

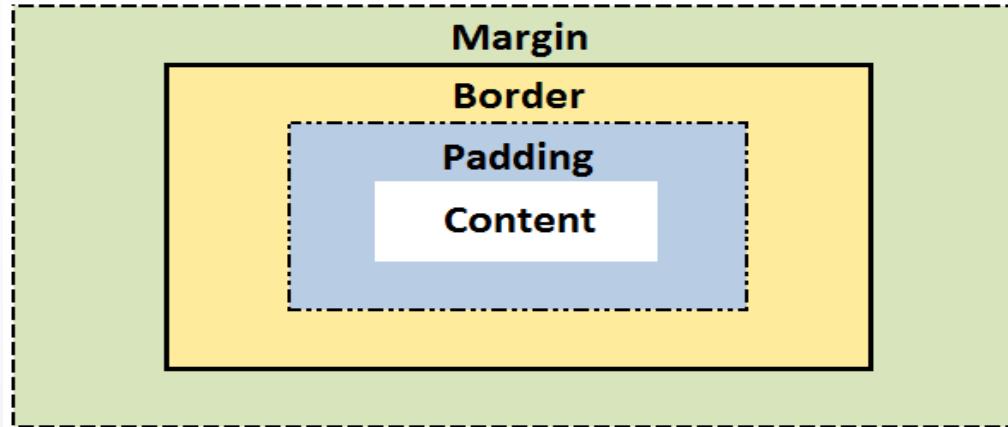
WEB222 - Web Programming Principles

Week 9: More on CSS,
Page Layouts and Navigation

Agenda

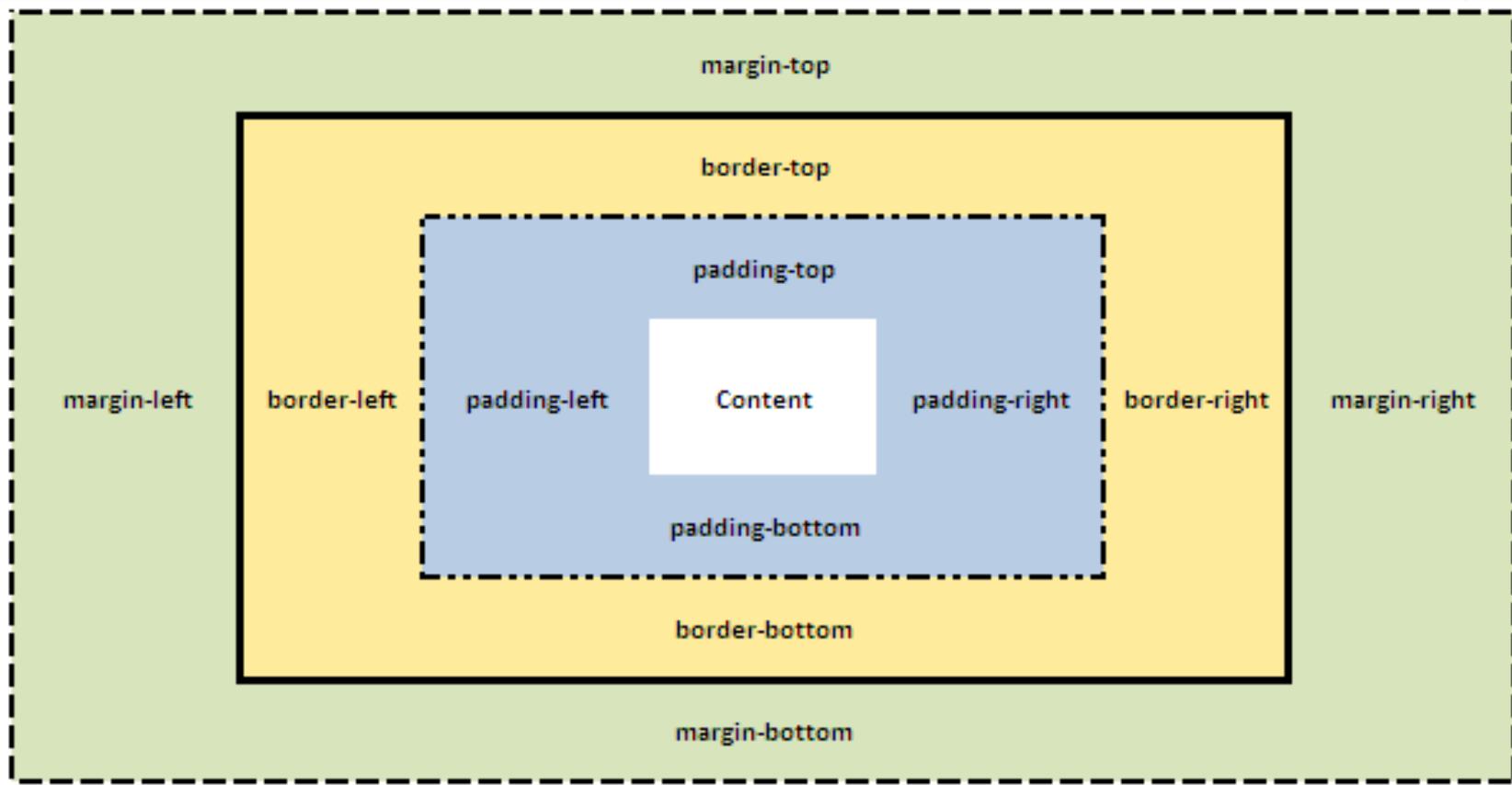
- CSS Box model
 - margin, border, padding, shorthand
- Table formatting with CSS
- Centering and positioning
- CSS3 shadow effects and rounded corners
- HTML 5 structural elements
- Page layouts
- Navigation/menus
- Image gallery example

The CSS Box model



- All elements can be considered to be box.
- The box model is the specification that defines how a box and its attributes relate to each other.
- A box is made up of four distinct parts, from the outside one to the inside one:
 - **margin, border, padding, and content.**

The CSS Box model



[box-model.html](#)

CSS Properties for Box Model

- CSS margin
- CSS border
 - border-width
 - border-style (required)
 - border-color
- CSS padding
- Using shorthand

The CSS Box Model – margin

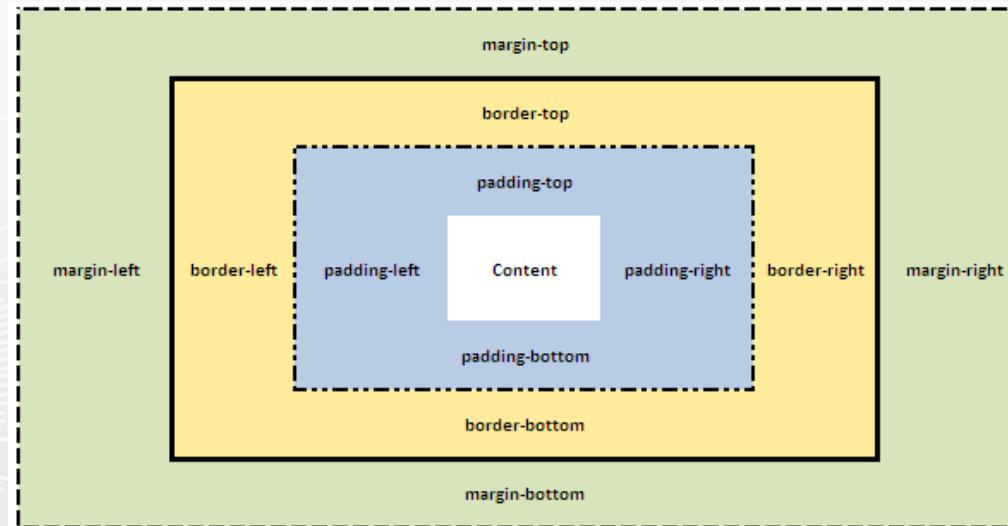
- CSS Margin property defines the white space around an HTML element's border. See the "Box model".

margin	Applies to all sides
margin: 6px; /* this is a shortcut */	Applies a margin to all sides of an element
margin-top	Applies a margin to the top of an element
margin-right	Applies a margin to the right of an element
margin-bottom	Applies a margin to the bottom of an element
margin-left	Applies a margin to the left of an element

□ [box-margin.html](#)

CSS Shorthand for Box Model

- CSS Shortcuts allow for a property to have a single or multiple values.
- Shortcuts/shorthand order: **CLOCKWISE**:
 - **top** -> **right**-> **bottom**→ **left**



The margin Shorthand Property

- To set all the margin properties in one declaration:

- **margin:10px 5px 15px 20px;**
 - ▶ top margin is 10px
 - ▶ right margin is 5px
 - ▶ bottom margin is 15px
 - ▶ left margin is 20px
- **margin:10px 5px 15px;**
 - ▶ top margin is 10px
 - ▶ right and left margins are 5px
 - ▶ bottom margin is 15px
- **margin:10px 5px;**
 - ▶ top and bottom margins are 10px
 - ▶ right and left margins are 5px
- **margin:10px;**
 - ▶ all four margins are 10px

Margin Collapsing

- Top and bottom margins of blocks are sometimes combined (collapsed) into a single margin whose size is the largest of the margins combined into it, a behavior known as margin collapsing.
- e.g.

```
h1 { margin: 0 0 25px 0; background: #fcf; }  
p { margin: 20px 0 0 0; background: #cf9; }
```



<http://www.sitepoint.com/web-foundations/collapsing-margins/>

The CSS Box Model – border

- The **border** property allows for setting the **width**, **style** and **color** and of the borders around an element.
- The border **style** property value for the border **must be stated**, otherwise no border will show up.

The **border-width** Property

border-width	Applies to all sides
border-width: 6px; border-style: solid;	Applies a solid border to all sides
border-top-width	Applies only to the top border
border-right-width	Applies only to the right border
border-bottom-width	Applies only to all bottom border
border-left-width	Applies only to the left border

- The **border-width** can be set in pixels, ems, or one of the three pre-defined values: **thin**, **medium**, or **thick**.
- [border-width.html](#)

The border-width Shorthand

Example	Description
border-width:6px;	adds a border - 6px to all four sides
border-width:6px 12px;	adds a border - 6px to top and bottom - 12px to the right and left
border-width:6px 12px 10px;	adds a border - 6px to the top, 12px to the right, 10px to bottom and 12px to the left
border:6px solid red;	Width, style, color!

[border-short.html](#)

The **border-style** property

border-style	Applies to
border-style: solid; /* default width of 3px */	Applies a solid border to all sides
border-top-style	Applies the style only to the top border
border-right-style	Applies the style only to the right border
border-bottom-style	Applies the style only to all bottom border
border-left-style	Applies the style only to the left border

- The property can have a value from the list:
 - **dotted, dashed, solid, double, groove, ridge, inset, outset, hidden.**
- [border-style.html](#)

The **border-color** property

border-color	Applies to
border-color:#ff0000; border-style: solid;	Applies a solid border to all sides
border-top-color	Applies only to the top border
border-right-color	Applies only to the right border
border-bottom-color	Applies only to all bottom border
border-left-color	Applies only to the left border

[border-color.html](#)

The Border Shorthand Property

- The **border** property is a shorthand for the following individual border properties:
 - border-width
 - border-style (required)
 - border-color
- Example:

```
p.type4 { border:6px inset #ff0000; }
```

[border-short.html](#)

The CSS Box Model – padding

- CSS **Padding** property defines the white space around the inside of an HTML element's border. See the "Box model".

padding	Applies to all sides
padding:6px; /* this a shortcut */	Applies padding to all sides of an element
padding-top	Applies padding to the top of an element
padding-right	Applies padding to the right of an element
padding-bottom	Applies padding to the bottom of an element
padding-left	Applies padding to the left of an element

[box-padding.html](#)

Table Formatting with CSS

- HTML Tables are the most complex elements in HTML. A table may contain a caption, row, cell, row groups, and column groups.
- Some CSS properties for table formatting:
 - margin, padding, width, height, text-align, vertical-align, background-color, background-image, border

Table Formatting with CSS

- Formatting table examples:

```
table { margin: auto; width: 80%; }  
#table2 { border: 1px solid black;  
          background-color: yellow; }
```

- Formatting table cell and table header examples:

```
td, th { border: 4px inset #4400FF;  
          padding: 10px 20px;  
          background-color: green;  
          height: 100px; width: 400px; }  
td { text-align: left; vertical-align: bottom; }
```

- [css_table.html](#)

Border Collapse

- Property: **border-collapse** sets whether the table borders are collapsed into a single border or separated.
- E.g.:

```
table
{
    border-collapse: collapse; /* default: separate */
}

table, th, td
{
    border: 1px solid black;
}
```

Table Sections/Groups

- **<thead>** - group the first one or more rows of a table for formatting
- **<tbody>** - group the middle rows of a table for formatting
- **<tfoot>** - group the last one or more rows of a table for formatting

[css table section.html](#)

CSS – display Property

- The **display** CSS property specifies the type of rendering box used for an element.
 - default value: inline
 - the value **none** lets you turn off the display of an element.
- e.g.

```
p.inline { display: none; }
```

□ [css display.html](#)

The display Property Values

Value	Description
inline	Default value. Displays an element as an inline element (like <code></code>)
block	Displays an element as a block element (like <code><p></code>)
inline-block	Displays an element as an inline-level block container. The inside of this block is formatted as block-level box, and the element itself is formatted as an inline-level box
inline-table	The element is displayed as an inline-level table
list-item	Let the element behave like a <code></code> element
table	Let the element behave like a <code><table></code> element
table-caption	Let the element behave like a <code><caption></code> element
table-column-group	Let the element behave like a <code><colgroup></code> element
table-header-group	Let the element behave like a <code><thead></code> element
table-footer-group	Let the element behave like a <code><tfoot></code> element
table-row-group	Let the element behave like a <code><tbody></code> element
table-cell	Let the element behave like a <code><td></code> element
table-column	Let the element behave like a <code><col></code> element
table-row	Let the element behave like a <code><tr></code> element
none	The element will not be displayed at all (has no effect on layout)

Centering Lines Of Text

- Centering lines of text in a paragraph or in a heading.
 - Examples:

```
p { text-align: center }  
h2 { text-align: center }
```
 - Note: never use the <center> tags, which is obsolete to center text.
- [center text.html](#)

Centering a Block (horizontally)

- Rephrase: making left and right **margin** to be equal.
 - set the **margin-left** and **margin-right** to 'auto'.
 - used with a block with a fixed width.

```
div.center {  
    width: 400px;  
    border: 2px solid red;  
    margin: 0 auto;  
    /* margin-left: auto; margin-right: auto; */  
}  
  
<div class ="center"> ... </div>
```

center_block.html

Centering – Vertically

- Specify the outer block as a table cell, the contents of a table cell can be centered vertically.

```
div { height: 100px; width: 500px; }  
div.center { border: 10px dotted red;  
            display: table-cell;  
            vertical-align: middle;  
            text-align: center;  
}  
  
<div class="center"> This div is centered </div>
```

[center_vertical.html](#)

Positioning

- CSS **position** property
 - can be used to position elements precisely in HTML pages.
- a browser renders html statements in the order that they are in the html file - this is called **normal flow**

Positioning

- Use the **position** property. Values:
 - **static** - position using normal flow (default)
 - **absolute** - position precisely within the containing element
 - **relative** - position precisely relative to normal flow
 - **fixed** - position precisely within the browser window, and does not move when the page is scrolled
- Use "**left**" or "**right**" with "**top**" or "**bottom**" properties. Their values: numeric offsets in px or %.
- Example:

```
div#textbox { position:absolute; left:150px; top:150px; }
```

- [position.html](#)
- [position2.html](#)

Positioning

- What if the text takes more than the allotted space?
- use the property "**overflow**" to specify an action:
 - { overflow:scroll; } - include scroll bars
 - { overflow:auto; } - scroll if required
 - { overflow:hidden; } - hide overflow
 - { overflow:visible; } – default

[position.html](#)

Positioning

- graphics and titles can be positioned in a similar fashion:

position_graphic.html

CSS3 Rounded Corners

- Property: **border-radius**
- Example:

```
div { border: 2px solid #a1a1a1;  
      padding: 10px 40px;  
      background: grey;  
      width: 300px;  
      border-radius: 50px;  
      /* border-radius:10%; */  
      -moz-border-radius:25px; /* Firefox 3.6 and earlier */  
}
```

□ [boxShadow roundedCorners.html](#)

CSS3 Rounded Corners

border-radius:2em;

is equivalent to:

border-top-left-radius: 2em;

border-top-right-radius: 2em;

border-bottom-right-radius: 2em;

border-bottom-left-radius: 2em;

CSS3 Box Shadow

- CSS3 provides not only text-shadow but also box-shadow effects.
- Property: **box-shadow**
- box-shadow value: h-shadow v-shadow blur spread color inset;
- ❑ [boxShadow roundedCorners.html](#)

CSS3 Box Shadow

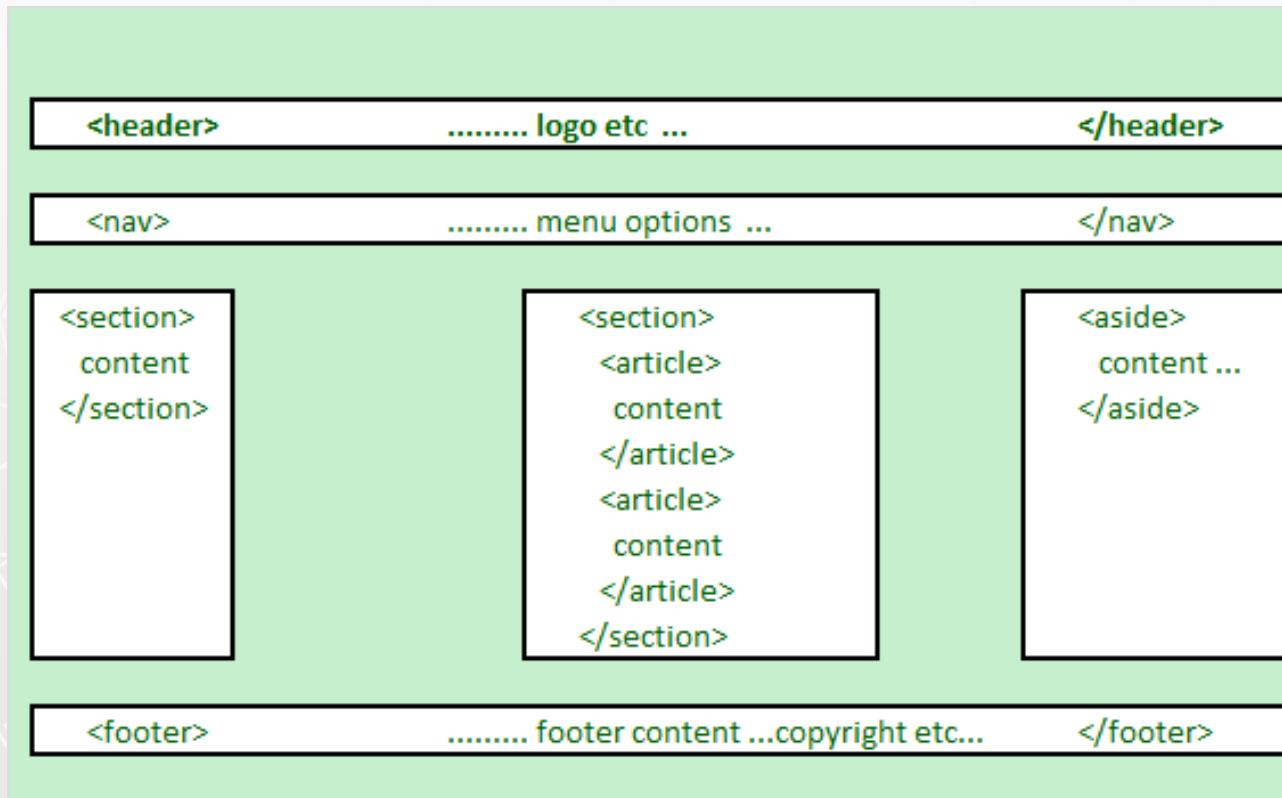
Value	Description
<i>h-shadow</i>	Required. The position of the horizontal shadow. Negative values are allowed
<i>v-shadow</i>	Required. The position of the vertical shadow. Negative values are allowed
<i>blur</i>	Optional. The blur distance
<i>spread</i>	Optional. The size of shadow
<i>color</i>	Optional. The color of the shadow. The default value is black. Look at CSS Color Values for a complete list of possible color values.
<i>inset</i>	Optional. Changes the shadow from an outer shadow (outset) to an inner shadow

 [css-ball.html](#)

<http://www.cssmatic.com/box-shadow>

HTML5 Structural Elements

- HTML 5 defines a number of new container elements for constructing documents.
 - header, nav, section, aside, article and footer



HTML5 Structural Elements

Elements	Description
<header>	the web page/site header content
<nav>	the navigation functionality for the page/site.
<section>	the grouping of related subjects on the web page.
<main>	the main content on the web page.
<article>	contains a standalone content on the web page.
<aside>	used for content that's not central to the web page.
<footer>	the web page/site footer content

Notes: The <div> element is the generic container for flow content, which does not inherently represent anything. It should be used only when no other semantic element (such as above elements) is appropriate.

HTML5 Structural Elements

```
...  
<body>  
  <header>..... logo etc ... </header>  
  <nav> ..... menu options ... </nav>  
  <section id="sidebar1">..... section 1 ... </section>  
  <section id="main"><!-- may be replaced by main element -->  
    <article>article within the section </article>  
    <article>another article within the section </article>  
  </section>  
  <aside>..... aside content ... </aside>  
  <footer> ..... footer content ...copyright etc... </footer>  
</body>  
</html>
```

[html5_structure.html](#)

HTML5 Structural Elements

- The `<article>` Tags may also contain these elements, e.g.

```
<article>
  <header> ... </header>

  <section id="introduction">
  </section>

  <section id="content">
  </section>

  <section id="summary">
  </section>

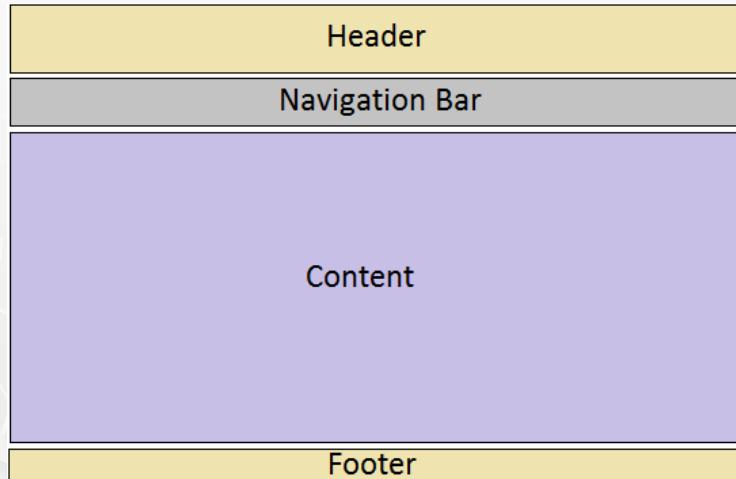
  <footer>...</footer>
</article>
```

The HTML4 Structural Elements: <div>

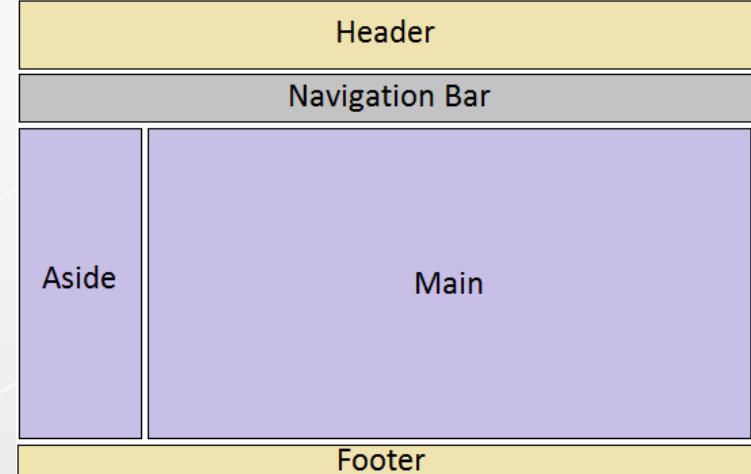
```
....  
<body>  
  <div class="header">..... logo etc ...</div>  
  <div class="navigation"> ..... menu options ... </div>  
  <div class ="sidebar1">..... Column 1 ... </div>  
  <div class="main">  
    .....main column content goes in here.....  
  </div >  
  <div class="aside">..... aside content ... </div>  
  <div class="footer"> ..... footer content ...copyright etc...  
</div>  
</body>  
</html>
```

Web Pages Layouts

➤ One-column layout



➤ Two-column layout

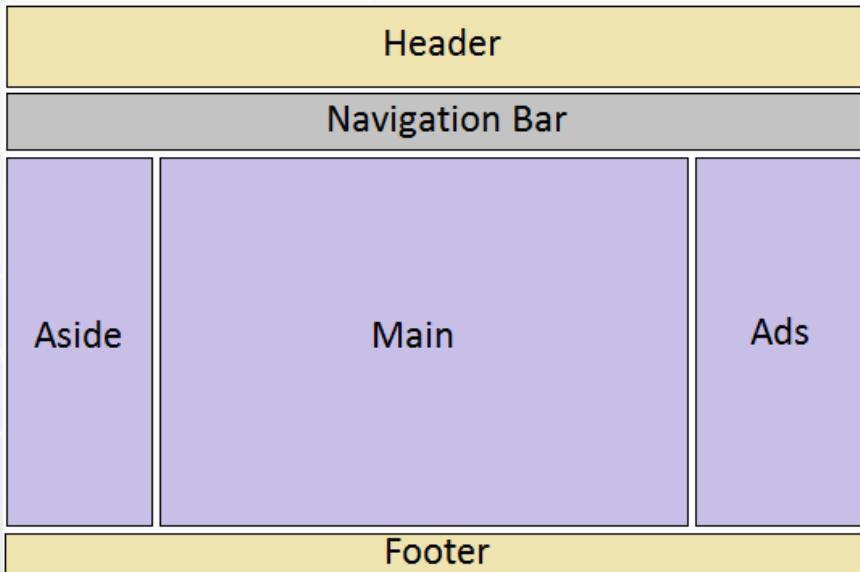


[layout-1-column.html](#)

[layout-2-column.html](#)

Web Pages Layouts

- Three-column layout



- [layout-3-column.html](#)

Create Layouts with HTML5 and CSS3

➤ Creating 2-column fluid (float-based) layouts with CSS

▪ HTML5 document without CSS:

- html5_structure.html

1. Set the width of the page (e.g. 960px) and center the page:

```
.container { max-width: 960px; margin: auto; }
```

2. Set the width of the “aside” block and float it to left:

```
aside { width: 192px; float: left; }
```

3. Set the width of the “main” section and float it to left:

```
main { width: 768px; float: left; }
```

Create Layouts with HTML5 and CSS3

3. Set the clear property of the footer to 'both':

```
footer { clear: both; background-color: #aaa; }
```

4. Set margin, border, padding, background-color, ... to each structural element, e.g.:

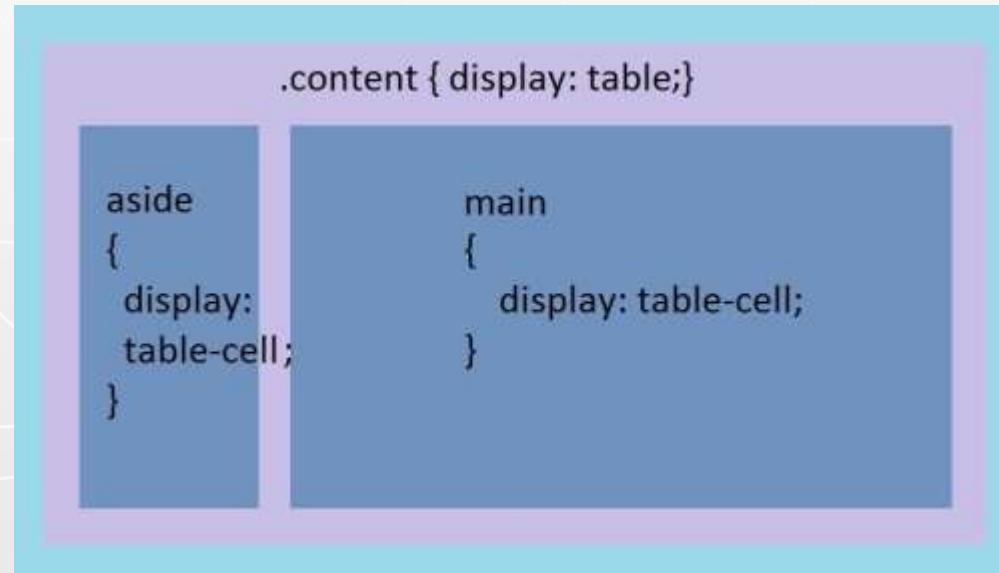
```
aside, main {margin-top: 58px; margin-right: 10px; margin-left: 10px; }
```

Note: You may use relative width values (e.g. 80%) for the page and columns.

Create Layouts with HTML5 and CSS3

- Creating 2-column tabular layouts with CSS
 - HTML5 document without CSS:
 - [html5_structure.html](#)

- Set CSS:



The diagram illustrates a 2-column layout structure. It features a light blue header bar at the top and a light purple footer bar at the bottom. The main content area is a light blue rectangle divided into two vertical columns. The left column is a light blue rectangle containing the CSS rule for the 'aside' element. The right column is a light blue rectangle containing the CSS rule for the 'main' element. The 'aside' rule is: 'aside { display: table-cell; }'. The 'main' rule is: 'main { display: table-cell; }'. The 'content' element, which is the parent of the 'aside' and 'main' elements, is styled with 'display: table;' in the header bar.

```
.content { display: table; }

aside
{
  display: table-cell;
}

main
{
  display: table-cell;
}
```

Create Layouts with HTML5 and CSS3

- Creating 2-column tabular layouts with CSS (cont')
 - CSS code for 2-column tabular layouts :

```
.container { display: table; width: 960px; margin: auto; }
aside { display: table-cell; width: 192px; }
main { display: table-cell; width: 720px; }
footer { background-color: #aaa; }
aside, main {margin-top: 58px; margin-right: 10px; margin-left: 10px;}
```

- ✓ Note: using HTML table to create page layouts is obsolete and not allowed in WEB222 assignments

Navigation and Menus

- Web page navigation: list of links

```
<nav>
  <ul>
    <li><a href="#top">Home</a></li>
    <li><a href="#timetable">Timetable</a></li>
    <li><a href="#standards">Standards</a></li>
    <li><a href="ibc233/ibc233.html">IBC233</a></li>
    <li><a href="int222/web222.html">WEB222</a></li>
    <li><a href="bt220/bt220.html">BTI220</a></li>
  </ul>
</nav>
```

- Using CSS to convert the unordered list to a navigation bar or menu.
- Usually, navigation/menus one each page of a website should be identical.

Navigation and Menus

- Single Level Menu Options
 - Horizontal Single Level Menu Example
 - **Navigation bar**
 - Vertical Single Level Menu Example
- Multi Level Menu Options
 - Horizontal Multi Level Menu Example**
 - **Dropdown menu**
 - Vertical Multi Level Menu Example

Creating a simple website

- Create a plan
 - How many web pages? What are the file names for these pages?
 - Website directory structure?
 - e.g. Create “css” sub-folder for storing CSS files; create “images” sub-folder for image files
- Create a template HTML document: template.html
 - HTML: links to external CSS file(s) and JavaScript file(s).
 - Design and code nav menu, footer, and/or theme/basic page layout.
 - Menu: items link to corresponding planned web pages.
- Copy and name template.html to all planned web pages
 - Nav menu, footer, theme of all pages should be the same.
 - Create page layout for each page.
 - index.html – the default home page shown on a website if no file name is specified.

Image Gallery Example

- CSS can be used to create an image gallery.
- Wrapping up images
 - In image gallery, an image may have a caption, description, or an action, which is performed when the image is clicked and JavaScript may be used.
- Examples of wrapping up images
 - Example 1:

```
<div class="image">
  <a target="_blank" href="images/travel-1.jpg">
    
  </a>
  <div class="desc">Travel</div>
</div>
```

Image Gallery Example

- Example 2:

```
<figure>
```

```
  <img src='images/tnnatgeo201201.jpg' alt=' '
    onclick='imageView("images/natgeo201201.jpg");'>
  <figcaption>The Matterhorn: Night Clouds #2 -- The Matterhorn,
  4478 m, at full moon. © Nenad Saljic/National Geographic
  Photo Contest)</figcaption>
```

```
</figure>
```

Image Gallery Example

- Setting up CSS for image boxes, e.g.

```
figure {  
    float: left; // for grid galleries  
    height: 175px; // size of image boxes  
    margin: 1em 2em 0 0;  
}
```

- Setting up CSS for images, e.g.

```
figure img {  
    padding: 10px;  
    border: 1px solid black;  
    border-radius: 5px;  
    margin: 10px;  
    cursor: pointer;  
}
```

Image Gallery Example

- Setting up CSS for captions, descriptions, e.g.

```
figure figcaption {  
    width: 200px;  
    font-size: 0.7em;  
    padding: 5px;  
    margin-left: -1000em;  
    margin-top: -20px;  
    background-color: #ffa;  
    border: 1px solid #ffad33;  
    border-radius: 5px;  
    position: absolute;  
}
```

Image Gallery Example

- Adding two more div elements on the page for showing up full-size images.

```
<div id="popup">  
  <!-- large version of the image -->  
    
  <div class="close" onclick="imageClose();">  
    &nbsp;X&nbsp;  
  </div>  
</div>  
<div id="popupbg"></div>
```

- grid-gallery.html

Image Gallery Example

➤ JavaScript for showing up full-size images

```
function imageView(bigImage) {  
  // on the full-size image, set the 'src' attribute that's passed in the '#popupImage' element is an img  
  document.querySelector('#popupImage').setAttribute('src', bigImage);  
  
  // show the full-size image. // the '#popup' element is a div  
  document.querySelector('#popup').style.display = 'block';  
  
  // show the faded background image. // the '#popupbg' element is a div  
  document.querySelector('#popupbg').style.visibility = 'visible';  
}
```

➤ JS for closing full-size images

```
function imageClose() {  
  // hide the full-size image  
  document.querySelector('#popup').style.display = 'none';  
  
  // hide the faded background image  
  document.querySelector('#popupbg').style.visibility = 'hidden';  
}
```

Resourceful Links

- CSS Library
- CSS Reference
<http://reference.sitepoint.com/css>
- Box model - CSS | MDN
https://developer.mozilla.org/en-US/docs/Web/CSS/box_model
- CSS Properties
<http://reference.sitepoint.com/css/propertyref>
- CSS Selectors
<http://reference.sitepoint.com/css/selectorref>
- Bootstrap's Responsive Grid and Navbar

Thank You!