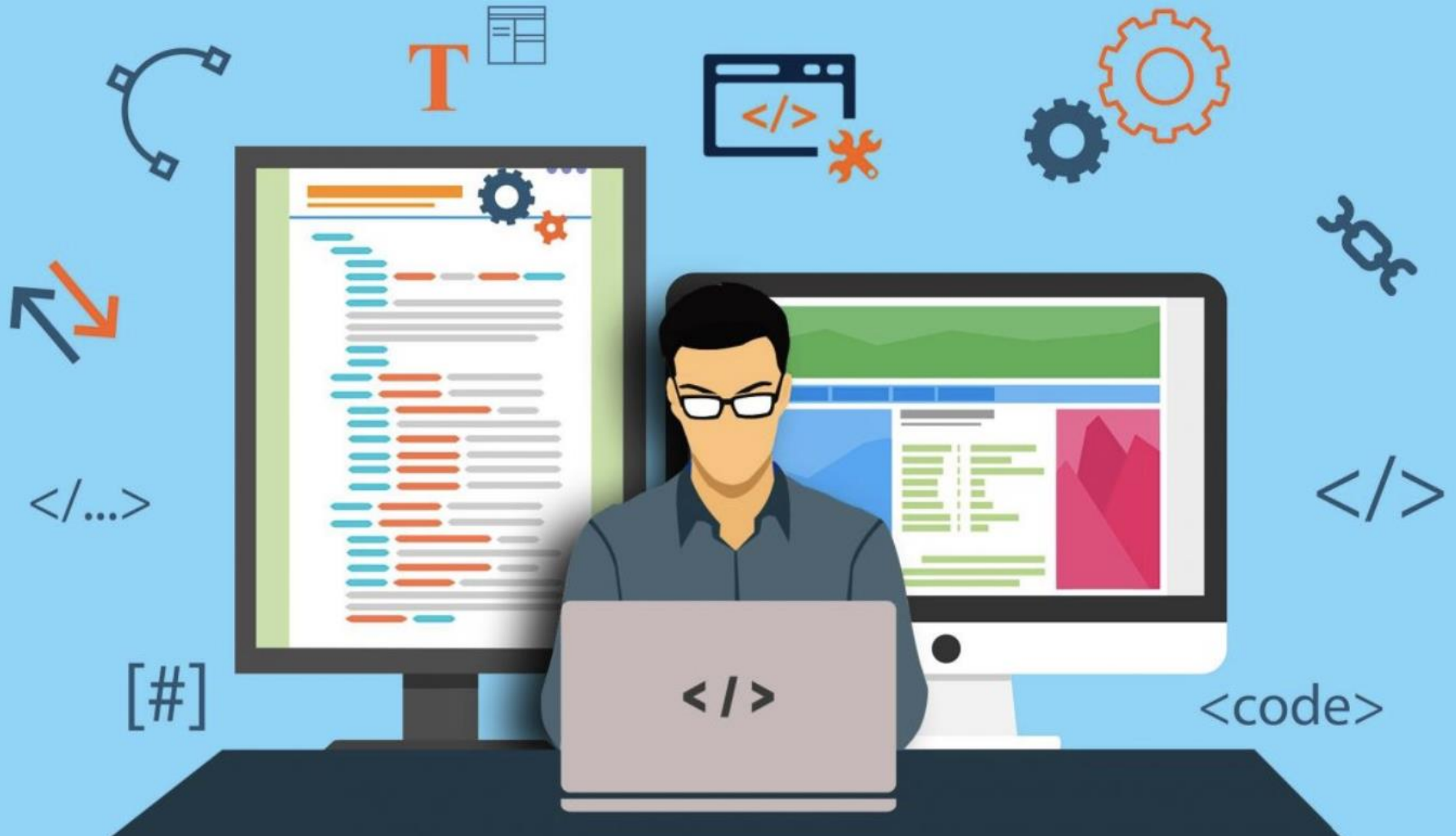


Cascading Style Sheets II



Blog submission

- Blogs must be completed **Friday 23rd**
October
 - Only Weeks 1-4 are required
-

Introduction

1. Recap of Week 3
 2. Pseudoclasses and pseudoelements
 3. Positioning & aligning layers with CSS
 4. Other CSS page layout tips
-

Pseudoclasses & Pseudoelements

Using Pseudoclasses

- **Pseudoclasses** are identifiers that are understood by user agents and apply to elements of certain types without the elements having to be explicitly styled
 - **Example:** A handful of pseudoclasses can be used with anchor elements `<a>`
-

Using Pseudoclasses

Pseudoclass	Match
:link	Unvisited links
:visited	Visited links
:active	Active links
:hover	The link that the browser pointer is hovering over
:focus	The link that currently has the user interface focus

Using Pseudoclasses

a:link

```
{  
    font-family: Tahoma;  
    color:black;  
    font-size: 10pt;  
    text-decoration:none;  
}
```

a:visited

```
{  
    font-family: Tahoma;  
    color:purple;  
    font-size: 10pt;  
    text-decoration:none;  
}
```

This definition is in the separate CSS file.

Every time a <a> tag is used, the attributes defined in the style sheet are automatically used

Using Pseudoclasses

When setting the style for several link states, there are some order rules:

- `a:hover` MUST come after `a:link` and `a:visited`
- `a:active` MUST come after `a:hover`

Using Pseudoelements

- **Pseudoelements**** are another virtual construct to help apply styles dynamically to elements within a document, e.g., :first-line, :first-letter

```
<head>
<title></title>
<style type="text/css">
    p:first-line {text-decoration: underline}
</style>
</head>
```

- Whenever <p> is used, the first line of the paragraph will be automatically underlined

****Pseudo classes are very similar to pseudo elements, but pseudo classes apply the styling rules to the element depending on some state.**

An Exercise...

- Using the handout provided:
 - Link the css file to the html file
 - Use HTML5 tags for divs where possible
 - In the .css file, style the following html elements:
 - `<body>`
 - `<p>`
 - `<h1>` and `<h2>`
 - Create a second nav inside the footer. Use ids to differentiate navs.
 - Put id declaration with relevant attributes in css file
 - Put relevant html code in html file
-

Another Exercise...

- ❑ Create a **class** which will be used to hold the main content of the web page
 - Put class declaration with relevant attributes in css file
 - Put relevant html code in html file

 - ❑ Change the colour of the hyperlink
 - The hyperlink should be **navy** whether it has been visited nor not
 - Remove the underline from the hyperlink
-

Positioning & Aligning Layers with CSS

What are the options?

- Position attribute – *this week*
 - Float attribute – *this week*
 - Box Model – *next week*
 - Display attribute – *next week*
 - CSS Grid and CSS frameworks – *later weeks*
-

CSS attribute - position

- Position has 4 values:
 - ❑ static (default value)
 - ❑ fixed
 - ❑ relative
 - ❑ absolute
-

CSS attribute – **position: static**

- A **static** element is positioned according to the normal flow of the page.
 - These elements are not affected by top, bottom, left and right.
-

CSS attribute – **position: fixed**

- A **fixed** element is positioned relative to the browser window. Even if scrolled, it does not move.
 - It is removed from the natural flow.
 - These elements use top, bottom, left and right for positioning.
-

CSS attribute – position: fixed

```
<!DOCTYPE html>
<html>
<head>
<style>
h2.pos_top
{
position:fixed;
top:50%;
}
</style>
</head>

<body>
<h2>This is a heading with no position</h2>
<h2 class= pos_top >This heading is placed acc
window</h2>
</body>
```

CSS attribute – position: fixed

```
<!DOCTYPE html>
<html>
<head>
<style>
h2.pos_top
{
position:fixed;
top:50%;
}
</style>
</head>

<body>
<h2>This is a heading with no position</h2>
<h2 class="pos_top">This heading is placed according to the browser
window</h2>

</body>
```

This is a heading with no position

This heading is placed according to the browser window

CSS attribute – position: relative

- A **relative** element is positioned relative to its normal position.
 - It has a place in the natural flow.
 - These elements use top, bottom, left and right for positioning.
-

CSS attribute – position: relative

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
h2.pos_top
```

```
{
```

```
position: relative;
```

```
top: -50px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h2>This is a heading with no position.</h2>
```

```
<h2 class="pos_top">This heading is moved upwards according to its normal position</h2>
```

```
<p><b>Note:</b> Even if the content of the relatively positioned element is moved, the reserved space for the element is still preserved in the normal flow.
```

```
</p>
```

```
</body>
```

```
</html>
```

CSS attribute – position: relative

```
<!DOCTYPE html>
<html>
<head>
<style>
h2.pos_top
{
position:relative;
top:-50px;
}
</style>
</head>

<body>
<h2>This is a heading with no position</h2>
<h2 class="pos_top">This heading is moved upwards according to its normal
position</h2>
<p><b>Note:</b> Even if the content of the relatively positioned element is
moved, the reserved space for the element is still preserved in the normal flow.
</p>
</body>

</html>
```

This heading is moved upwards according to its normal position

Note: Even if the content of the relatively positioned element is moved, the reserved space for the element is still preserved in the normal flow.

CSS attribute – position: absolute

- An **absolute element** is positioned relative to the first parent layer.
 - It is removed from the natural flow.
 - These elements use top, bottom, left and right for positioning.
-

Right/Left aligning using position

```
.right
{
  position:absolute;
  right:0px;
  width:300px;
  background-color:#b0e0e6;
}
```

CSS attribute – **position: sticky**

- A **sticky** element is positioned based on the user's scroll position.
 - A sticky element toggles between relative and fixed, depending on the scroll position.
 - positioned **relative** until a given offset position is met in the viewport
 - then it "sticks" in place (like **position:fixed**).
 - These elements use top, bottom, left and right for positioning.
-

CSS attribute – position: sticky

relative

Try to **scroll** inside this frame to understand how sticky positioning works.

Note: IE/Edge 15 and earlier versions do not support sticky position.

I am sticky!

In this example, the sticky element sticks to the top of the page (top: 0), when you reach its scroll position.

Scroll back up to remove the stickiness.

Some text to enable scrolling.. Lorem ipsum dolor sit amet, illum definitiones no quo, malisset concludaturque et eum, altera fabulas ut quo. Atqui causae gloriatur ius te, id agam omnis evertitur eum. Affert laboramus repudiandae nec et. Inciderint efficiantur his ad. Eum no molestiae voluptatibus.

Some text to enable scrolling.. Lorem ipsum dolor sit amet, illum definitiones no quo, malisset concludaturque et eum, altera fabulas ut quo. Atqui causae gloriatur ius te, id agam omnis evertitur eum. Affert laboramus repudiandae nec et. Inciderint efficiantur his ad. Eum no molestiae voluptatibus.

After scroll - *fixed*

I am sticky!

Affert laboramus repudiandae nec et. Inciderint efficiantur his ad. Eum no molestiae voluptatibus.

Some text to enable scrolling.. Lorem ipsum dolor sit amet, illum definitiones no quo, malisset concludaturque et eum, altera fabulas ut quo. Atqui causae gloriatur ius te, id agam omnis evertitur eum. Affert laboramus repudiandae nec et. Inciderint efficiantur his ad. Eum no molestiae voluptatibus.

Use position: fixed

Some text

Some text

Some text

Some text

Some text

Some text

Some text

Some text

Some text

Some text

This text never moves.

Use position: fixed

```
<!DOCTYPE html>
<html>
<head>
<style>
p.pos_fixed {
    position: fixed;
    top: 30px;
    right: 5px;
    color: red;
}
</style>
</head>
<body>

<p>Some text</p><p>Some text</p><p>Some text</p><p>Some
text</p><p>Some text</p><p>Some text</p><p>Some text</p>
<p>Some text</p><p>Some text</p><p>Some text</p><p>Some
text</p><p>Some text</p><p>Some text</p><p>Some text</p>
<p>Some text</p><p>Some text</p>
<p class="pos_fixed">This text never moves.</p>
</body>
</html>
```

Use position: relative

Heading with no position

This heading is moved left according to its normal position

This heading is moved right according to its normal position

Use position: relative

```
<!DOCTYPE html>
<html>
<head>
<style>
h2.pos_left {
    position: relative;
    left: -20px;
}

h2.pos_right {
    position: relative;
    left: 20px;
}
</style>
</head>
<body>
<h2>Heading with no position</h2>
<h2 class="pos_left">This heading is moved left according
to its normal position</h2>
<h2 class="pos_right">This heading is moved right
according to its normal position</h2>
</body>
</html>
```

Use position: absolute

With absolute positioning, an element can be placed anywhere on a page. The heading below is placed 100px from the left of the page and 150px from the top of the page.

This heading has an absolute position

Use position: absolute

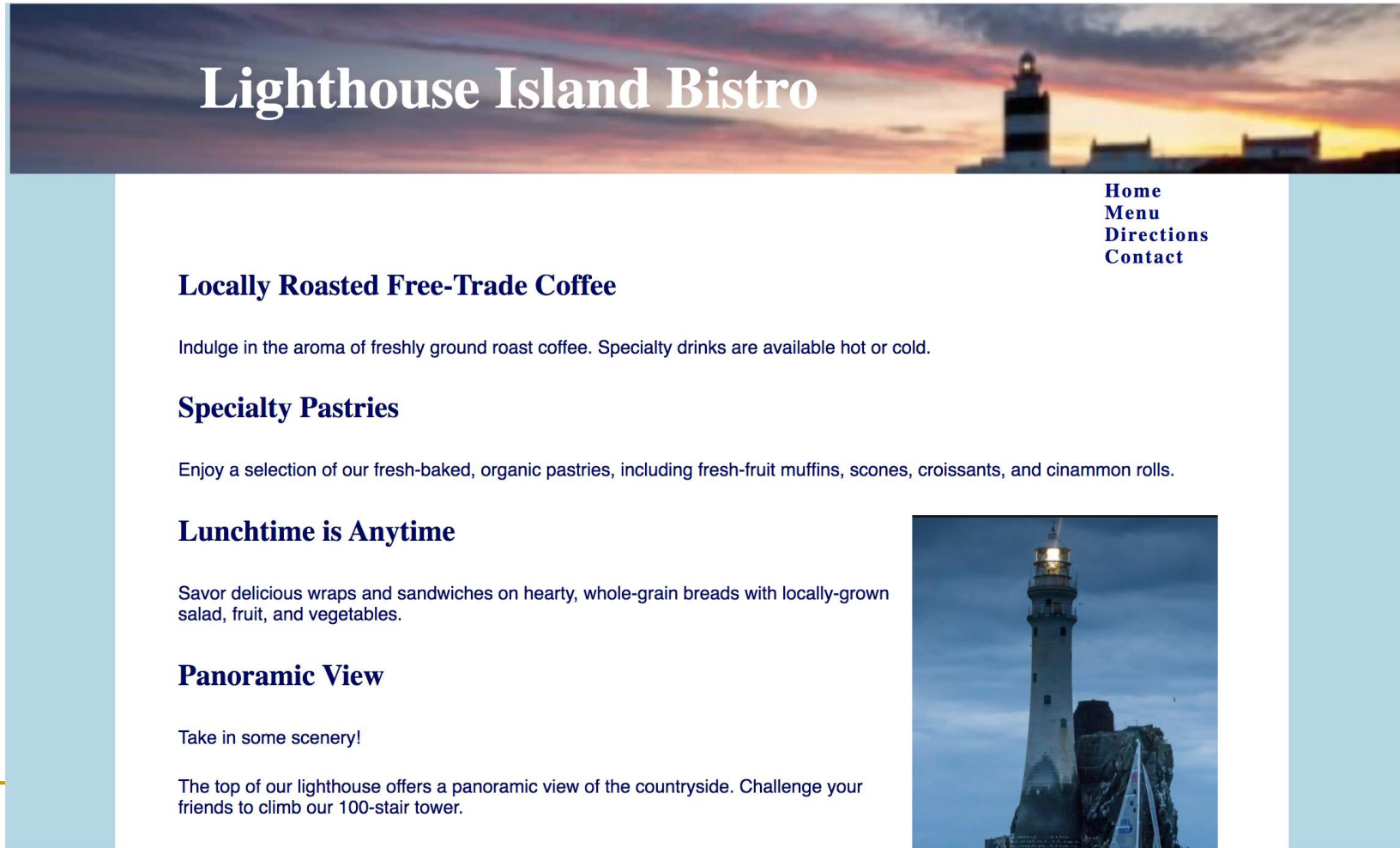
```
<!DOCTYPE html>
<html>
<head>
<style>
h2 {
    position: absolute;
    left: 100px;
    top: 150px;
}
</style>
</head>
<body>

<h2>This heading has an absolute position</h2>
<p>With absolute positioning, an element can be placed
anywhere on a page. The heading below is placed 100px
from the left of the page and 150px from the top of the
page.</p>

</body>
</html>
```


Lighthouse website – Using position

- Make the header of the website fixed i.e. **position:sticky**



Other CSS Page Layout tips

CSS Float

- Elements that appear to float on the right or left side of either the browser window or another element are often configured using the **float** property
- `float:right`
- `float:left`



Concourse A ↑
ターミナレ Terminal
手荷物受取所 Luggage Claim
発券 Hotel

Floats: Left and Right

Using floats can introduce up to two new flows: left and right.

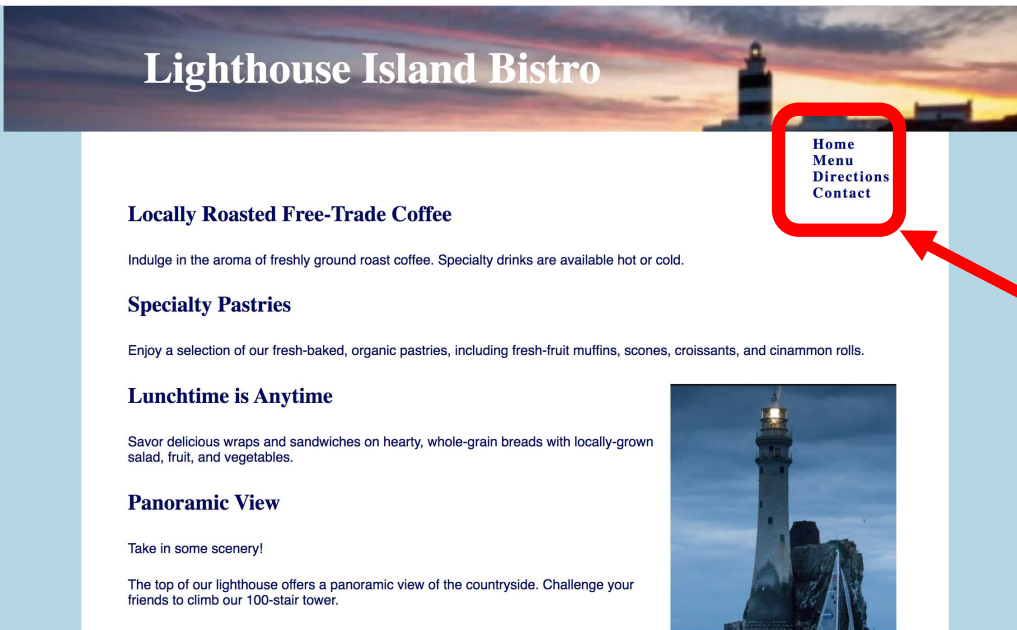


<https://medium.freecodecamp.com/css-floats-explained-by-riding-an-escalator-57fa55232333#.adlt0m4gk>

Right/Left aligning using float

```
.right  
{  
  float:right;  
  width:300px;  
  background-color:#b0e0e6;  
}
```


Lighthouse website – Using float



```
#mainnav { float: right;  
            width: 150px;  
            letter-spacing: 0.1em;  
            font-weight: bold;  
        }
```

CSS: Clearing a Float

- The **clear** property is often used to terminate, or clear a float
 - A common technique is to add a line break configured with the clear property
 - `clear:right`
 - `clear:left`
 - `clear:both`
-

Lighthouse website – Using clear

```
footer {font-size:70%;  
        padding: 10px;  
        clear: both;  
}
```

CSS Overflow

- The **overflow** property is often used to clear a float, although its intended purpose is to configure how content should be displayed if it is too large for the area allocated
- `overflow:visible`
- `overflow:hidden`
- `overflow:auto`
- `overflow:scroll`

Centre aligning

```
.center  
{  
margin-left:auto;  
margin-right:auto;  
width:70%;  
background-color:#b0e0e6;  
}
```

Split the available margins equally

Cannot be 100%

Horizontal Navigation with an Unordered List

- How can I use an unordered list for a horizontal navigation menu? **display** property
 - `display: inline`
-

Horizontal Navigation with an Unordered List

```
<div id="nav">
```

```
  <ul>
```

```
    <li>.....</li>
```

```
    <li>.....</li>
```

```
    <li>.....</li>
```

```
    <li>.....</li>
```

```
  </ul>
```

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

Write the css which will output:

Home Menu Directions Contact

CSS Debugging Techniques

- Verify correct HTML syntax
 - Verify correct CSS syntax
 - Configure temporary background colours
 - Configure temporary borders
 - Use comments to find unexpected cascade
-

CSS Debugging Techniques

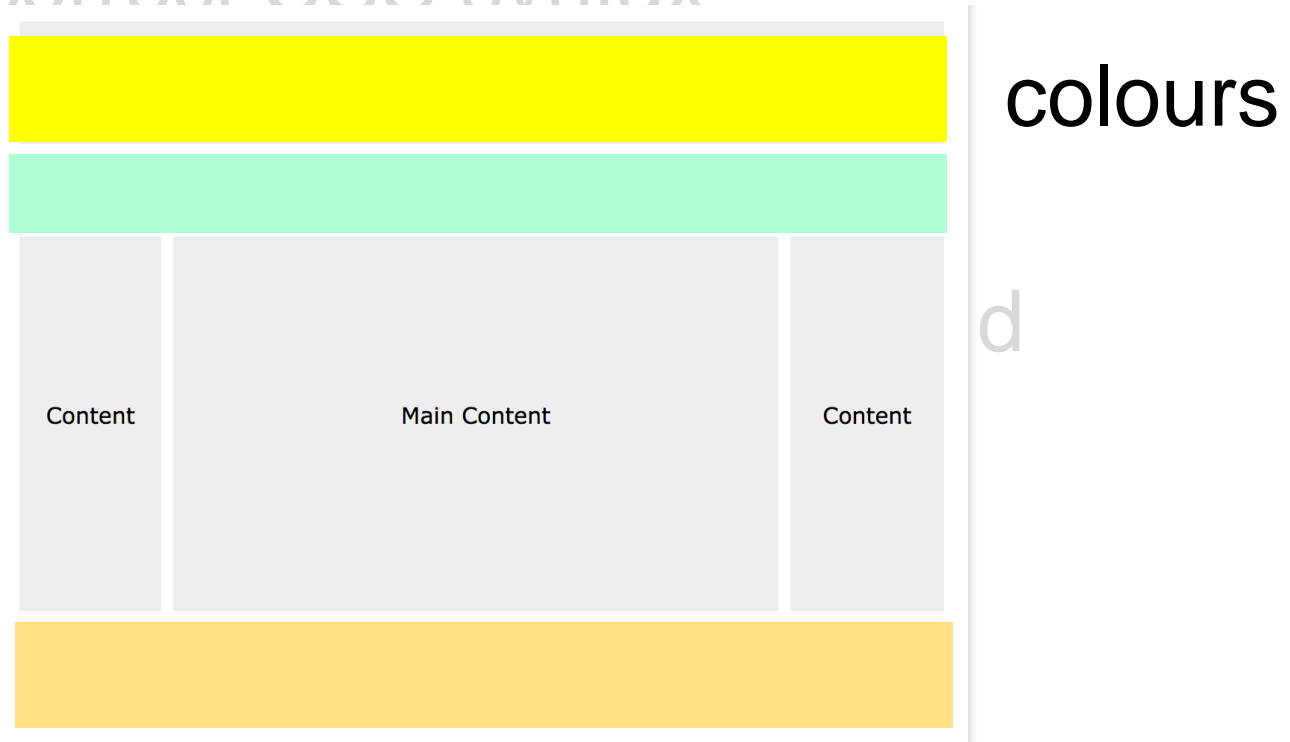
- Verify correct HTML syntax
- Verify correct CSS syntax
- Configure temporary background colours
- Configure **HTML Validator - <https://validator.w3.org/>**
- Use comments to find unexpected cascade

CSS Debugging Techniques

- Verify correct HTML syntax
- Verify correct CSS syntax
- Configure temporary background colours
- **CSS Validator -**
<https://jigsaw.w3.org/css-validator/>
- Use comments to find unexpected cascade

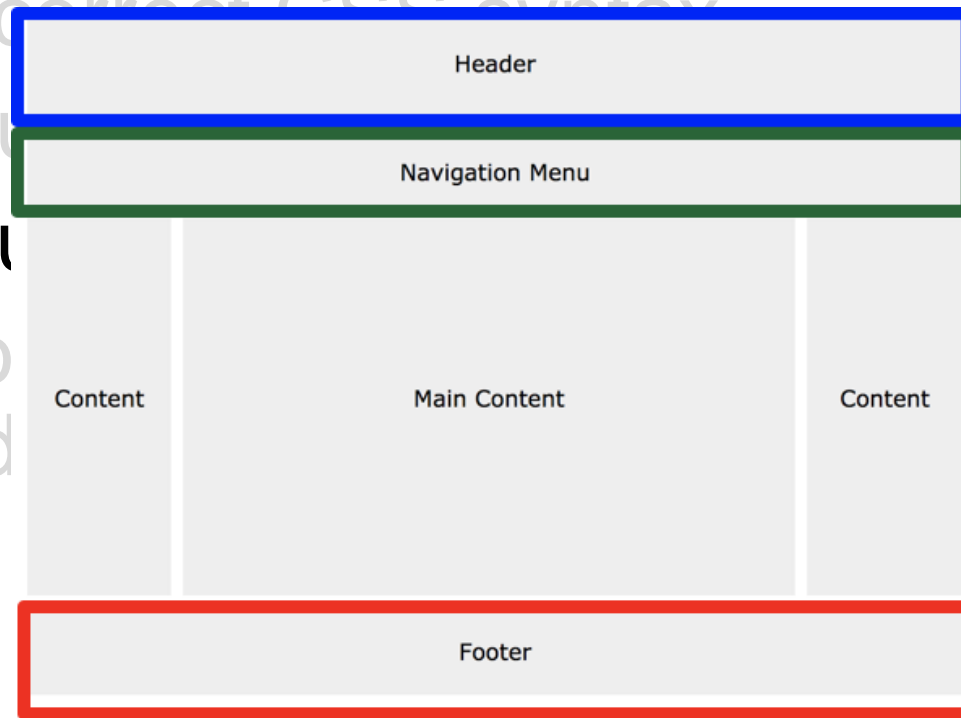
CSS Debugging Techniques

- Verify correct HTML syntax
- Verify correct CSS syntax
- **Configure**
- Configure
- Use cascading



CSS Debugging Techniques

- Verify correct HTML syntax
- Verify correct CSS syntax
- Configure browser developer tools
- Configure browser zoom
- Use cascading



colours

d