



Web Development

Week 4: More CSS



Overview

- ↗ Revision – CSS Syntax
- ↗ Files, Folders and Paths
- ↗ Inline Blocks
- ↗ Box Model
- ↗ Margin and Padding
- ↗ Grouping and CSS Selectors
- ↗ Positioning
- ↗ Navigation Menu - CSS

Revision:: Syntax of CSS rules

➤ `p { color: green; }`

➤ Each CSS style declaration is made up of a property and a value.

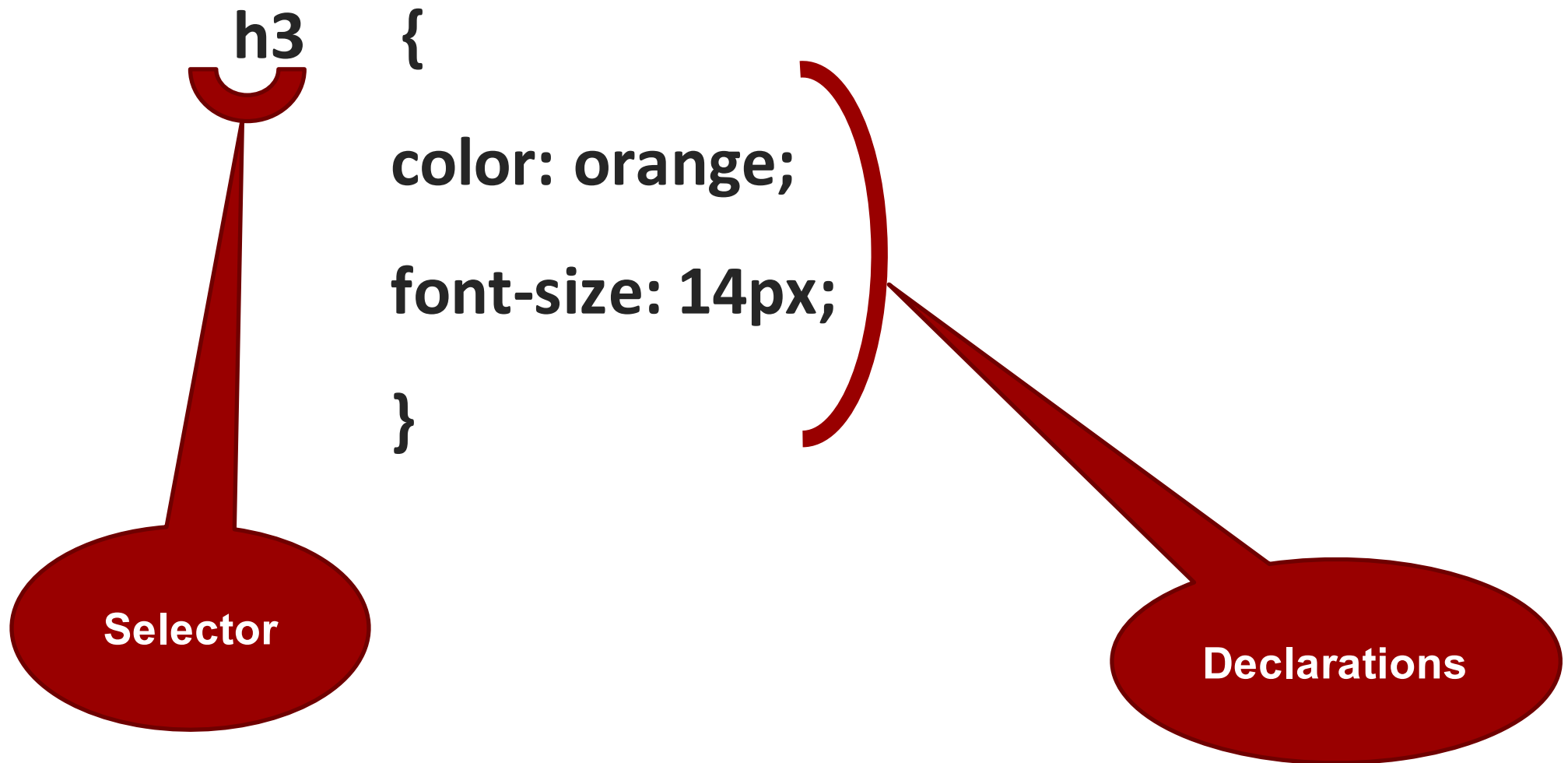
➤ The property and the value are separated by a colon (:).

➤ Each declaration must be followed with a semi-colon (;).

➤ This follows the format of a name: value pair.

➤ The property (name) identifies the feature of the element that requires styling, the value describes how this will display on the page.

Revision:: Syntax of CSS rules



Revision:: Id Selector

- The id selector can be added to an element so that the element can be uniquely identified on a page. An attribute called id is added to the elements opening tag and is given a meaningful name. For a given page, there should only be one element with a given name on the page. Eg.

id attribute

Unique Name

```
1 <p id="example">
2 This is an example
3 </p>
```

For a given id attribute here should only be one element with a name of example on the page.

Revision:: Id Selector

- When an element has been given an id attribute a CSS rule can be written for this. The name given to the id must be unique to the page.
- In the stylesheet, a hash symbol (#) must be added before the ID when you use it in a selector. (ie. Creating the CSS rule)

<pre>1 <p id="example"> 2 This is an example 3 </p></pre>	HTML ⚙	<pre>1 #example { 2 color: green; 3 font-size: 25px; 4 }</pre>	CSS ⚙
1	JAVASCRIPT ⚙	This is an example	

Revision:: Class Selector

- When an element has been given a class attribute a CSS rule can be written for this. The name given to the class does not have to be unique to the page. There can be multiple elements with the same class attribute on the same page. This can be used to style multiple occurrences of the same element on the page (eg. Multiple headings or paragraphs)
- In the stylesheet, a period symbol (.) must be added before the Class when you use it in a selector. (ie. Creating the CSS rule)

Revision:: Class Selector

↗ Example:

<pre>1 <p class="details"> 2 This is an example 3 </p> 4 <p class="details"> 5 This is an example 6 </p></pre> <p>HTML ⚙️</p>	<pre>1 .details { 2 color: blue; 3 font-size: 18px; 4 }</pre> <p>CSS ⚙️</p>
<pre>1</pre> <p>JAVASCRIPT ⚙️</p>	<p>This is an example</p> <p>This is an example</p>

Absolute Path

- An absolute path gives the full file path, including the domain name.
- Eg. <http://www.amitycode.com/img/about-bg.jpg>

Relative Path

➤ Relative paths do not use the full path or the domain name.

➤ Eg.

➤ Index.html

➤ /img/pict.jpg

➤ Img/pict.jpg

	Description	Example
/	Root Directory	/img/pict.jpg
../	Back one directory	../img/pict.jpg
../../	Back two directories	../../img/pict.jpg
Img/pict.jpg	For the current folder were in, there is an Img folder containing a file pict.jpg	

Files, Folders and Paths

- When creating a web application it is important to structure the project folder.
- Similar content items should be grouped together in folders.
- Eg: Images, CSS files, JavaScript files, videos etc.....

Sample Project Folder Structure

➤ MyProjectFolder

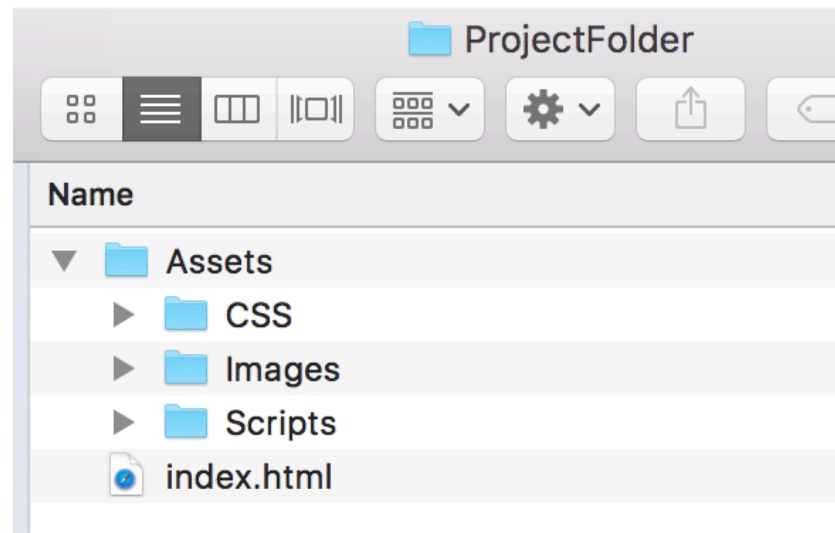
➤ Assets

➤ CSS

➤ Images

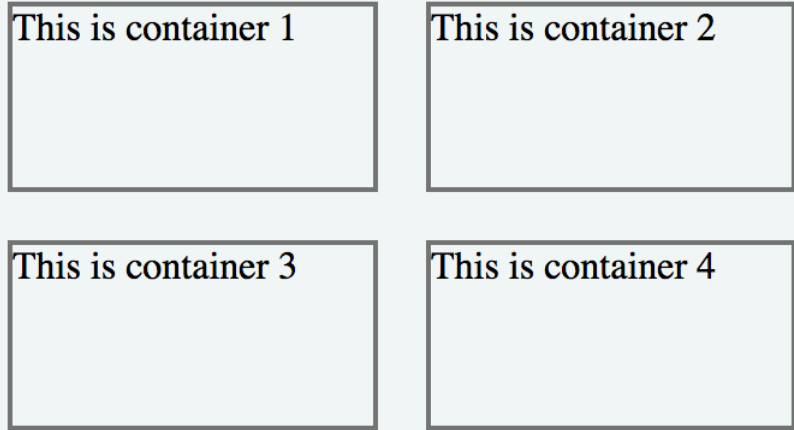
➤ Scripts

➤ .html files



In Class Example

Inline Blocks – Old way using float

<pre>1 <div class="genericContainer">This is container 1</div> 2 <div class="genericContainer">This is container 2</div> 3 <div class="genericContainer">This is container 3</div> 4 <div class="genericContainer">This is container 4</div> 5</pre> <p>HTML ⚙️</p>	<pre>1 .genericContainer { 2 float: left; 3 width: 150px; 4 height: 75px; 5 margin: 10px; 6 border: 2px solid #777777; 7 }</pre>
<pre>1</pre> <p>JAVASCRIPT ⚙️</p>	

Inline Blocks – New Way

```
1 <div class="genericContainer">This is container 1</div> HTML ⚙️  
2 <div class="genericContainer">This is container 2</div>  
3 <div class="genericContainer">This is container 3</div>  
4 <div class="genericContainer">This is container 4</div>  
5
```

JAVASCRIPT ⚙️

```
1 .genericContainer {  
2   display: inline-block;  
3   width: 150px;  
4   height: 75px;  
5   margin: 10px;  
6   border: 2px solid #777777;  
7 }
```

This is container 1

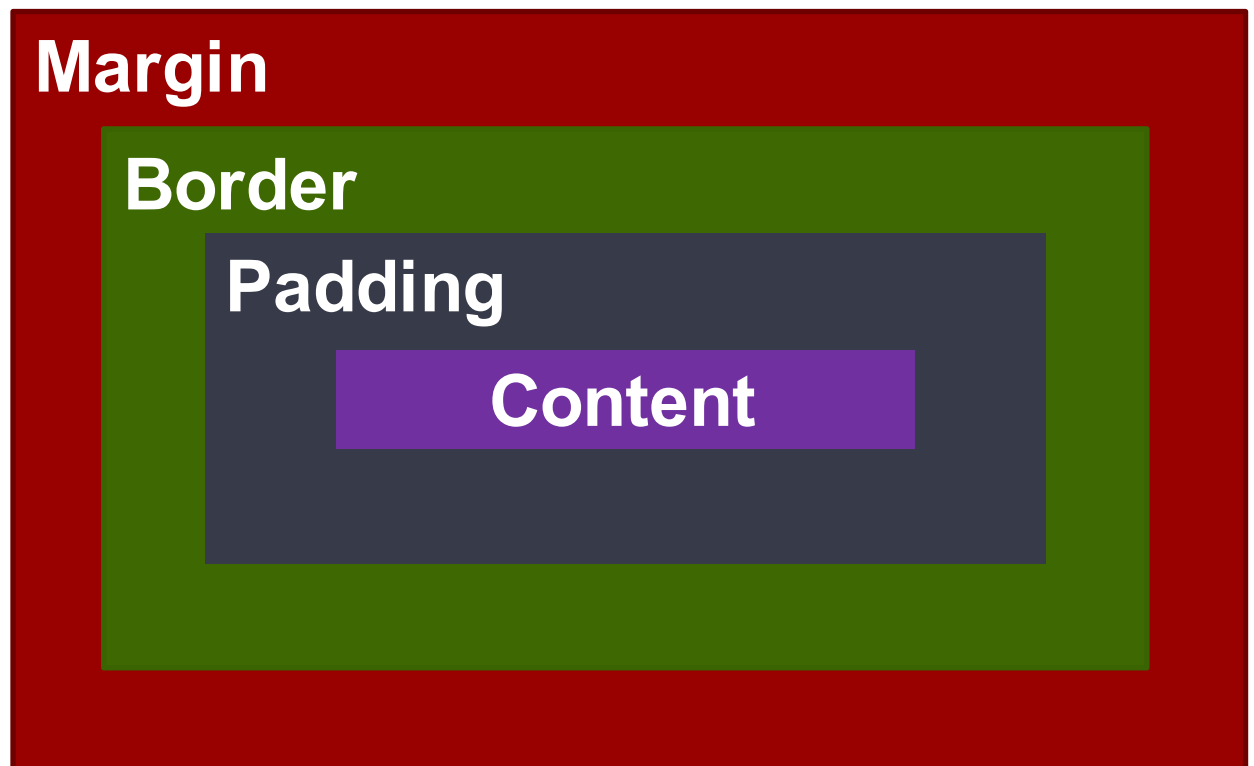
This is container 2

This is container 3

This is container 4

Box Model

All elements added to the page can be referred to as boxes. Each box on the page can be styled on the page. Each box has a margin, a border, padding and its content. This is referred to as the box model.



Box Model

↗ Content

- ↗ The content is what displays on the page.
- ↗ This can be thought of as the content of the box!!

↗ Padding

- ↗ Padding adds some area around the content in the box.
- ↗ The padding will be transparent.

Box Model

↗ Border

↗ The border goes around the padding and the content.

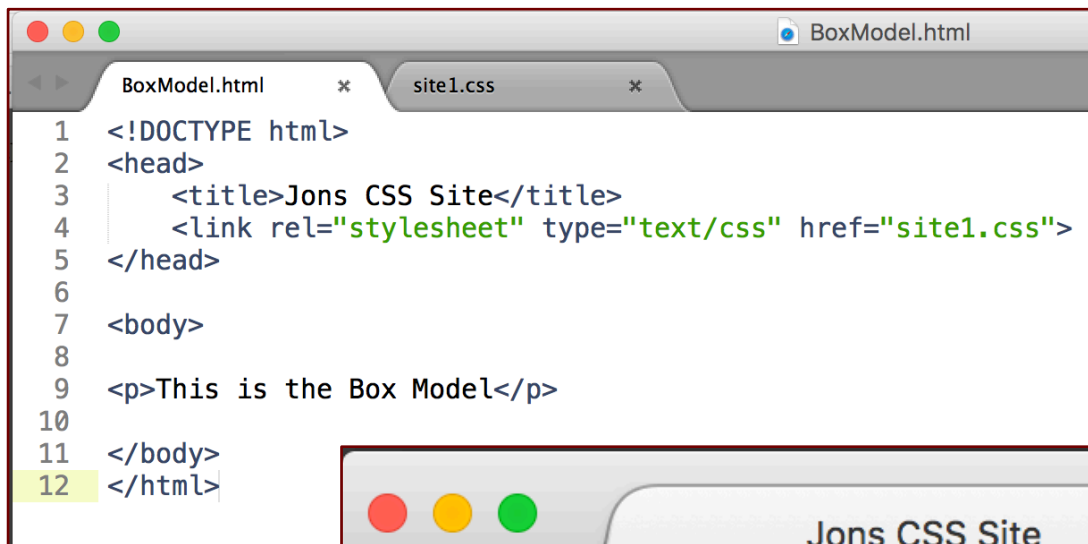
↗ Margin

↗ Margin adds some area outside the border.

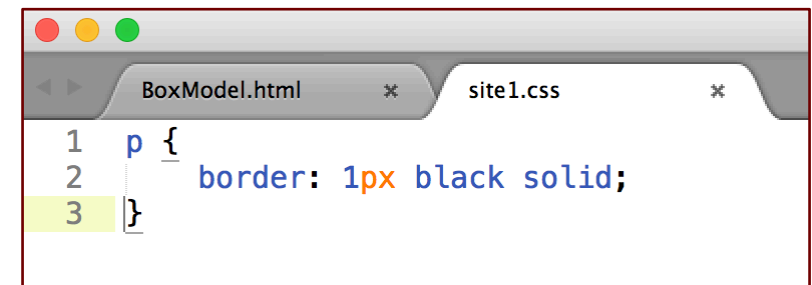
↗ The margin will be transparent.

Box Model-> Example

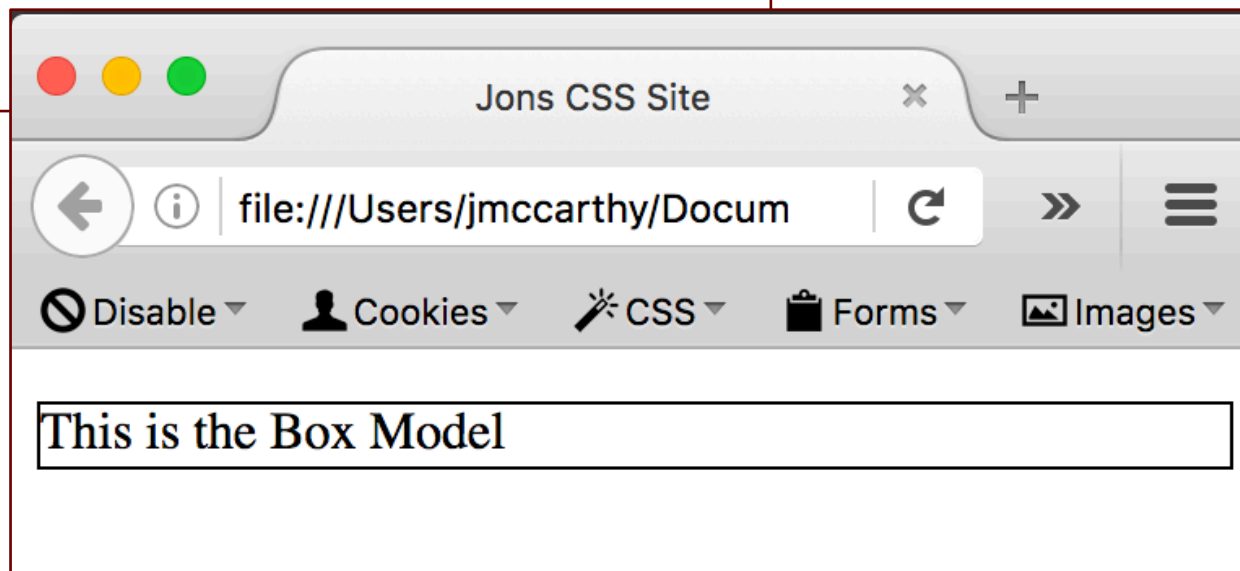
➤ Step 1. Add content to page and set a border.



```
BoxModel.html
1 <!DOCTYPE html>
2 <head>
3   <title>Jons CSS Site</title>
4   <link rel="stylesheet" type="text/css" href="site1.css">
5 </head>
6
7 <body>
8
9 <p>This is the Box Model</p>
10
11 </body>
12 </html>
```

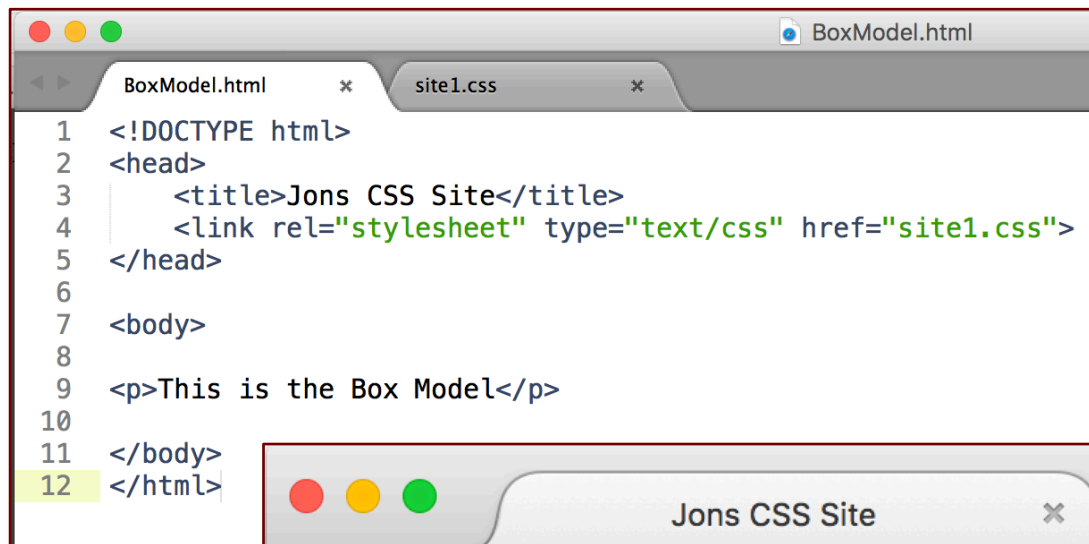


```
BoxModel.html
1 p {
2   border: 1px black solid;
3 }
```

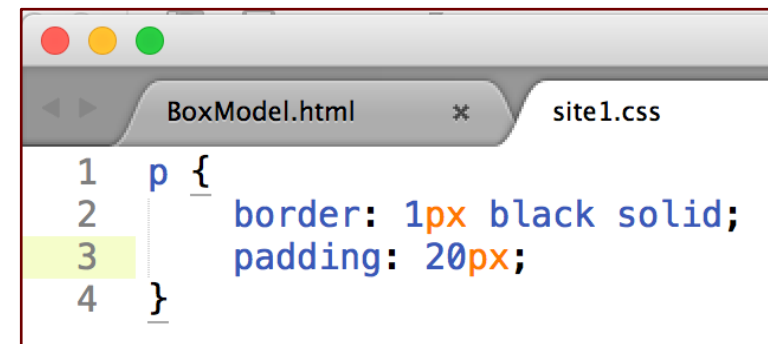


Box Model-> Example

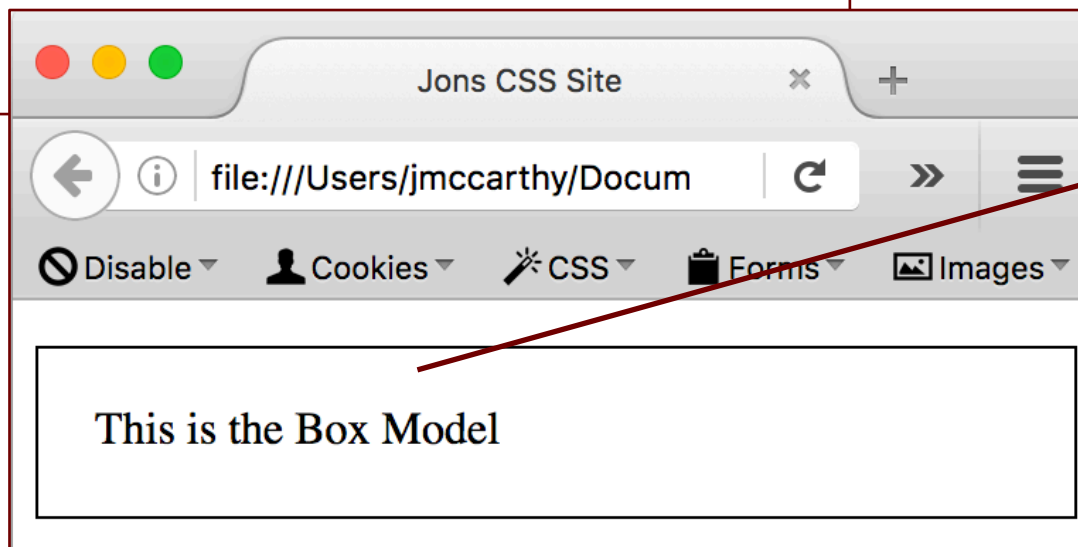
➤ Step 2. Add padding to CSS.



```
1 <!DOCTYPE html>
2 <head>
3   <title>Jons CSS Site</title>
4   <link rel="stylesheet" type="text/css" href="site1.css">
5 </head>
6
7 <body>
8
9 <p>This is the Box Model</p>
10
11 </body>
12 </html>
```



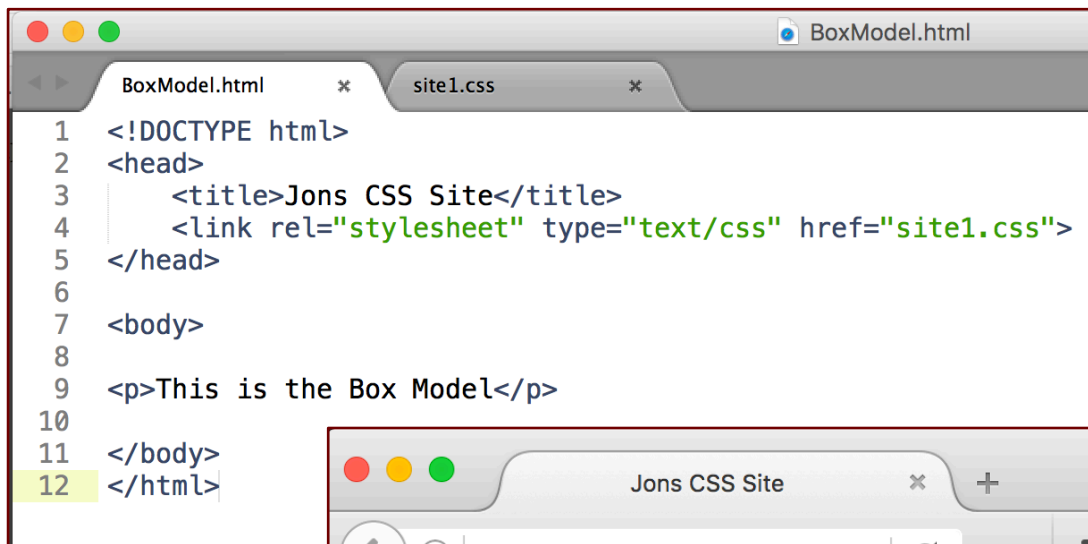
```
1 p {
2   border: 1px black solid;
3   padding: 20px;
4 }
```



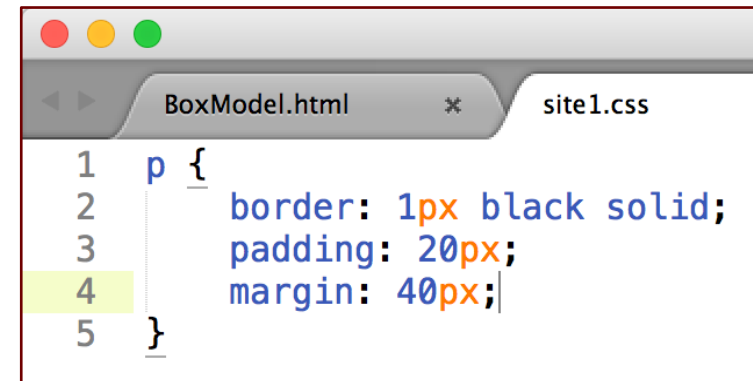
Padding space
around the content

Box Model-> Example

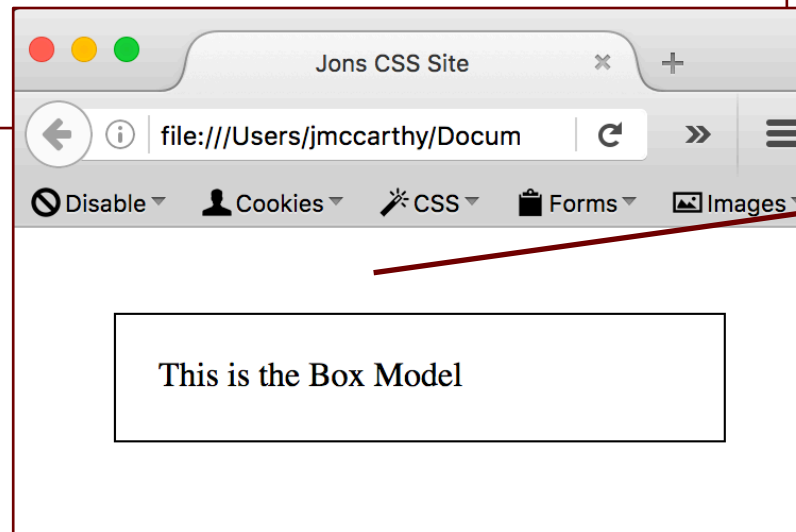
➤ Step 2. Add margin to CSS.



```
1 <!DOCTYPE html>
2 <head>
3   <title>Jons CSS Site</title>
4   <link rel="stylesheet" type="text/css" href="site1.css">
5 </head>
6
7 <body>
8
9 <p>This is the Box Model</p>
10
11 </body>
12 </html>
```



```
1 p {
2   border: 1px black solid;
3   padding: 20px;
4   margin: 40px;
5 }
```



Margin space
around the border

Padding

- Padding adds space around the content of a given element.
- This can be thought of as space between the content and border for a given element.
- Using the padding property puts the space on all four sides of the element. To just put padding space on one side use:
 - padding-top
 - padding-right
 - padding-bottom
 - padding-left

Margin

- ↗ The margin adds space outside the border of a given element.
- ↗ This can be thought of as space between the border and the other elements on the page.
- ↗ Using the margin property puts the space on all four sides of the element. To just put padding space on one side use:
 - ↗ `margin-top`
 - ↗ `margin-right`
 - ↗ `margin-bottom`
 - ↗ `margin-left`

Grouping

```

1  <!DOCTYPE html>
2  <head>
3      <title>Jons CSS</title>
4      <link rel="stylesheet" type="text/css" href="site1.css">
5  </head>
6
7  <body>
8  <p>This is a heading</p>
9  <span id="spanHeading">This is another heading</span>
10 <br>
11 <label id="labelHeading">This is another heading</label>
12
13 </body>
14 </html>

```

This is a heading

This is another heading

This is another heading

```

1  p {
2      color: red;
3      font-size: 20px;
4      font-style: italic;
5  }
6
7  #spanHeading {
8      color: red;
9      font-size: 20px;
10     font-style: italic;
11 }
12
13 #labelHeading {
14     color: red;
15     font-size: 20px;
16     font-style: italic;
17 }

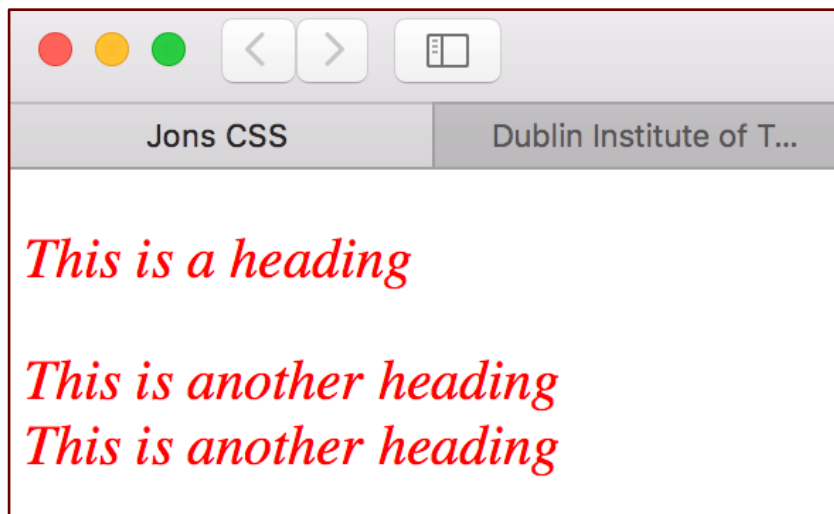
```

3 CSS rules with
the same styling

Grouping

```
1 <!DOCTYPE html>
2 <head>
3   <title>Jons CSS</title>
4   <link rel="stylesheet" type="text/css" href="site1.css">
5 </head>
6
7 <body>
8 <p>This is a heading</p>
9 <span id="spanHeading">This is another heading</span>
10 <br>
11 <label id="labelHeading">This is another heading</label>
12
13 </body>
14 </html>
```

```
1 p, #spanHeading, #labelHeading {
2   color: red;
3   font-size: 20px;
4   font-style: italic;
5 }
6
```



1 CSS rule for multiple elements with the same styling. We can separate selectors with commas in one line and apply the same properties to them all.

CSS Combinators

- ↗ A CSS selector can contain more than one simple selector.
- ↗ There are four different combinators in CSS3:
 - ↗ descendant selector (space)
 - ↗ child selector (>)
 - ↗ adjacent sibling selector (+)
 - ↗ general sibling selector (~)

Descendant Selector

- The descendant selector will match all descendant elements for a given element.
- Eg. If there were paragraphs inside a section element, a rule can be written to style all paragraphs inside a section tag.

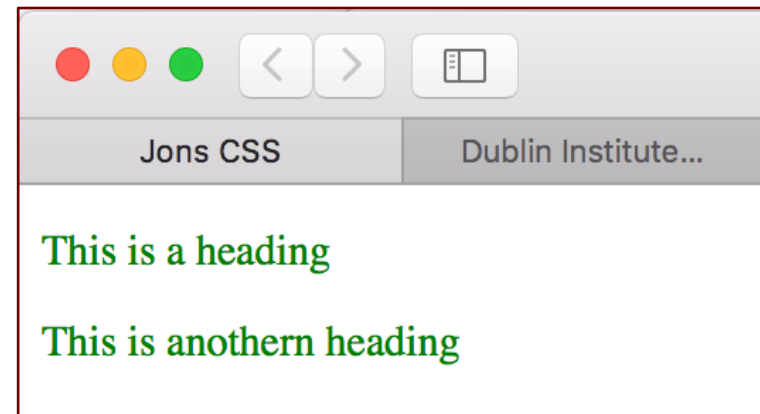
Descendant Selector - Example



```
1 <!DOCTYPE html>
2 <head>
3   <title>Jons CSS</title>
4   <link rel="stylesheet" type="
5 </head>
6
7 <body>
8 <section>
9 <p>This is a heading</p>
10 <p>This is another heading</p>
11 </section>
12
13 </body>
14 </html>
```

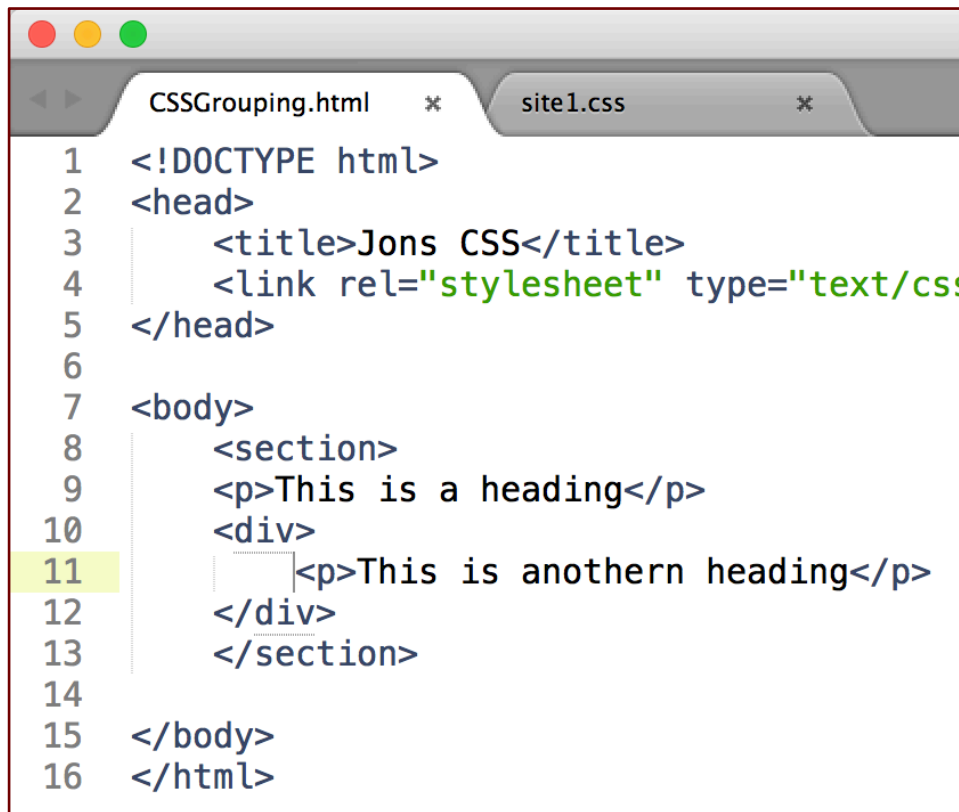


```
1 section p {
2   color: green;
3   font-size: 15px;
4 }
5
```



Child Selector

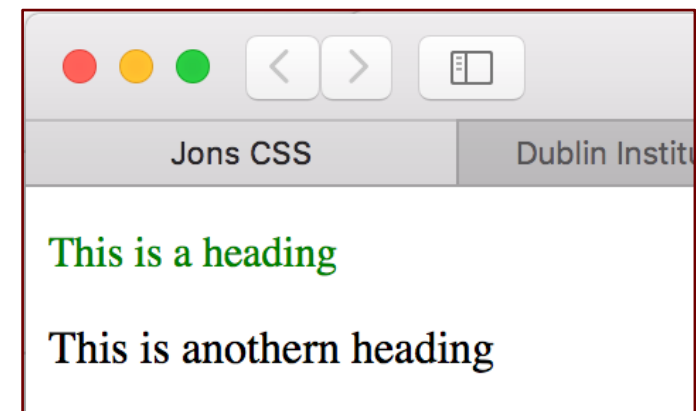
➤ The child selector selects all elements that are the immediate children of a specified element.



```
1 <!DOCTYPE html>
2 <head>
3   <title>Jons CSS</title>
4   <link rel="stylesheet" type="text/css" href="site1.css">
5 </head>
6
7 <body>
8   <section>
9     <p>This is a heading</p>
10    <div>
11      <p>This is another heading</p>
12    </div>
13  </section>
14
15 </body>
16 </html>
```



```
1 section > p {
2   color: green;
3   font-size: 15px;
4 }
5
```



Max and Min Width and Height

- The min-height property can be used to set the minimum height of an element.
- The max-height property can be used to set the maximum height of an element.
- The min-width property can be used to set the minimum width of an element.
- The max-width property can be used to set the maximum width of an element.

Span and Label

↗

- ↗ The element can be used to embed content inline and to style its content.
- ↗ The span tag offers no initial styling.

↗ <label>

- ↗ The <label> tag defines a label for an <input> element.
- ↗ The <label> element does not offer any initial styling.

Inline Blocks – New Way

```
1 <div class="genericContainer">This is container 1</div> HTML ⚙️
2 <div class="genericContainer">This is container 2</div>
3 <div class="genericContainer">This is container 3</div>
4 <div class="genericContainer">This is container 4</div>
5
```

JAVASCRIPT ⚙️

```
1 .genericContainer {
2     display: inline-block;
3     width: 150px;
4     height: 75px;
5     margin: 10px;
6     border: 2px solid #777777;
7 }
```

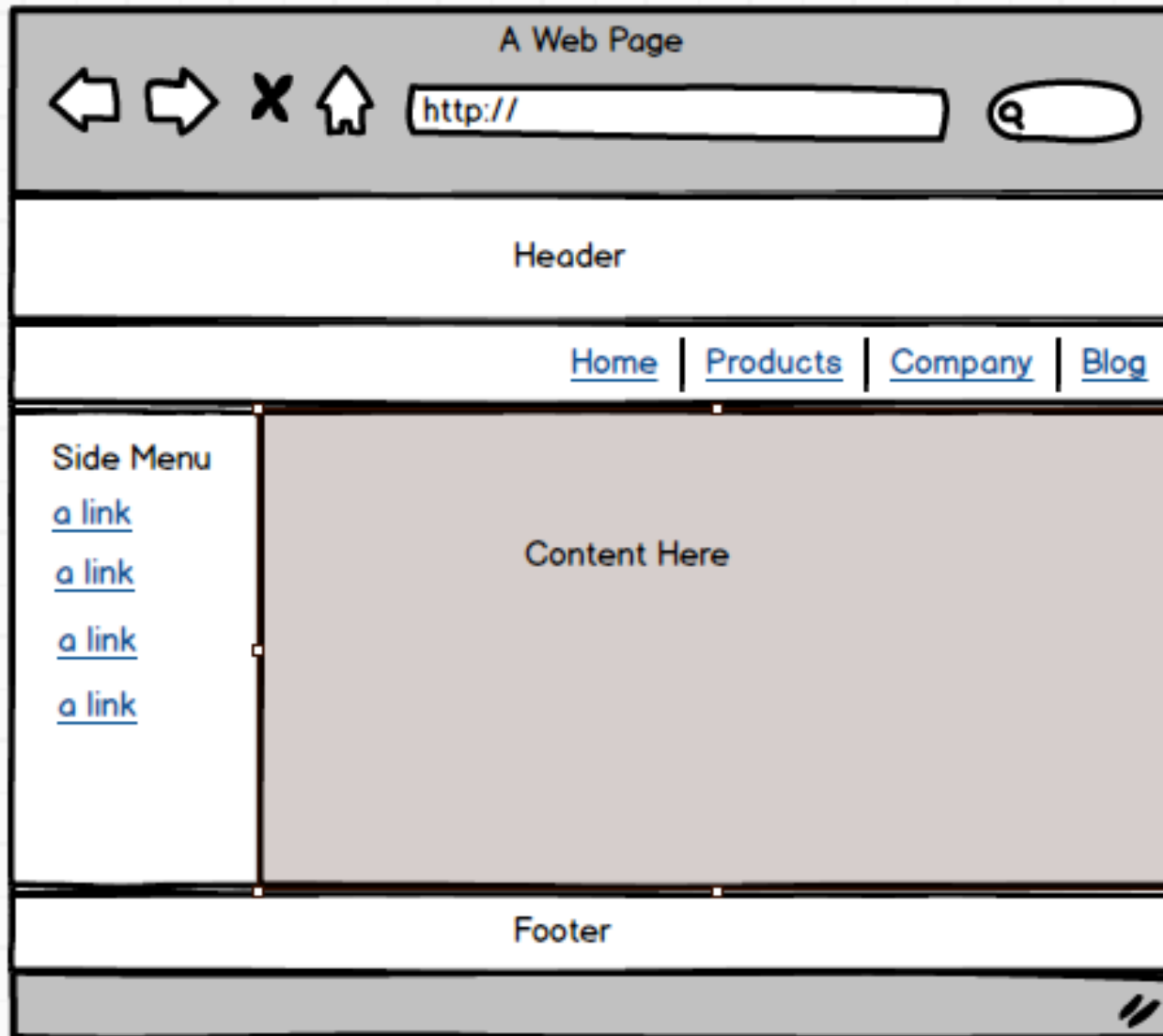
This is container 1

This is container 2

This is container 3

This is container 4

Page Layout with CSS



How can we achieve this with HTML and CSS??

In-Class:: Page Layout with CSS

```

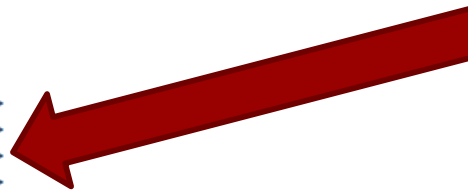
1  <!DOCTYPE html>
2  <head>
3      <title>Jons CSS Site</title>
4      <link rel="stylesheet" type="text/css" href="site.css">
5  </head>
6
7  <body>
8      <div id="main">
9          <header>
10             <h1>Jon's CSS Website</h1>
11             <hr>
12             <nav>
13                 <a href="index.html">Home</a> |
14                 <a href="about.html">About</a> |
15                 <a href="examples.html">Examples</a>
16             </nav>
17         </header>
18
19         <section id="content">
20             <section id="menuSection">
21                 <h2>Menu</h2>
22                 <a href="#">Link 1</a> <br>
23                 <a href="#">Link 2</a> <br>
24                 <a href="#">Link 3</a> <br>
25                 <a href="#">Link 4</a> <br>
26                 <a href="#">Link 5</a> <br>
27                 <a href="#">Link 6</a>
28             </section>
29             <section id="homeSection">
30                 <h2>Home Page</h2>
31                 <p>
32                     Lorem ipsum .....
33                 </p>
34             </section>
35         </section>
36     </div>
37
38     <footer>
39         Site by: Jon McCarthy &copy; 2016
40     </footer>
41
42 </body>
43 </html>

```

```

38 #menuSection a {
39     color: grey;
40 }
41
42 #menuSection {
43     vertical-align: top;
44     display: inline-block;
45     width: 18%;
46     padding: 20px;
47 }
48
49 #homeSection {
50     display: inline-block;
51     width: 70%;
52     padding: 20px;
53 }
54

```



Borders

➤ The border-style property adds a border for a given container.

➤ There are a number of different types of borders:

➤ dotted

➤ dashed

➤ solid

➤ double

➤ none

➤ Hidden

Eg:

```
p { border-style: dotted; }
```

```
p { border-style: solid; }
```

➤ See the W3C or the W3Schools for more details and other types of borders.

CSS Positioning

- ↗ The position property specifies the type of positioning for a given element
 - ↗ Static
 - ↗ Relative
 - ↗ fixed
 - ↗ absolute

Static Positioning

- ↗ All elements on the page are static by default.
- ↗ Static positioned elements are not affected by the top, bottom, left, and right properties.
- ↗ The elements position will go with the normal flow of the page.

Fixed Positioning

- If an element is set with a fixed position, it will stay in the same place, even when the page is scrolled.
- The fixed element does not leave a gap in the page, it just displays where it is told to!!

Fixed Positioning Example

```

21 <body>
22   <header id="fixedHeader">
23     <h1>Jon's CSS Website</h1>
24   </header>
25   <div id="main">
26     Loads of Lorem ipsum here!!
27   </div>
28 </body>

```

```

5   #fixedHeader {
6     position: fixed;
7     top: 0px;
8     left: 0px;
9     background-color: yellow;
10    height: 60px;
11    width: 100%;
12    text-align: center;
13  }
14
15  #main {
16    margin-top: 60px;
17  }

```

Jon's CSS Website

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer viverra purus
 eget risus varius porta. Aliquam hendrerit maximus eros, quis mattis sem
 condimentum a. Etiam venenatis ornare accumsan. Class aptent taciti sociosqu
 ad litora torquent per conubia nostra, per inceptos himenaeos. Morbi facilisis
 dui sit amet nibh fringilla lacinia. Etiam vulputate, lacus ac bibendum
 dignissim, nulla mi ornare ipsum, ac ultricies ex ante et mauris. Aliquam id
 ornare justo, a placerat odio. Quisque in sem id nibh sagittis maximus. Nunc
 orci enim, ultricies ultrices sapien sodales, vestibulum consequat elit. Cras
 dapibus enim vel tellus volutpat, vitae pulvinar dolor vulputate. Praesent
 consectetur sed tortor id ultricies. Sed semper lacus ac elit laoreet, eget
 rhoncus risus porttitor. Morbi ac lectus ultrices, lacinia ligula id, iaculis dolor.
 Fusce purus eros, suscipit in ornare nec, rhoncus ac ipsum. Pellentesque
 habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas

If you scroll the header
shouldn't move!!

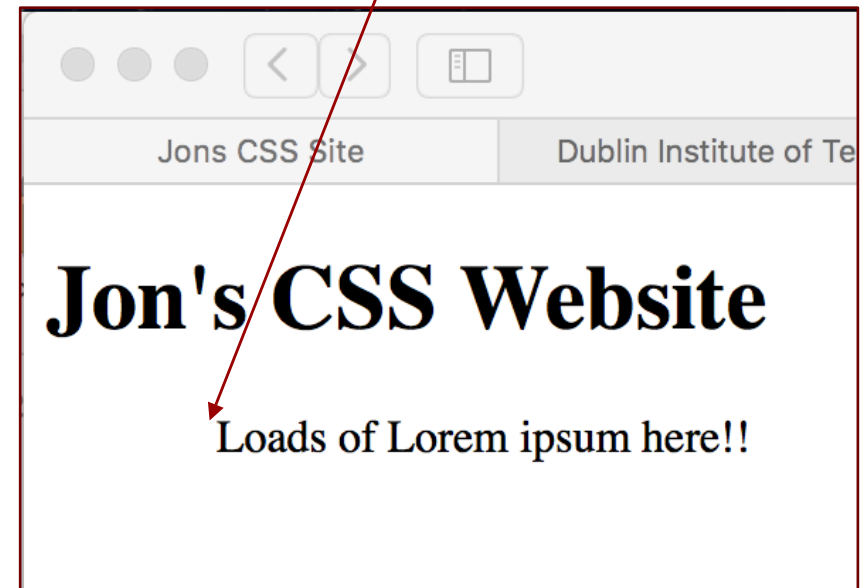
Relative Positioning

- An element following relative positioning will be positioned on the page relative to its normal position.
- Using top, right, bottom, and left properties will cause it to be adjusted away from its normal position on the page.
- This can also be described as changing the position of the element relative to itself. I.e. Where it was meant to be on the page.

Relative Positioning Example

```
12 <body>
13   <header id="fixedHeader">
14     <h1>Jon's CSS Website</h1>
15   </header>
16   <div id="main">
17     Loads of Lorem ipsum here!!
18   </div>
19 </body>
```

```
5   #main {
6     position: relative;
7     left: 60px;
8   }
```



Absolute Positioning

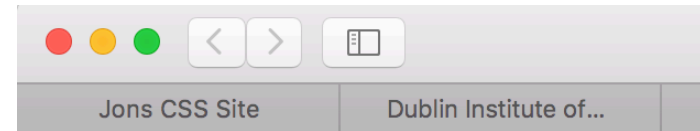
- An element with absolute position is positioned relative to the nearest positioned ancestor.

Absolute Positioning Example

```

23 <body>
24   <header id="fixedHeader">
25     <h1>Jon's CSS Website</h1>
26   </header>
27   <div id="main">
28     <div id="container">Loads of Lorem ipsum here!!</div>
29   </div>
30 </body>
31

```

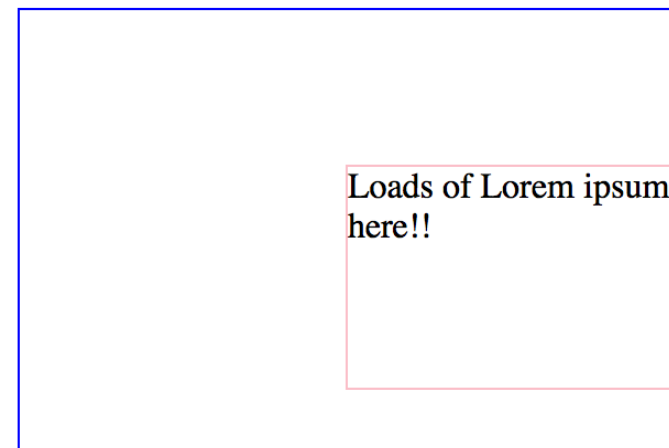


```

5   #main {
6     border: 1px blue solid;
7     position: relative;
8     width: 300px;
9     height: 200px;
10  }
11
12  #container {
13    position: absolute;
14    border: 1px pink solid;
15    width: 150px;
16    height: 100px;
17    right: 0px;
18    top: 70px;
19  }

```

Jon's CSS Website

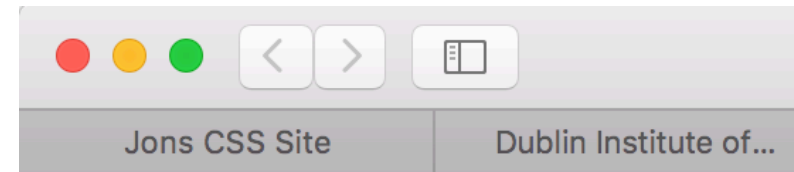


Navigation Bar

```
34 <nav>
35   <ul>
36     <li><a href="#">Home</a></li>
37     <li><a href="#">Products</a></li>
38     <li><a href="#">Contact</a></li>
39     <li><a href="#">About</a></li>
40   </ul>
41 </nav>
```

Navigation Bar

```
5  ul {  
6      list-style: none;  
7      margin: 0;  
8      padding: 0;  
9      background-color: #333;  
10 }
```



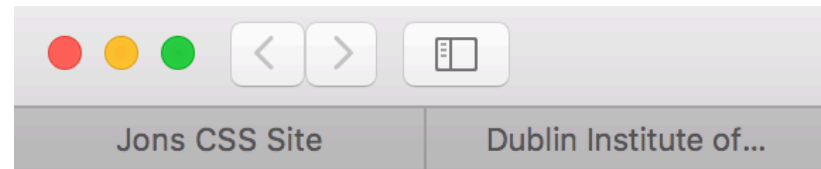
Jon's CSS Website

[Home](#)
[Products](#)
[Contact](#)
[About](#)

Loads of Lorem ipsum here!!

Navigation Bar

```
5 ▼ ul {  
6     list-style: none;  
7     margin: 0;  
8     padding: 0;  
9     background-color: #333;  
10 }  
11  
12 li {  
13     display: inline-block;  
14 }  
15
```



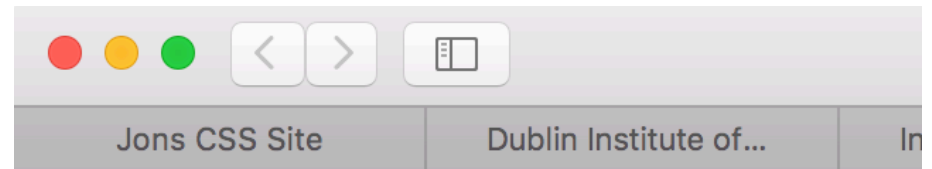
Jon's CSS Website

[Home](#) [Products](#) [Contact](#) [About](#)

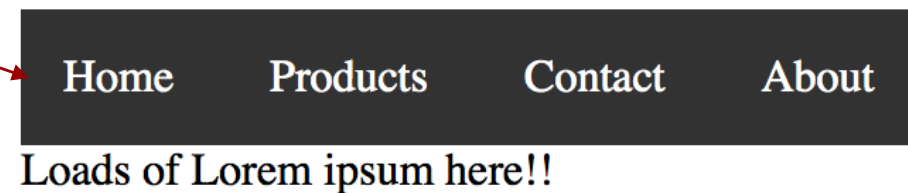
Loads of Lorem ipsum here!!

Navigation Bar

```
5  ul {
6      list-style: none;
7      margin: 0;
8      padding: 0;
9      background-color: #333;
10 }
11
12 li {
13     display: inline-block;
14 }
15
16 li a {
17     display: block;
18     color: white;
19     text-align: center;
20     padding: 15px 15px;
21     text-decoration: none;
22 }
```



Jon's CSS Website



Questions

