you are using)

- Doctype – tells the browser to expect a certain version of HTML
- <html> tag
- <head> tag – contains the information about the page such
 - <title> tag - tells the the browser what to display in the title bar
 - meta data – provide additional information that is not seen on the screen such as for search engines
 - CSS markup and Javascript code
- <body> tag – this is the place that it all happens. This tag contains everything that you see on the screen
- closing tags for all the tag elements



```
basic.html
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4   <title>The Most Basic Web Page in the World</title>
5   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6 </head>
7 <body>
8   <h1>The Most Basic Web Page in the World</h1>
9   <p>This is a very simple web page to get you started. Hopefully
10  you will get to see how the markup that drives the page
11  relates to the end result that you can see on screen.</p>
12  <p>This is another paragraph, by the way. Just to show how it
13  works.</p>
14  <ul>
15    <li>Red</li>
16    <li>Blue</li>
17    <li>Green</li>
18  </ul>
19 </body>
20 </html>
```

Line: 18 Column: 9 HTML Tab Size: 4

The Basic elements or tags that you will use frequently

- <h1>, <h2>, <h3>, ...
- <p>
- and

SYMBOLS AND ENTITIES

The Most Common Character Entities:

| Result | Description | Entity Name | Entity Number |
|--------|--------------------|------------------------------|---------------|
| | non-breaking space | | |
| < | less than | < | < |
| > | greater than | > | > |
| & | ampersand | & | & |
| " | quotation mark | " | " |
| ' | apostrophe | ' (does not work in IE) | ' |

Some Other Commonly Used Character Entities:

| Result | Description | Entity Name | Entity Number |
|--------|----------------------|-------------|---------------|
| ¢ | cent | ¢ | ¢ |
| £ | pound | £ | £ |
| ¥ | yen | ¥ | ¥ |
| € | euro | € | € |
| § | section | § | § |
| © | copyright | © | © |
| ® | registered trademark | ® | ® |
| × | multiplication | × | × |
| ÷ | division | ÷ | ÷ |

ADDING EVEN MORE STRUCTURE TO A PAGE

The <div> tag

```
basic.html
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <title>The Most Basic Web Page in the World</title>
5 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6 </head>
7 <body>
8 <div id="content_main">
9 <h1>The Most Basic Web Page in the World</h1>
10 <p>This is a very simple web page to get you started. Hopefully you
  will get to see how the markup that drives the page relates to the end
  result that you can see on screen.</p>
11 <p>This is another paragraph, by the way. Just to show how it
  works.</p>
12 <ul>
13 <li>Red</li>
14 <li>Blue</li>
15 <li>Green</li>
16 </ul>
17 </div>
18 <div id="content_secondary">
19 <p>This where I would put content that is not the main focus of a
  page.</p>
20 </div>
21 </body>
22 </html>
```

BEST PRACTICE use elements for the purpose for which they were intended

LINKS <a> tag

- this is the fundamental thing that makes the Web function as the Web
- the ability to link to other pages, other sites
- brings in the element of interactivity

BEST PRACTICE never say “click here” for a link

A COUPLE OF OTHER TAGS

- <blockquote> - use it with quotations
- <cite> - source the quotation came from
- and - emphasize content > BP use sparingly
-
 - do not use to space between paragraphs

LET'S PUT THIS INTO PRACTICE

You can download the basic code from my Web site:

<http://jeffbridgforth.com/website101>

STYLIN' WITH CSS

As we mentioned earlier, CSS is a language that we use to control how a page looks. It is the “presentation” layer of a Web page.

Allows you to change the look of elements on a page

- text size
- text color
- bolding of text
- background colors
- border styles
- padding and margin

A CSS RULE

Let us look at what the anatomy of a CSS Rule

```
p{
  font-weight: bold;
  color:blue;
}
```

Selector

```
p{
  font-weight: bold;
  color:blue;
}
```

Curly Brace

```
p{
  font-weight: bold;
  color:blue;
}
```

Property

```
p{
  font-weight: bold;
  color:blue;
}
```

Value

```
p{
  font-weight: bold;
  color:blue;
}
```

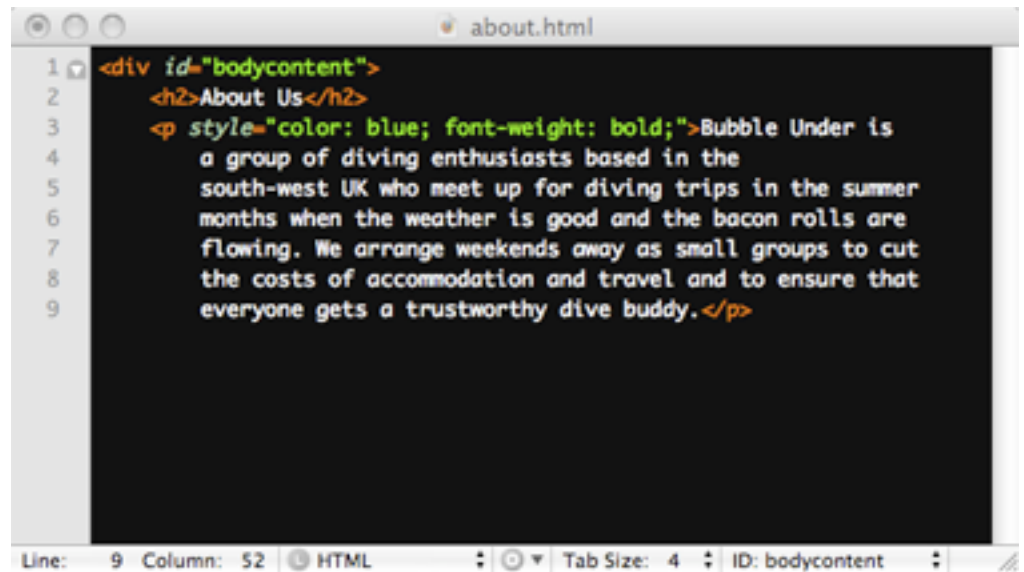
Declaration

```
font-weight: bold;
```

HOW DO WE ADD STYLING TO OUR HTML DOCUMENT?

There are three different ways to add styling to an HTML document:

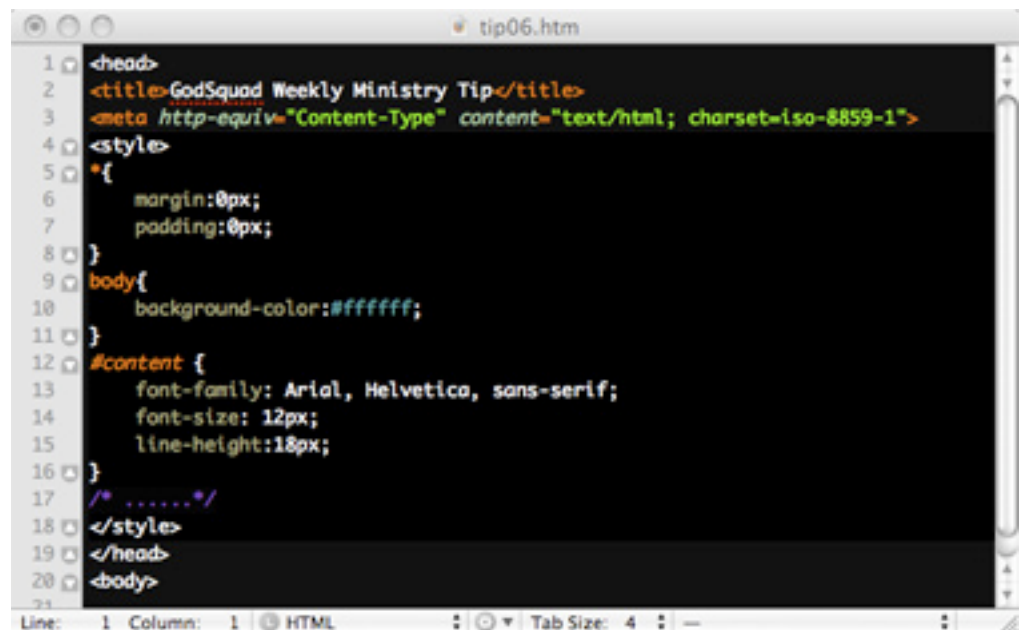
- **Inline Styles**



```
about.html
1 <div id="bodycontent">
2   <h2>About Us</h2>
3   <p style="color: blue; font-weight: bold;">Bubble Under is
4     a group of diving enthusiasts based in the
5     south-west UK who meet up for diving trips in the summer
6     months when the weather is good and the bacon rolls are
7     flowing. We arrange weekends away as small groups to cut
8     the costs of accommodation and travel and to ensure that
9     everyone gets a trustworthy dive buddy.</p>
```

Line: 9 Column: 52 HTML Tab Size: 4 ID: bodycontent

- **Embedded Styles**

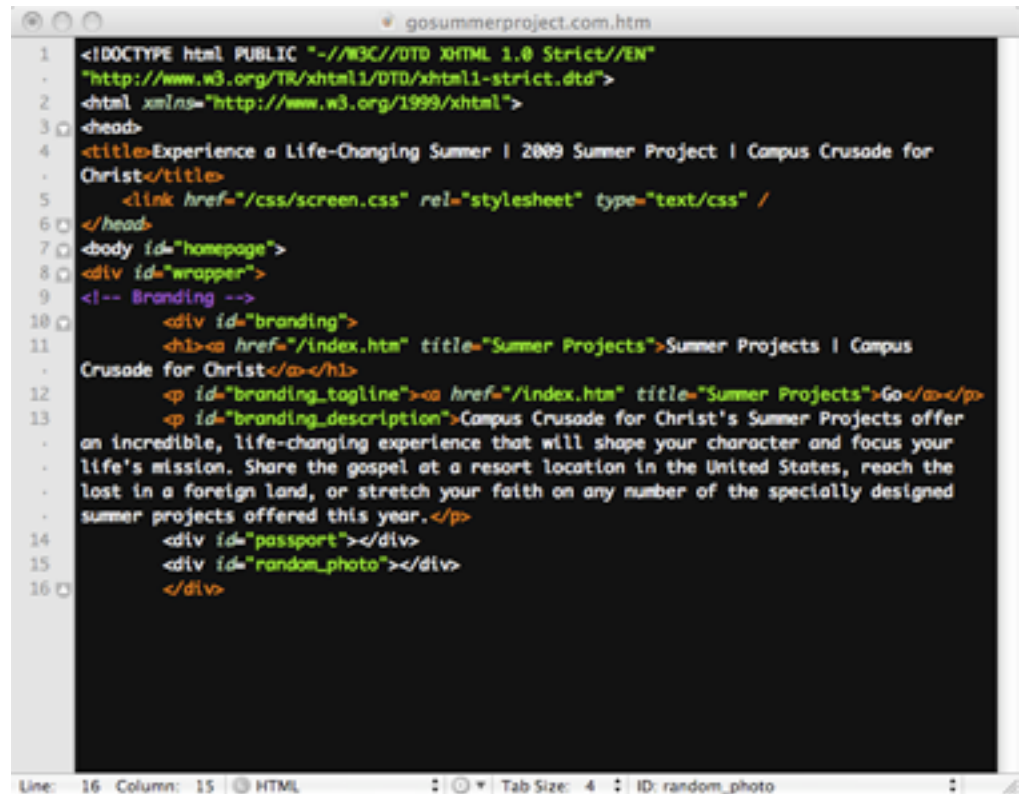


```
tip06.htm
1 <head>
2   <title>GodSquad Weekly Ministry Tip</title>
3   <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
4   <style>
5     *{
6       margin:0px;
7       padding:0px;
8     }
9     body{
10      background-color:#ffffff;
11    }
12    #content {
13      font-family: Arial, Helvetica, sans-serif;
14      font-size: 12px;
15      line-height:18px;
16    }
17    /* .....*/
18  </style>
19 </head>
20 <body>
```

Line: 1 Column: 1 HTML Tab Size: 4

BEST PRACTICE • Linking to an External Stylesheet

Why? If you want to change your styles later, you only have to change them in one place opposed to changing them in every page of your site.



```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <title>Experience a Life-Changing Summer | 2009 Summer Project | Campus Crusade for
  Christ</title>
5 <link href="/css/screen.css" rel="stylesheet" type="text/css" /
6 </head>
7 <body id="homepage">
8 <div id="wrapper">
9 <!-- Branding -->
10 <div id="branding">
11 <h1><a href="/index.htm" title="Summer Projects">Summer Projects | Campus
  Crusade for Christ</a></h1>
12 <p id="branding_tagline"><a href="/index.htm" title="Summer Projects">Go</a></p>
13 <p id="branding_description">Campus Crusade for Christ's Summer Projects offer
  an incredible, life-changing experience that will shape your character and focus your
  life's mission. Share the gospel at a resort location in the United States, reach the
  lost in a foreign land, or stretch your faith on any number of the specially designed
  summer projects offered this year.</p>
14 <div id="passport"></div>
15 <div id="random_photo"></div>
16 </div>
```

BEGINNER'S PALETTE OF STYLE OPTIONS

- **color and background-color**
- **font-family**

This property takes a list of fonts in order of preference. If the user does not have a specified font installed, then it will go down the order until it finds one.

A common one for Web use would be:
font-family: arial, helvetica, verdana, san-serif

- **font-size**

There are several different properties you could use for this property. The best place to start would be using pixels which is a fixed font size. A typical body font-size would be 12pixels and would be written:

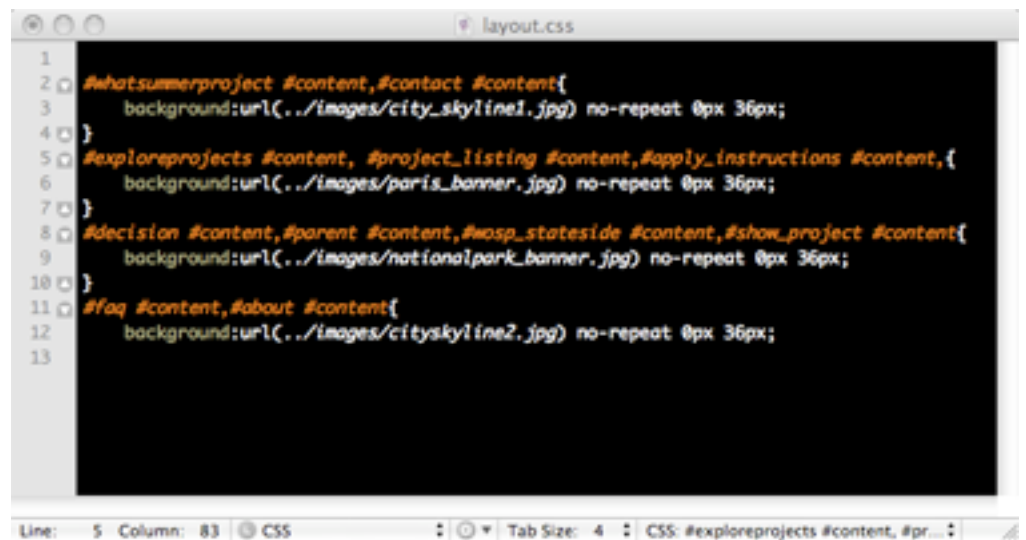
font-size: 12px;

- **font-weight:** bold or normal
- **font-style:** italic or normal
- **text-decoration:** none, underline, overline or line-through

LET'S TRY SOME STYLING Exercise

GROUPING If you want to repeat the same style with different elements on a Web page, you don't have to repeat yourself. You can group styles together by adding a comma between CSS selectors.

Example from gosummerprojects.com



```
1
2 #whatsummerproject #content,#contact #content{
3     background:url(../images/city_skyline1.jpg) no-repeat 0px 36px;
4 }
5 #exploreprojects #content, #project_listing #content,#apply_instructions #content,{
6     background:url(../images/paris_banner.jpg) no-repeat 0px 36px;
7 }
8 #decision #content,#parent #content,#asp_stateside #content,#show_project #content{
9     background:url(../images/nationalpark_banner.jpg) no-repeat 0px 36px;
10 }
11 #faq #content,#about #content{
12     background:url(../images/cityskyline2.jpg) no-repeat 0px 36px;
13 }
```

I used 4 different background images on the pages of the site. I wanted to use the same background images on several of the pages. So I grouped my CSS selectors together to make cleaner code.

WHICH RULE WINS? Determining which CSS rule wins when different rules are specified in a stylesheet depends on two factors:

- Placement in the stylesheet - later rules will overwrite rules written earlier in a stylesheet
- Specificity - a more specific CSS rule will overwrite a more general rule.

Styling Links

Class and id selectors

POSITIONING WITH CSS

Before we can use CSS for positioning our elements on the page, we need to understand the difference between block-level elements and inline elements.

Block-level element

Any element that can contain other elements. Think of it as a box or container.

Here are some examples of block-level elements:

- headings (h1, h2, h3, etc.)
- paragraphs (p)
- divs
- blockquotes
- ordered and unordered lists (ol,ul)
- list items (li)

Inline Elements

elements that sit inside other elements

Some examples:

- em
- strong
- cite
- a

Other inline elements can be nested within inline elements

Why does it matter?

The two have very different capabilities when using CSS.

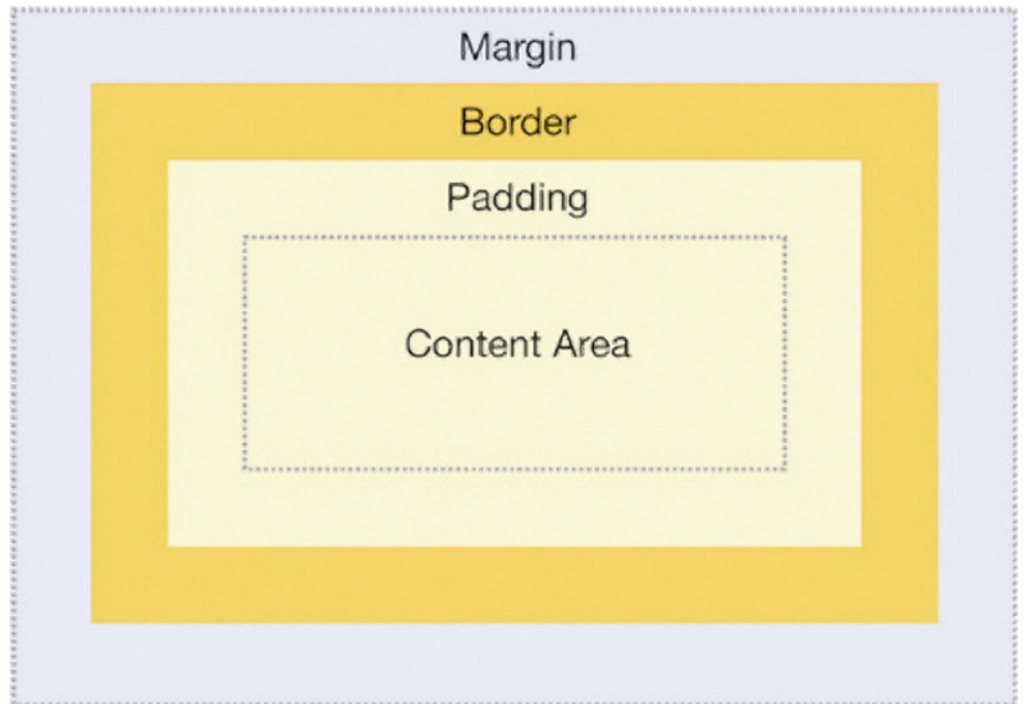
Inline elements only allow for a very limited range of styling options, which are mostly cosmetic.

WHAT YOU CAN DO WITH BLOCKS

Block-level elements can do much more:

- a fixed width or height - give it a set dimension
- can apply padding and/or margin to create space between it and other elements
- you can move a block to any location on the Web page regardless of the position that it appears in the HTML markup
- add borders

THE BOX MODEL



A block-level element can be manipulated using a combination of margin, border, padding, height and width values. This is known as the **box model**.

SO, HOW BIG IS MY BOX?

It is important to understand how padding, border, and margin effect an element as you consider how your element will interact with other elements on the page.

What is the true dimension of your box?

Let us say you have a `<div>` element.

If you set the width of the `<div>` at 100px, the padding at 15px, a border of 5px and a margin of 5px, how much width would your box take up on the page?

$\text{Content} + 2x(\text{Padding}) + 2x(\text{Border}) + 2x(\text{Margin}) = \text{Total size}$

So your `<div>` element would take up 150 pixels of space on the page. This is important to recognize especially if you are putting elements together with no extra space in between them. After five years of working with CSS, I can still at times forget this and end up with problems with my design.

LET'S LAYOUT A PAGE

Now that we understand the box model, we can begin to layout our page.

Like many things, there is more than one way to “skin a cat.”

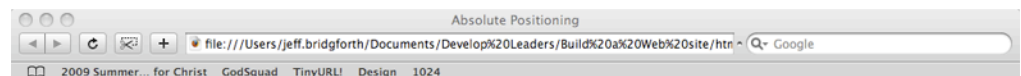
THE 3 MODELS OF POSTIONING

Absolutely

One of the easiest methods of positioning is called absolute positioning. To use absolute positioning, you specify the top and left positions or coordinates of your box.

Here is an example:

```
absolute_positioning_1.html
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5   <title>Absolute Positioning</title>
6   <style type="text/css">
7     #redblock {
8       position: absolute;
9       top: 200px;
10      left: 200px;
11      color: white;
12      background-color: red;
13      width: 90px;
14      height: 90px;
15      padding: 5px;
16    }
17  </style>
18 </head>
19
20 <body>
21   <h1>Absolute Positioning</h1>
22   <div id="redblock">This is the red block.</div>
23   <p>The red block is positioned 200 pixels from the top and 200
24     pixels from the left.</p>
25 </body>
26 </html>
```



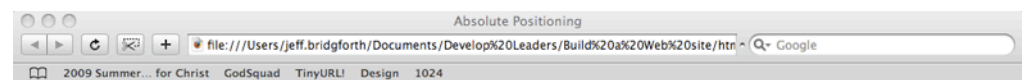
Absolute Positioning

The red block is positioned 200 pixels from the top and 200 pixels from the left.



In this example, the element was absolutely positioned in relation to the viewport of the browser. A more advanced method allows you to absolutely position elements inside other elements of the page. The code below would position a yellow box in relation to the red box that contains it.

```
absolute_positioning_2.html
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>Absolute Positioning</title>
6 <style type="text/css">
7   #redblock {
8     position: absolute;
9     top: 200px;
10    left: 200px;
11    background-color: red;
12    width: 100px;
13    height: 100px;
14  }
15  #yellowblock {
16    position: absolute;
17    top: 20px;
18    left: 20px;
19    background-color: yellow;
20    color: red;
21    width: 50px;
22    height: 50px;
23    padding: 5px;
24  }
25 </style>
26 </head>
27 <body>
28 <h1>Absolute Positioning</h1>
29 <div id="redblock">
30   <div id="yellowblock">Yellow!</div>
31 </div>
32 <p>The red block is positioned absolutely using coordinates of
33   200 pixels from the top, and 200 pixels from the left.</p>
34 <p>The yellow block inside is positioned 20 pixels from the
35   top of its parent element, and 20 pixels to the left.</p>
36 </body>
37 </html>
```



Absolute Positioning

The red block is positioned absolutely using coordinates of 200 pixels from the top, and 200 pixels from the left.

The yellow block inside is positioned 20 pixels from the top of its parent element, and 20 pixels to the left.



IT'S ALL RELATIVE

A second positioning method is called relative positioning. A value you apply to an element will be applied relative to the position it was in before the positioning rule was applied or relative to other elements on the page.

Advantages of using relative positioning over absolute:

- More flexibility in how the elements position, especially if you have elements of the page that you do not know how much height they will have or the height will change over multiple pages due to difference in content length
- Relative positioned elements stay in the "flow" of a document, where absolute positioning takes the element out of the flow.

The disadvantage is that relative positioning is not quite as easy as absolute and can take a bit more thought and trial and error to pull off what you want to accomplish.

PRACTICE, PRACTICE, PRACTICE

In practice, I use a combination of all three positioning methods to pull off the layouts that I want to achieve.

ROOT BEER FLOAT, ANYONE?

The last positioning method is called floating. This is a method favored by many developing and designing for the Web. When you float an element, you move it to one side and allow the other content to flow around it.

A similar method in print design is text wrap.

One place that I use floating often is with images.

About Me

The defining thing about my life is my Christian faith. It shapes all that I do from family to design to choice of career. I joined the staff of [Campus Crusade for Christ](#) after I graduated from the [University of Kansas](#).






I started this blog to share helpful resources, software and other Web sites that I have benefitted from in my personal and professional life. Most of my writing is related to Web design and development. I also created the site to be a "sandbox" so that I could play around with different design






```
.imgright {  
    float:right;  
    padding-bottom:15px;  
    padding-left:15px;  
}
```

Example from gosummerprojects.com

PROMOTIONAL VIDEOS

| Hooray | WOSP: Stateside | WOSP: International |
|---|--|---|
| Viral Video Promo | Inspirational and Informative | Inspirational and Informative |
|  |  |  |
| Watch Now | Watch Now | Watch Now |
| Download (200 MB) | Download (28.4 MB) | Download (37.6 MB) |
| iPod/iPhone Version (41.7 MB) | iPod/iPhone Version (10.2 MB) | iPod/iPhone Version (13.1 MB) |

Spotlight Videos

| Why Go? | Community | Outreach | Jobs | A Day in Tokyo |
|---|---|--|---|---|
|  |  |  |  |  |
| Watch Now | Watch Now | Watch Now | Watch Now | Watch Now |
| Download (16.74 MB) | Download (15.52 MB) | Download (17.74 MB) | Download (18.08 MB) | Download (31.9 MB) |
| iPod/iPhone Version (6.38 MB) | iPod/iPhone Version (4.53 MB) | iPod/iPhone Version (5.85 MB) | iPod/iPhone Version (5.92 MB) | iPod/iPhone Version (10.99 MB) |

Each of these elements are floated elements:

```
#content_videos li {  
    display:block;  
    float:left;  
    margin-right:18px;  
    width:222px;  
}
```

WARNING, DANGER AHEAD One word of caution

Floating elements can be tricky. At some point in your layout, you may need to “clear” a floated element in order to achieve the layout you want. This takes some experience and practice and it is an area that I still run into problems with.



```
#left_column{
  float:left;
  width:150px;
}

#right_column{
  float:right;
  width: 300px;
}

#footer{
  clear:both;
  width:100%;
}
```

Floating is one of the things you can get in trouble with in relation to the box model. If the containing block is too narrow for all the floated elements, the remaining floats will drop down until there is sufficient space for the element to occupy.

RESOURCES TO HELP YOU GO FURTHER

Build Your Own Web Site the Right Way Using HTML and CSS by Ian Lloyd. SitePoint Books.

This is the book that I used to put together this training. This would be a good place to start with books.

SitePoint's HTML and CSS Online References

<http://reference.sitepoint.com/>

CSS Mastery by Andy Budd - when you want to take your CSS beyond a basic understanding

Software I use in Web development:

- **Coda (Mac)** - This software helps me to do everything in my workflow. It includes an code editor, FTP client and keeps track of changes in my local files and flags those files to be uploaded to the production server.
- **Fireworks** - My graphics creator of choice.
- **Photoshop** - I use Photoshop as my image editor and then move into Fireworks.
- **Firefox Web browser**
- **Firebug Extension for Firefox** - I use this plug in everyday. It helps me to determine where I have problems in my CSS or to pinpoint what elements I need to style in a design.
- **Transmit** - FTP client by the same makers of Coda. I use this when I have really big files to transfer or have a large quantity of files to transfer.

YOUR INSTRUCTOR



Jeff Bridgforth has been designing professionally for the Web for the past 8 years. He built his first Web site 11 years ago. Jeff has been building Web sites using HTML and CSS for the past five years.

This training was put together using Ian Lloyd's book, *Build Your Own Web Site the Right Way Using HTML and CSS*.

Where you can find Jeff on the Web

- <http://jeffbridgforth.com>
- <http://twitter.com/webcraftsman>
- <http://flickr.com/photos/bridgforth>
- <http://del.icio.us/JBridg4th>