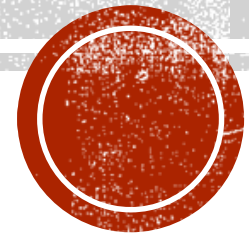


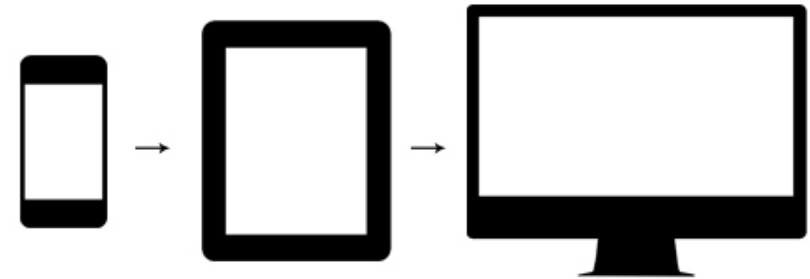
ITMD-362 WEEK 3: CSS REVIEW



WHY MOBILE FIRST DESIGN?

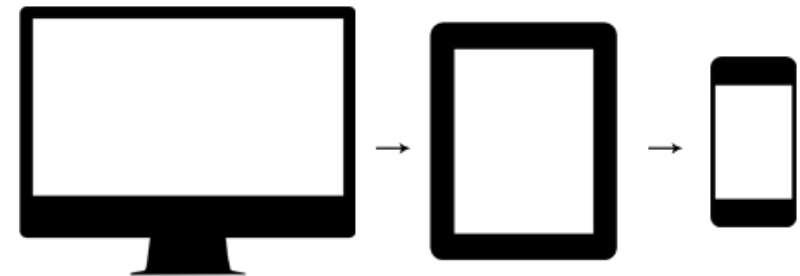
- Progressive Enhancement vs. Graceful Degradation

- Mobile-First = Content-First
- Responsive Design = Scale up
- Downside? Hardest part is first



- ABC's Excellent Mobile First design!

- Google Maps



ANATOMY OF A CSS RULE

declaration

selector { property: value; }

declaration block

selector {
property1: value1;
property2: value2;
property3: value3;
}



font-weight

Values: normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | inherit

Default: normal

Applies to: all elements

Inherits: yes

font-style

Values: normal | italic | oblique | inherit

Default: normal

Applies to: all elements

Inherits: yes

sample of oblique Times

sample of true italic Times

This is an example of oblique Times as rendered in a browser.

This is an example of italic Times as rendered in a browser.

CSS PROPERTIES

- Different Properties take different value types
- Measurement values should have NO SPACE between number and value, 3px not 3 px
 - %, px, em are the most common but there are more
 - Pixels are not created equally: See [comparisons](#)
- Properties with color values – Common Methods
 - Hex RGB Hexadecimal values, #34D2FF, #4D2
 - RGB, rgb(red, green, blue), 8bit 0-255, rgb(100,210,255)
 - Predefined Color Names, 147 named colors
 - Other methods supported in modern browsers
 - RGBA, HSL, HSLA – alpha value is a decimal 0 – 1
- Properties that take a URL need the value to be wrapped in functional notation url()
- <https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>



CSS CONCEPTS

- Inheritance
 - Some properties inherit their settings from their parent element
 - Mostly styles that effect text
- Parents & Children
 - Elements nested inside other elements are said to be children of that element.
 - The element that a given element is nested inside is its parent.
- It is very important that you understand the way your HTML is structured and how each of your elements are nested within each other to be successful writing complex CSS rules.



CSS INHERITANCE

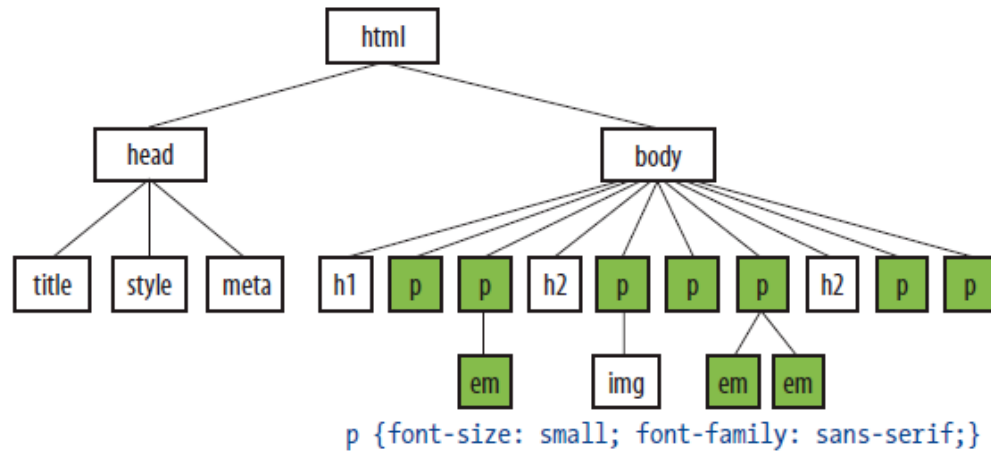


Figure 11-7. Certain properties applied to the `p` element are inherited by their children.

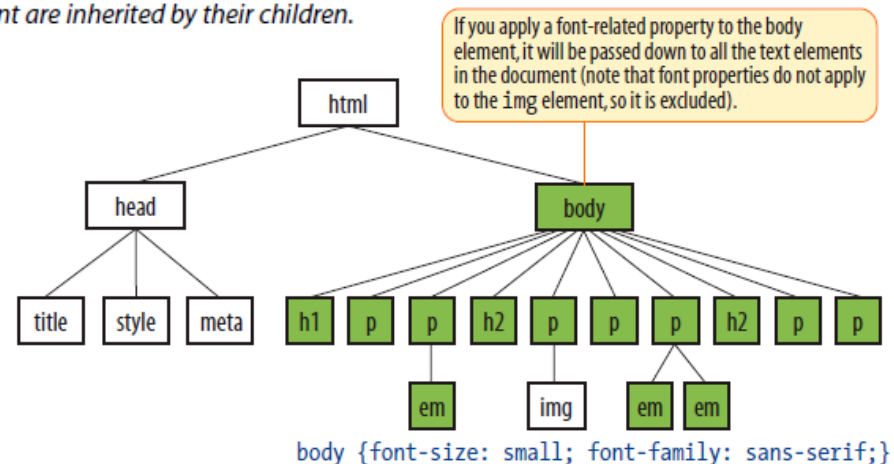


Figure 11-8. All the elements in the document inherit certain properties applied to the body element.



CSS SPECIFICITY

CSS Selector Specificity - *Cheat Sheet*

	*	id	classes	elements	result [*]
	a	b	c	d	
li	0	0	0	1	1
.lorem	0	0	10	0	10
#ipsum	0	100	0	0	100
style = " "	1000	0	0	0	1000
ul li	0	0	0	2	2
ul ol + li	0	0	0	3	3
ul ol li.red	0	0	10	3	13
li:first-line	0	0	0	2	2
#ipsum li	0	100	0	1	101
a:link	0	0	10	1	11

Legend:

a=1000

b=100

c=10

d=1

style = " "

ID attributes

Classes and pseudo-classes

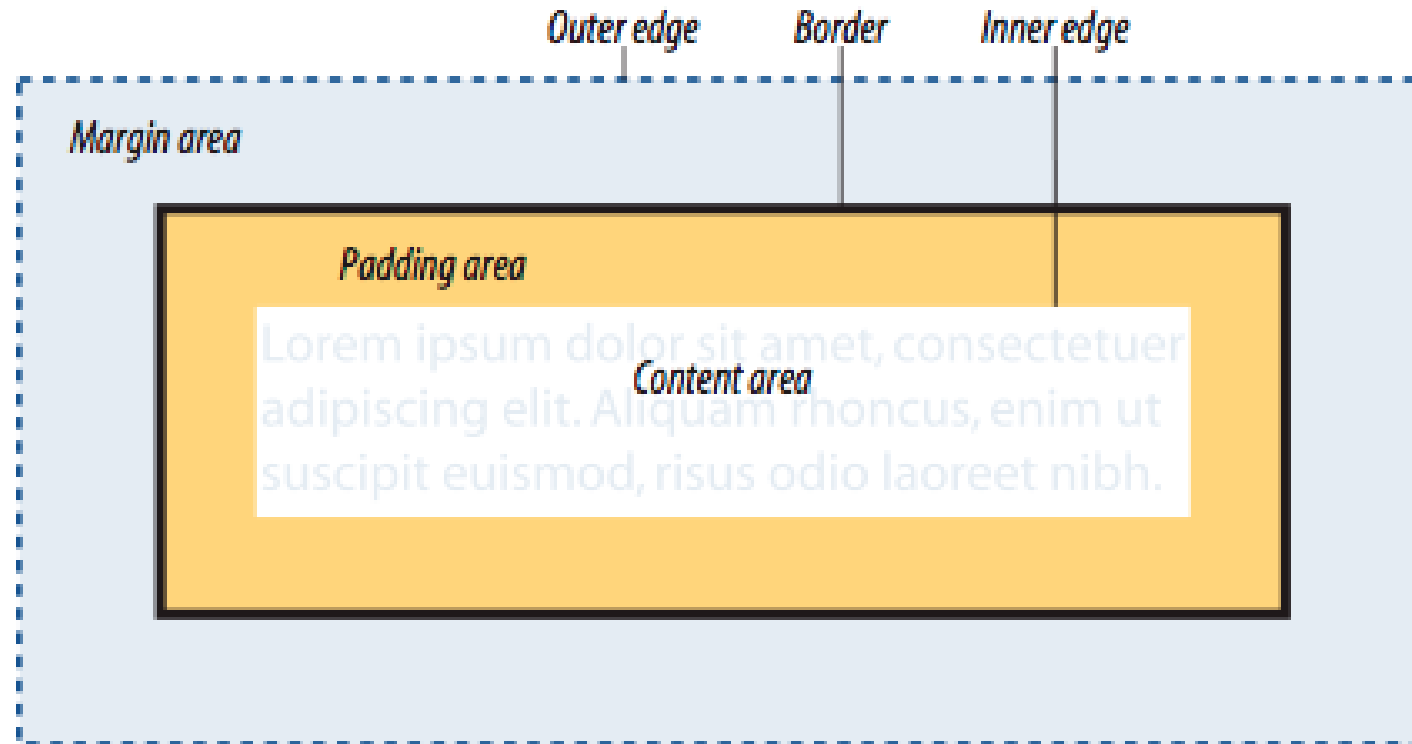
Elements and pseudo-elements

** Bigger
is
more
specific*



CSS BOX MODEL

- The Browser sees every element, block or inline, as a little rectangular box



ITMD-362

CSS FLOATS



**Does it...
Float or
Sink?**



CSS FLOATS

- “The float CSS property specifies that an element should be taken from the normal flow and placed along the left or right side of its container, where text and inline elements will wrap around it. A floating element is one where the computed value of float is not none.” – Mozilla MDN
- <https://developer.mozilla.org/en-US/docs/Web/CSS/float>
- Float value can be left, right, or none. Value of none turns off float
- Float property causes element to move as far as possible to the left or right and allow other elements to wrap around it.
- Always provide a width to floated element
- Floated elements behave like block elements
- Margins do not collapse like they do on a standard block element (top and bottom margins)



CSS FLOATS

- Clearing the float.
 - Apply the clear property to the element following the float that you don't want to wrap around the content.
 - It tells the element it is on that it can not be positioned next to a floated element.
- If a containing element only contains floated elements it will have no height unless you set some other properties.
 - Search clearfix
 - Or use overflow property set to hidden.
- You can use floats to create column layouts
- Floated elements need to appear in html before content that wraps around them



ITMD-362

CSS POSITION

CSS POSITION

- 4 values for CSS positioning (position property): [MDN LINK](#)
- Static
 - Normal default position in the document flow
- Relative
 - element is moved relative to position in document flow
 - Space in document flow **is reserved**
- Absolute
 - Removed from the document flow
 - Positioned with respect to **containing element with a position**
- Fixed
 - Removed from flow and positioned fixed to the window
- Then use top, bottom, left, or right properties to set position
- z-index property sets render layer. Only works when element has position set



ITMD-362

CSS RESETS

CSS RESETS

- Used to reset all css properties so they will be consistent in all browsers.
- Eric Meyer's css reset
 - Complete reset to no styles
 - <http://meyerweb.com/eric/tools/css/reset/>
- Normalize
 - Targets only styles that need to be reset
 - <http://necolas.github.com/normalize.css/>
- Yahoo YUI css reset
 - Mostly complete reset
 - <http://yuilibary.com/yui/docs/cssreset/>
- <http://www.cssreset.com/>



ITMD-362

NAVIGATION

CSS HORIZONTAL NAVIGATION

- Navigation is a list of links so we should use a list as the html element
- `ul`
 - Remove the bullets – `list-style-type: none`
 - Remove the margins and padding
- `ul li`
 - Float list items left - `float: left;`
- `ul li a`
 - Make anchor block and apply styles – `display: block;`
- Clear the element that follows – `clear: left;`



ITMD-362

LAYOUT

CSS LAYOUT

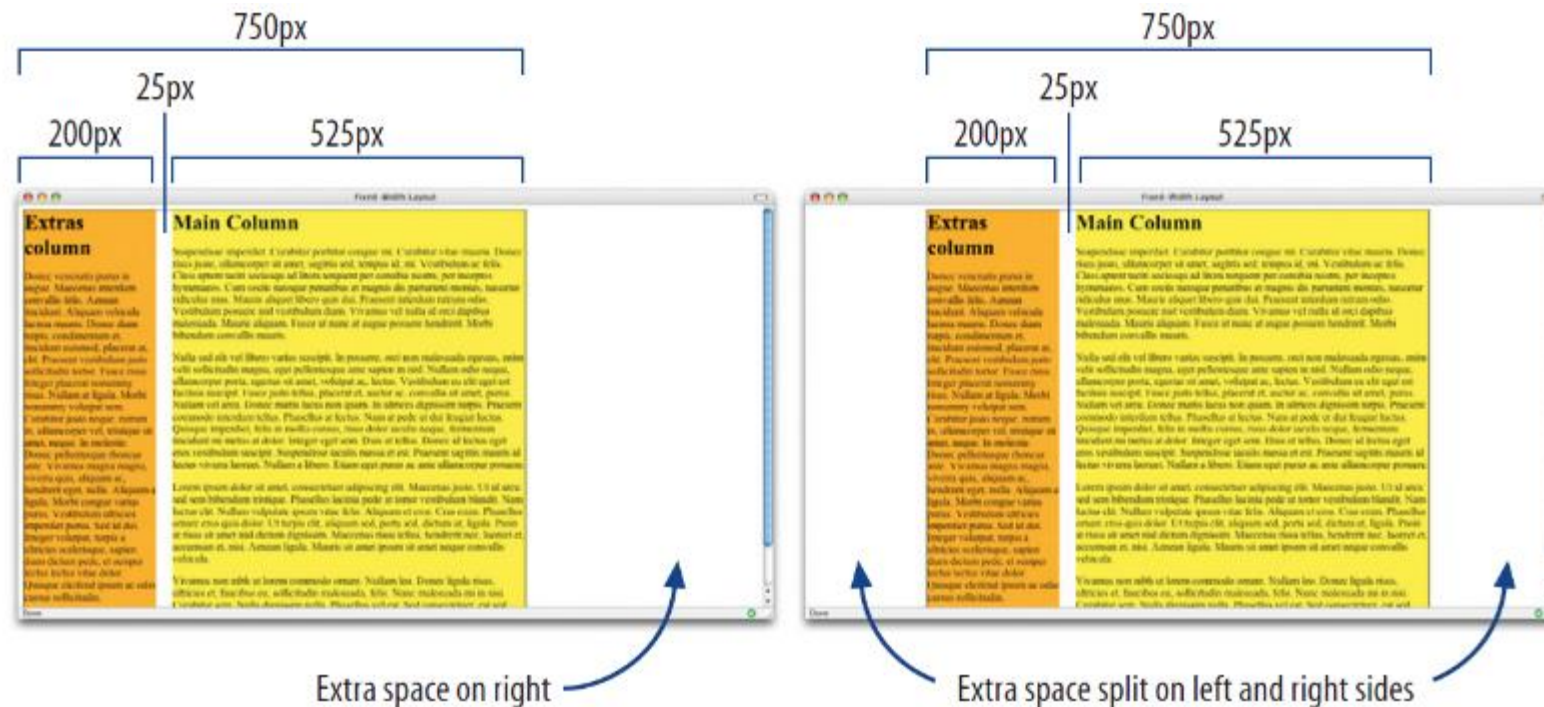
- **Fixed**
 - Page width is set to a specific size, usually pixels
 - Most common size is 960px wide
 - Why? Easier to design and build, predictable layout
- **Fluid**
 - Page expands & contracts in relation to browser window
 - Page dimensions usually set in %
 - Fills browser widow
- **Elastic**
 - Page resizes proportionally based on text size
 - 1em = 16px (default browser font size)
 - Why? Line breaks stays the same when expanding

FIXED LAYOUT

```
#wrapper {width: 750px;
position: absolute;
margin-left: auto;
margin-right: auto;
border: 1px solid black;
padding: 0px;}

#extras {position: absolute;
top: 0px;
left: 0px;
width: 200px;
background: orange;}

#main {margin-left: 225px;
background-color: yellow;}
```

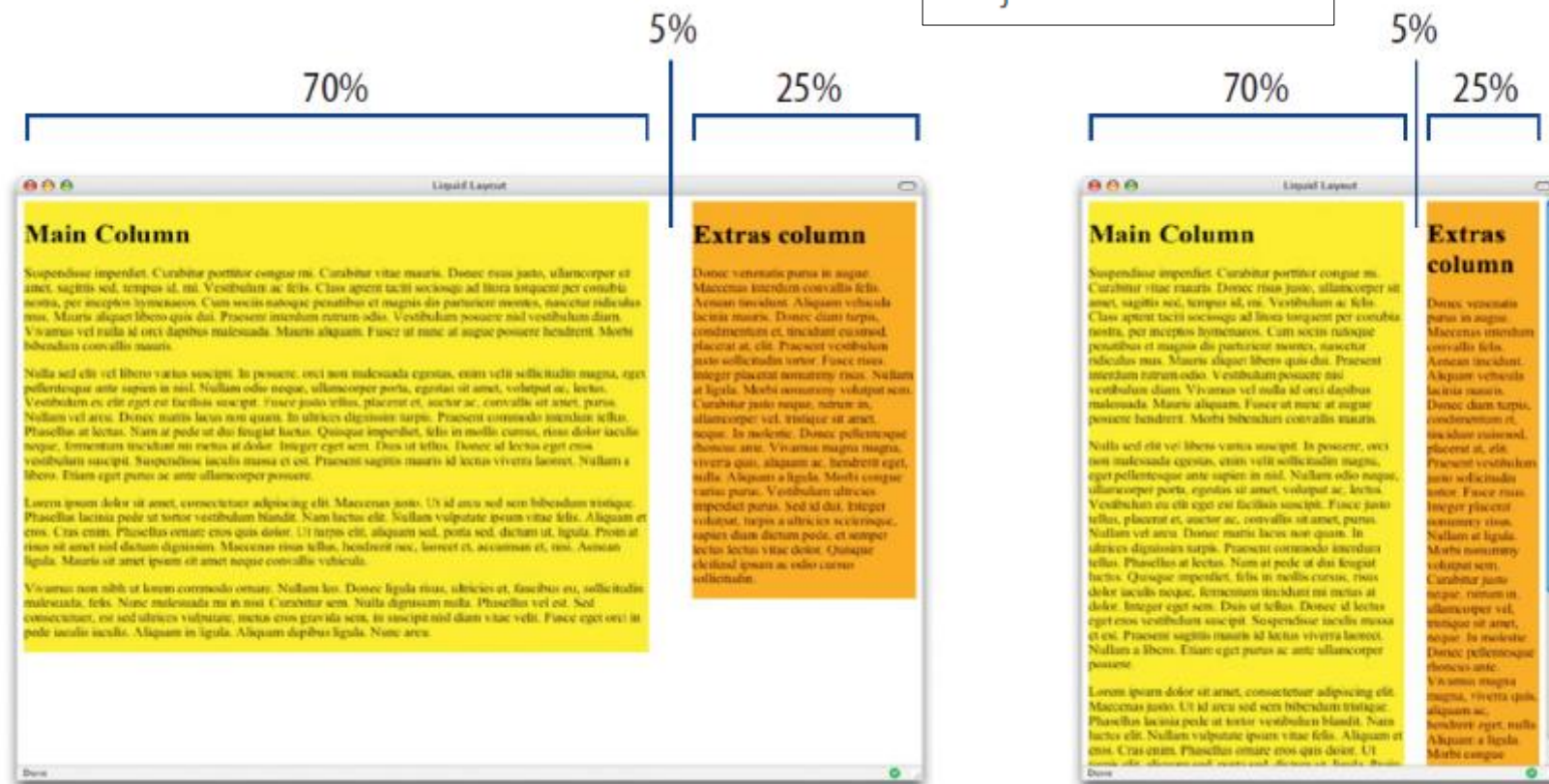


FLUID LAYOUT

Great Fluid Example

```
div#main {  
  width: 70%;  
  margin-right: 5%;  
  float: left;  
  background: yellow;  
}
```

```
div#extras {  
  width: 25%;  
  float: left;  
  background: orange;  
}
```



HYBRID OF ALL THREE

- **Fluid layout with css media query**
 - Set max size for screens bigger than 960px
- **Fixed layout with em or % on children.**
 - Easier than doing math
- **Main column fluid; additional column fixed**
 - Allows main column to fill extra space

CSS LAYOUT: FLOATS

TWO COLUMNS, FLUID LAYOUT

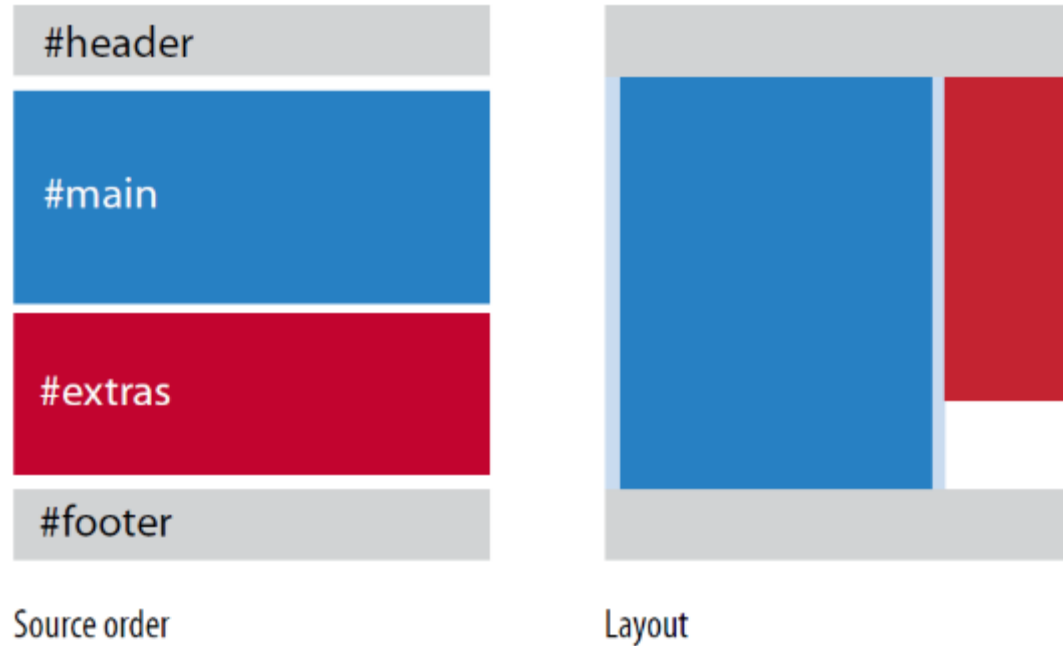
The markup

```
<div id="header">Masthead and headline</div>
<div id="main">Main article</div>
<div id="extras">List of links and news</div>
<div id="footer">Copyright information</div>
```

The styles

```
#main {
  float: left;
  width: 60%;
  margin: 0 5%;
}
#extras {
  float: left;
  width: 25%;
  margin: 0 5% 0 0;
}
#footer {
  clear: left;
}
```

TWO COLUMNS, FLUID LAYOUT



TWO COLUMNS: FIXED-WIDTH LAYOUT

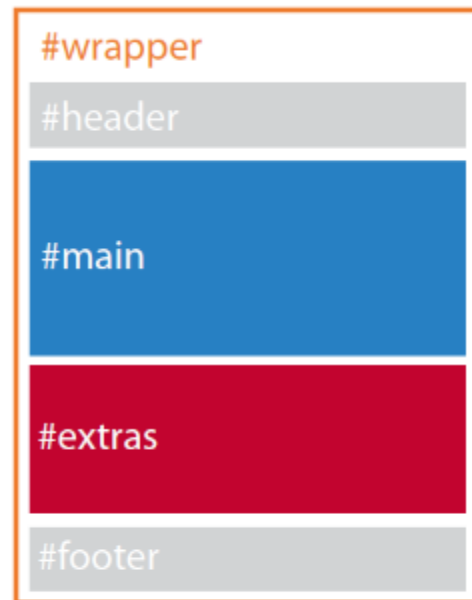
The markup

```
<div id="wrapper">
  <div id="header">Masthead and headline</div>
  <div id="main">Main article</div>
  <div id="extras">List of links and news</div>
  <div id="footer">Copyright information</div>
</div>
```

The styles

```
#wrapper {
  width: 960px;
}
#main {
  float: left;
  width: 650px;
  margin: 0 20px;
}
#extras {
  float: left;
  width: 250px;
  margin: 0 20px 0 0;
}
#footer {
  clear: left;
}
```

TWO COLUMNS, FIXED-WIDTH LAYOUT



Source order



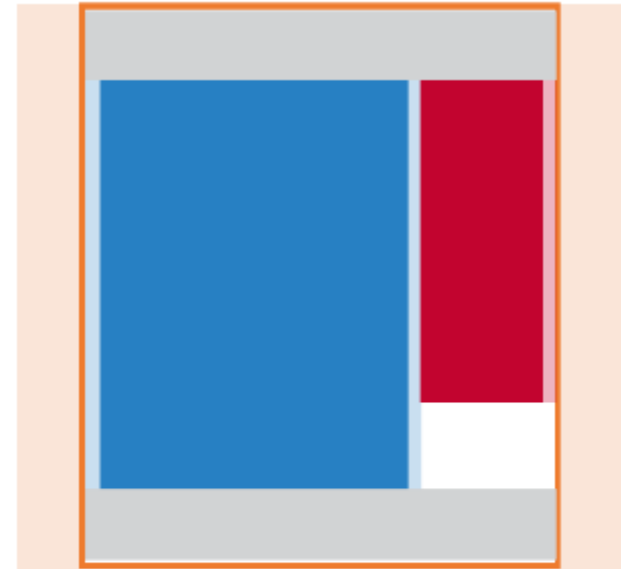
Layout

TWO COLUMNS: FIXED WIDTH, CENTERED

- Add this additional style:

The styles

```
#wrapper {  
  width: 960px;  
  margin: 0 auto;  
}
```



Layout

FULL-WIDTH HEADERS & FOOTERS

Take header and footer out of #wrapper

```
<div id="header">Masthead and headline</div>
<div id="wrapper">
  <div id="main">Main article</div>
  <div id="extras">List of links and news</div>
</div>
<div id="footer">Copyright information</div>
```

THREE COLUMNS: FLUID LAYOUT

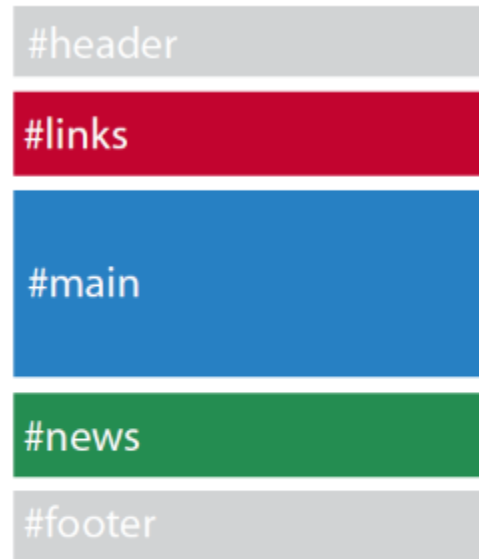
The markup

```
<div id="header">Masthead and headline</div>
<div id="links">List of links</div>
<div id="main">Main article</div>
<div id="news">News items</div>
<div id="footer">Copyright information</div>
```

The styles

```
#links {
  float: left;
  width: 22.5%;
  margin: 0 0 0 2.5%;
}
#main {
  float: left;
  width: 45%;
  margin: 0 2.5%;
}
#news {
  float: left;
  width: 22.5%;
  margin: 0 2.5% 0 0;
}
#footer {
  clear: left;
}
```


THREE COLUMNS: FLUID LAYOUT



Source order



Layout

NEGATIVE MARGINS: MIXED HTML ORDER

The markup

```
<div id="wrapper">
  <div id="header">Masthead and headline</div>
  <div id="main">Main article</div>
  <div id="news">News items</div>
  <div id="links">List of links</div>
  <div id="footer">Copyright information</div>
</div>
```

The styles

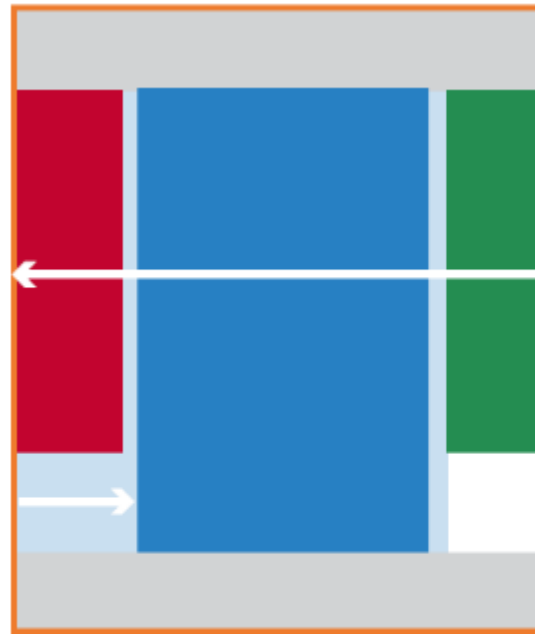
```
#wrapper {
  width: 960px;
  margin: 0 auto;
}
#main {
  float: left;
  width: 520px;
  margin-top: 0;
  margin-left: 220px;
  margin-right: 20px;
}
#news {
  float: left;
  width: 200px;
  margin: 0;
}
```

NEGATIVE MARGINS: MIXED HTML ORDER

```
#links {  
  float: left;  
  width: 200px;  
  margin-top: 0;  
  margin-left: -960px;  
}  
#footer {  
  clear: left;  
}
```



Source order



Layout

ITMD-362

RESPONSIVE DESIGN

RESPONSIVE DESIGN

- Use css media queries to conditionally load styles based on things like screen size
- Basic style sheet media query format:

```
@media screen and (min-width: 800px) {  
CSS Rules Go Here  
}
```

- Above rules will only apply to windows 800px or larger.
- Can also load linked stylesheet conditionally.

```
<link href="style.css" rel="stylesheet" media="screen and (min-  
width: 800px) ">
```

RESPONSIVE DESIGN

- Can also use with the css import directive
 - `@import url("file") screen and (min-width: 800px);`
 - Probably should avoid unless you understand how import works and may impact performance. Concurrent loading is one issue.
 - Google css import to read more.
- <http://mediaqueri.es/>
- <http://www.w3.org/TR/css3-mediaqueries/>
- https://developer.mozilla.org/en-US/docs/CSS/Media_queries
- <http://cssmediaqueries.com/>
- <http://bradfrost.github.com/this-is-responsive/resources.html>