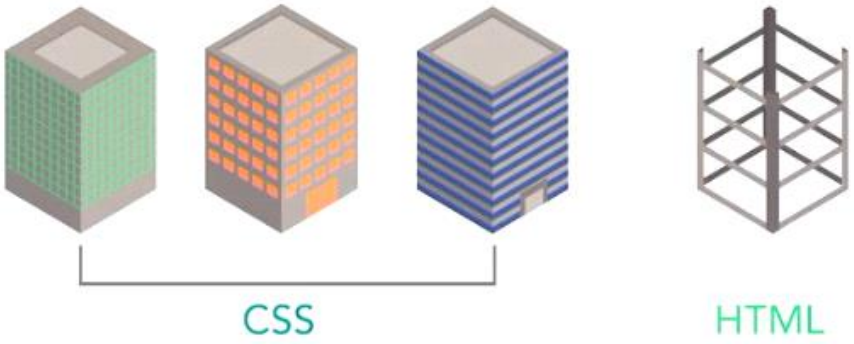
What is CSS?

Thursday, January 5, 2017 10:40 PM

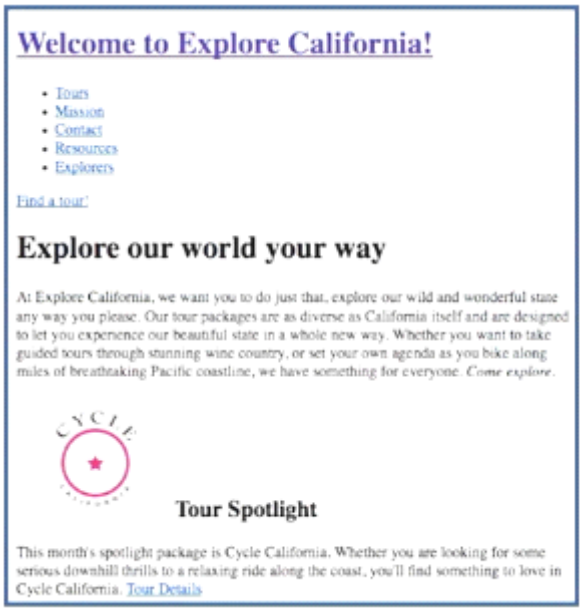
What is CSS?



- Style sheets are a collection of formatting rules.
- Styles are most commonly contained in external files.

Browser Default Styles

- HTML pages are always styled, but by default they are styled using browser default styles.



Welcome to Explore California!

- Tours
- Mission
- Contact
- Resources
- Explorers

Find a tour!

Explore our world your way

At Explore California, we want you to do just that, explore our wild and wonderful state any way you please. Our tour packages are as diverse as California itself and are designed to let you experience our beautiful state in a whole new way. Whether you want to take guided tours through stunning wine country, or set your own agenda as you bike along miles of breathtaking Pacific coastline, we have something for everyone. *Come explore.*

Tour Spotlight

This month's spotlight package is Cycle California. Whether you are looking for some serious downhill thrills to a relaxing ride along the coast, you'll find something to love in Cycle California. Tour Details

- So, when we style our HTML pages we override the default style sheets.

Class and ID Naming Conventions

- No whitespace or special characters
- Case sensitive
- Establish standards for your CSS and be consistent with them

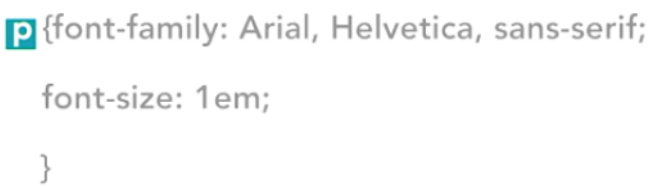
CSS syntax

Thursday, January 5, 2017 10:53 PM

- Css syntax is fairly simple and it only consits of tww things
  - Selectors
  - Declarations
- CSS is white space insensitive.

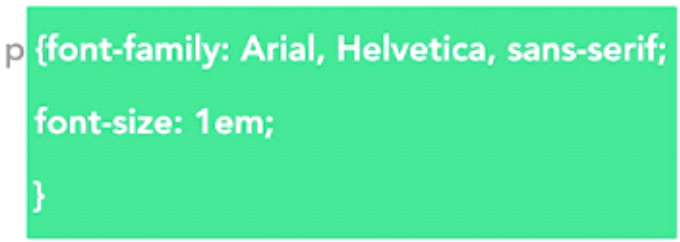
Selectors

- Selector tells CSS which thing to target while styling.



Declarations

- Declaration gives formatting rules to styles.



- Declaration consists of properties and values.

CSS Authoring Options

Friday, January 6, 2017 11:06 PM

Style Location

- Style rules can be located in three different ways, as follows

External Style Sheets


```
<html>
<head>
<link href="main.css" rel="stylesheet"
type="text/css" media="screen">
</head>
<body>
...
</body>
</html>
```

- Most efficent way of styling

Embedded Style Sheets


```
<html>
<head>
  <style>
    body {font-family: Arial; font-size: 100%;}
    h1, h1 {font-weight: normal; color: red;}
    p {font-size: .9em; margin-bottom: 1.2em;}
   .pullquote {background: yellow; padding:10px}
  </style>
</head>
<body>
```

- Only applicable to targets which are inside same file.

Inline Styles

index.htm

```
<html>
<head>
</head>
<body>
 <div>
 <p style="font-size: .9em; margin-bottom: 1.2em;">
...</p>
 </div>
</body>
</html>
```

Authoring Considerations

- Most projects will rely heavily on external styles.
- Embedded styles are mainly used to overwrite external styles.
- You should plan an overall site strategy for style placement.

How Browsers Apply Styles

Saturday, January 7, 2017 12:15 AM

- Browser starts applying styling from start of page and then overwrite using embedded and then overwrite any conflicting styling.
- **RULE OF THUMB - "The last rule applied wins"**
  (this rule is important because if externals are

NOTE - ONLY CONFLICTS WILL BE OVERWRITTEN.

Example 1- external overwrite embedded as external included after embedded.


```
<html>
<head>
 <style>
  body {font-family: Arial; font-size: 100%;}
  h1, h1 {font-weight: normal; color: red;}
  p {font-size: .9em; margin-bottom: 1.2em;}
  .pullquote {background: yellow; padding:10px}
 </style>
<link href="main.css" rel="stylesheet">
</head>
<body>
...
</body>
</html>
```

Example 2- a style of p overwrite other inside css file if conflict happens..


```
body {font-family: Arial; font-size: 100%;}
h1, h2 {font-weight: normal; color: red;}
p {font-size: .9em; margin-bottom: 1.2em;}
.pullquote {background: yellow; padding:10px}
div img {display: block; margin:0 auto;}
.specials p {color: red;}
li {margin-bottom: 2em;}
p {font-size: 1em; margin-bottom: 1.4em;}
```

Inheritance

```
body {font-family: Arial;
      font-size: 100%;
      color: blue;}
h1 {color: #900}
```

```
<html>
  <head>...</head>
<body>
<h1>Main heading</h1>
<p>This is body copy.</p>
<h2>This is a subheading</h2>
<p>More body copy</p>
</body>
</html>
```

Main heading

This is body copy.

This is a subheading

More body copy

- If style is applied on parent, child automatically inherits its styling....
- If another styling is written for child specifically , then it always overwrite Parent's, even if we have child's styling first and then comes parents.(specific is always give priority)
- This property is called **Specificity**.

Specificity

- In any conflict more specific rule wins.
- This table shows specificity points and more specific point wins.

NOTE-
Specificity rules apply when targets have different specificity rules, If they have same then last rule wins.

Selector	ID	classes	elements	specificity
body	0	0	1	1
#mainContent	1	0	0	100
quote	0	1	0	10
div p	0	0	2	2
#sidebar p	1	0	1	101

Basic Selector Types

Friday, January 6, 2017 10:40 PM

- Selector allows us to tell browser which element to focus for styling.
- We may wanted to apply same kind of styling to different element or want to style only a special class for an element, we can do this by different types of selectors.
- Following are some types of selectors.

The element Selector

The element selector selects elements based on the element name.

You can select all <p> elements on a page like this (in this case, all <p> elements will be center-aligned, with a red text color):

Example

```
p {
    text-align: center;
    color: red;
}
```

The id Selector

The id selector uses the id attribute of an HTML element to select a specific element.

The id of an element should be unique within a page, so the id selector is used to select one unique element!

To select an element with a specific id, write a hash (#) character, followed by the id of the element.

The style rule below will be applied to the HTML element with id="para1":

Example

```
#para1 {
    text-align: center;
    color: red;
}
```

The class Selector

The class selector selects elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the name of the class.

In the example below, all HTML elements with class="center" will be red and center-aligned:

Example

```
.center {
    text-align: center;
    color: red;
}
```

You can also specify that only specific HTML elements should be affected by a class.

In the example below, only <p> elements with class="center" will be center-aligned:

Example

```
p.center {
    text-align: center;
    color: red;
}
```

- Only those p comes which have '**center**' class.
- We can also select a elment having some specific classes

```
p.large.center {
    font-size: 30%;
}
```

* this will select <p> having 'large' and 'center' class both.

To group selectors, separate each selector with a comma.

In the example below we have grouped the selectors from the code above:

Example

```
h1, h2, p {
    text-align: center;
    color: red;
}
```

Combinator Selector

Monday, January 30, 2017 1:29 PM

CSS Combinators

A combinator is something that explains the relationship between the selectors.

There are four different combinators in CSS3:

- descendant selector (space)
- child selector (>)
- adjacent sibling selector (+)
- general sibling selector (~)

Descendant Selector

The descendant selector matches all elements that are descendants of a specified element.

The following example selects all <p> elements inside elements:

Example

```
div p {
    background-color: yellow;
}
```

- Only inside relationship is required.

Child Selector

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

The following example selects all <p> elements that are immediate children of a element:

Example

```
div > p {
    background-color: yellow;
}
```

- Immediate children relationship is required.
- Parent > child, parent is greater than child.

Adjacent Sibling Selector

The adjacent sibling selector selects all elements that are the adjacent siblings of a specified element.

Sibling elements must have the same parent element, and "adjacent" means "immediately following".

The following example selects all <p> elements that are placed immediately after elements:

Example

```
div + p {
    background-color: yellow;
}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
div + p {
    background-color: yellow;
}
</style>
</head>
<body>

<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
</div>

<p>Paragraph 3. Not in a div.</p>
<p>Paragraph 4. Not in a div.</p>

</body>
```

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3. Not in a div.

Paragraph 4. Not in a div.

- Only <p> which comes immediately after will be selected.
- Both and <p> are sibling here child of <body>

General Sibling Selector

The general sibling selector selects all elements that are siblings of a specified element.

The following example selects all <p> elements that are siblings of elements:

Example

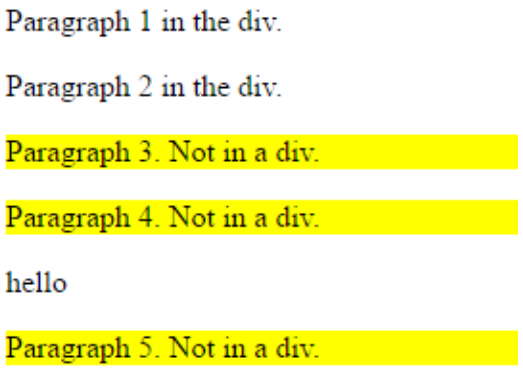
```
div ~ p {
    background-color: yellow;
}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
div ~ p {
    background-color: yellow;
}
</style>
</head>
<body>

<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
</div>

<p>Paragraph 3. Not in a div.</p>
<p>Paragraph 4. Not in a div.</p>
<a> hello</a>
<p>Paragraph 5. Not in a div.</p>

</body>
```



- Here all <p> which are sibling of will be selected.

Css element Selector

Monday, January 30, 2017 12:57 PM

CSS [attribute] Selector

The `[attribute]` selector is used to select elements with a specified attribute.

The following example selects all <a> elements with a target attribute:

Example

```
a[target] {
    background-color: yellow;
}
```

CSS [attribute="value"] Selector

The `[attribute="value"]` selector is used to select elements with a specified attribute and value.

The following example selects all <a> elements with a target="_blank" attribute:

Example

```
a[target="_blank"] {
    background-color: yellow;
}
```

For more - www.w3schools.com/css/css_attribute_selectors.html

Selector Refrence

Monday, January 30, 2017 12:57 PM

Selector	Example	Example description	CSS
.class	.intro	Selects all elements with class="intro"	1
#id	#firstname	Selects the element with id="firstname"	1
*	*	Selects all elements	2
element	p	Selects all <p> elements	1
element,element	div, p	Selects all elements and all <p> elements	1
element element	div p	Selects all <p> elements inside elements	1
element>element	div > p	Selects all <p> elements where the parent is a element	2
element+element	div + p	Selects all <p> elements that are placed immediately after elements	2

- Clearly in **div p , p is inside div**

For more -www.w3schools.com/cssref/css_selectors.html

Introduction and options

Wednesday, January 25, 2017 4:16 PM

CSS Font Options

- Allow the browser to display its default font
- Specify a system font and provide fallback options
- Use @font-face to point to a hosted font resource

Fonts requested in the CSS had to be installed on the client machine.

- So if 'Garamond' font is not available on client machine(because different m/c have different system fonts), then fallback will occur and machine default will be used.
- We can request several in for covering up fallbacks(all will be system fonts)

Requesting Fonts

```
h1 {
font-family: Arial, Helvetica, Verdana, sans-serif;
}
```

- Here designer requesting multiple fonts, apart from first all are fallback fonts, so if first is not available second is searched and so on.

Web Fonts

- We may wanted to use some other fonts and do not want to worry about whether these fonts are there in client's machine or not .
- Web fonts will be downloaded along with the page.

Property	Description	CSS
@font-face	A rule that allows websites to download and use fonts other than the "web-safe" fonts	3

Web fonts refer to the technique of having the browser download and install fonts requested in the page's styles, using the @font-face syntax.

@Font-Face Syntax

```
@font-face {
  font-family: 'fontName';
  src: url('fontName.eot');
  src: url('fontName.eot?#iefix') format('embedded-opentype'),
       url('fontName.woff') format('woff'),
       url('fontName.ttf')  format('truetype'),
       url('fontName.svg#svgFontName') format('svg');
  }
```

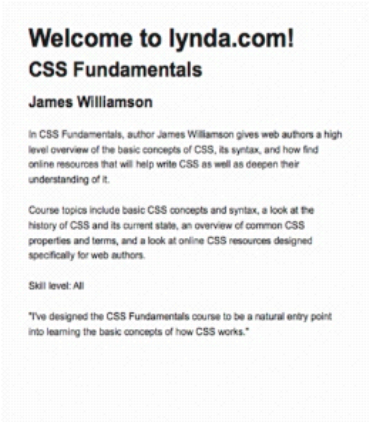
Formatting text

Wednesday, January 25, 2017 4:56 PM

- We can use **font-family** property to set a family for font, we can also provide fallbacks.

Defining Font Families

```
body {
font-family: Georgia,
"Times New Roman", Times, serif;
}
```



- NOTE - "Times New Roman" is inside double quote because it contains space in its name.

Defining Font Size

```
h1 {
font-size: <absolute-size> | <relative-size> | <length>
| <percentage> | inherit
}
```

- We can use any of given above options, **Inherit** means default value for text.
  'inherit' is used when we wanted to override a style to system default.

Fixed Units vs. Relative Units

- We can use two types of size here
  - **Absolute size (px)** - fixed on all devices.
  - **Relative size(em)** - relative to default size of browser

  NOTE- so 1em means exactly same as default size of browser on any device.

Fixed Units vs. Relative Units

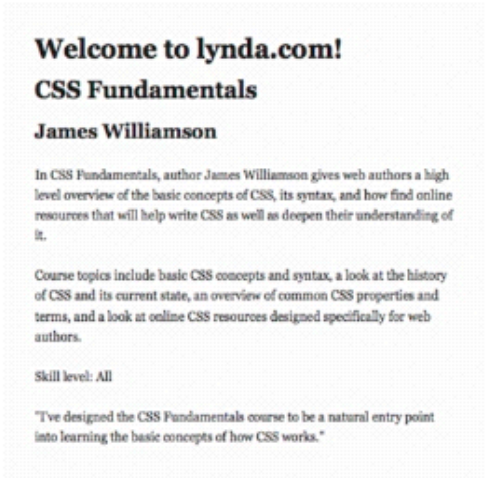
- Fixed units are displayed at the requested size, regardless of device or context.
- Relative units are displayed relative to the environment in which they are found.


```
h1 { font-size: 1.6em; }

h2 { font-size: 1.4em; }

h1 { font-size: 1.2em; }

p { font-size: 1em; }
```



Welcome to lynda.com!

CSS Fundamentals

James Williamson

In CSS Fundamentals, author James Williamson gives web authors a high level overview of the basic concepts of CSS, its syntax, and how find online resources that will help write CSS as well as deepen their understanding of it.

Course topics include basic CSS concepts and syntax, a look at the history of CSS and its current state, an overview of common CSS properties and terms, and a look at online CSS resources designed specifically for web authors.

Skill level: All

"I've designed the CSS Fundamentals course to be a natural entry point into learning the basic concepts of how CSS works."

- Here one thing to note is that using **em** we have also established relation between size of h1 and h2 and so on.

Font-Weight and Font-Style

```
h1 {
font-weight:  normal | bold | bolder | lighter | 100 | 200 | 300 | 400 |
500 | 600 | 700 | 800 | 900 | inherit;
font-style: normal | italic | oblique | inherit
}
```

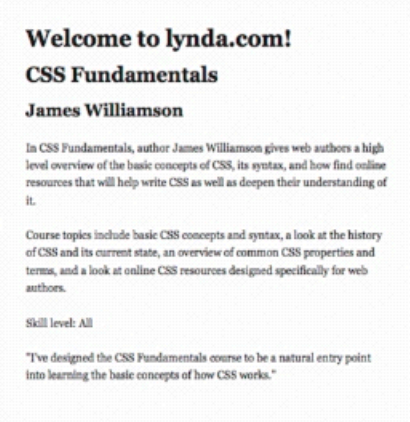
- 'Font-weight' can be from 100 to 900 (some constants are also there)
- 'font-style' can be any of four (inherit - use same as parent, if parent not have then use by default)

Setting Bold and Italicized Text

```
span { font-weight: bold; }

.title { font-weight: normal;
        font-style: italic; }

.author { font-size: 1.2em; }
```



Welcome to lynda.com!

CSS Fundamentals

James Williamson

In CSS Fundamentals, author James Williamson gives web authors a high level overview of the basic concepts of CSS, its syntax, and how find online resources that will help write CSS as well as deepen their understanding of it.

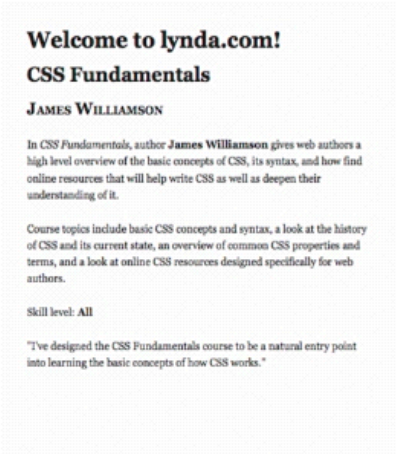
Course topics include basic CSS concepts and syntax, a look at the history of CSS and its current state, an overview of common CSS properties and terms, and a look at online CSS resources designed specifically for web authors.

Skill level: All

"I've designed the CSS Fundamentals course to be a natural entry point into learning the basic concepts of how CSS works."

Font-Variant

```
h3 {font-variant: small-caps;}
```



Welcome to lynda.com!

CSS Fundamentals

James Williamson

In *CSS Fundamentals*, author **James Williamson** gives web authors a high level overview of the basic concepts of CSS, its syntax, and how find online resources that will help write CSS as well as deepen their understanding of it.

Course topics include basic CSS concepts and syntax, a look at the history of CSS and its current state, an overview of common CSS properties and terms, and a look at online CSS resources designed specifically for web authors.

Skill level: **All**

"I've designed the CSS Fundamentals course to be a natural entry point into learning the basic concepts of how CSS works."

font-variant	Specifies whether or not a text should be displayed in a small-caps font

- Small-caps allow us to have small capital letters(in place of lowercase).
- Font-variant has only small-caps feature.

Text-transform

```
h2 {text-transform: uppercase;}

h3 {font-variant: small-caps;}
```

- 'Text-transform' can transform our text to lowercase, uppercase, capitalize.(all these three are possible)

```
p.uppercase {
    text-transform: uppercase;
}

p.lowercase {
    text-transform: lowercase;
}

p.capitalize {
    text-transform: capitalize;
}
```

Text-align

```
p {text-align: justify;}

.quote {text-align: center;}
```

- Controls horizontal allignment for text, following alignment are possible

Value	Description
left	Aligns the text to the left
right	Aligns the text to the right
center	Centers the text
justify	Stretches the lines so that each line has equal width (like in newspapers and magazines)
initial	Sets this property to its default value. Read about *initial*
inherit	Inherits this property from its parent element. Read about *inherit*

- Justify makes each line of paragraph of equal size.(if they are not)

Text-indent

```
p {text-indent: 1em;}
```

Welcome to lynda.com!

CSS Fundamentals

James Williamson

In CSS Fundamentals, author James Williamson gives web authors a high level overview of the basic concepts of CSS, its syntax, and how find online resources that will help write CSS as well as deepen their understanding of it.

Course topics include basic CSS concepts and syntax, a look at the history of CSS and its current state, an overview of common CSS properties and terms, and a look at online CSS resources designed specifically for web authors.

Skill level: All

"I've designed the CSS Fundamentals course to be a natural entry point into learning the basic concepts of how CSS works."

The text-indent property specifies the indentation of the first line in a text-block.

Note: Negative values are allowed. The first line will be indented to the left if the value is negative.

Letter-spacing

```
h2 {letter-spacing: .1em;}
```

- We can set custom distance between words using letter-spacing.
- We can also set value to negative to decrease.

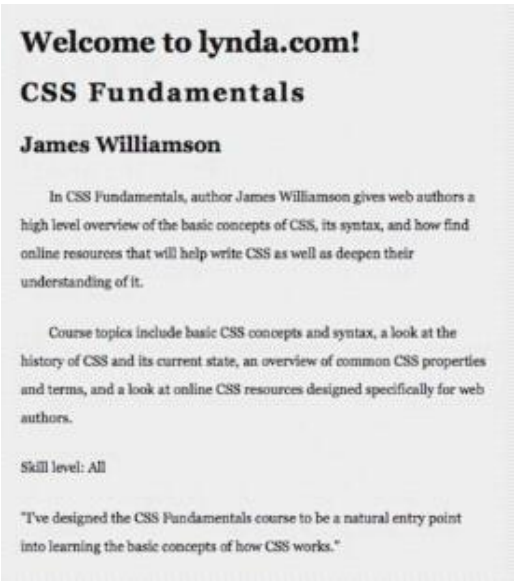
=>

Letter-height

```
p {line-height: 1.6;}
```

- It can be used to control space between multiple lines.

=>



Color:Units of measurement

Sunday, January 29, 2017 12:28 PM

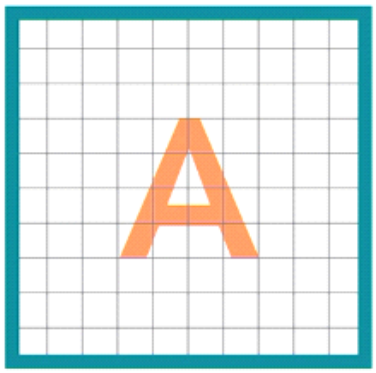
Absolute Values

- Used when physical properties of an element is known.

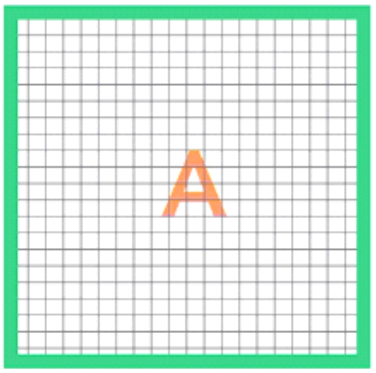
in	inches
cm	centimeters
mm	millimeters
q	quarter millimeters
pt	points
pc	picas
px	pixels

Pixels

Device pixels



Low resolution



High resolution

- Higher Resolution devices will have smaller pixels.
- So if we display the same text, image on different screens size will be different.
- To resolve this problem for designers WC3 created **REFRENCE PIXEL,** but it could not solve problem entirely.

Relative Values

- em ems
- rem root ems
- ex exes
- ch character
- vw viewport width
- vh viewport height
- vmin viewport minimum
- vmax viewport maximum

EMS

When used with font-size, 1em is equal to the default font size for the device.

```
body {font-size: 100%;}
h1 {font-size: 1.6em;}
h2 {font-size: 1.4em;}
h3 {font-size: 1.2em;}
p {font-size: 1em;}
```

for font-size

- for font-size em takes its size relative to the size of its parent.(if no parent then it will use system default)
- So we set font-size of parent and use that to use relative in our other elements.

```
body {font-size: 100%;}
h1 {font-size: 1.6em;}
h2 {font-size: 1.4em;}
h3 {font-size: 1.2em;}
p {font-size: 1em;}
```

- Here we have set default font-size of body, and then set size h1, 1.6 times of body fonts.

for others properties

- For other like **margins,** the calculation is diffent.
- For others **em i**s relative to the font-size of that elements itself.

Example - let device size is 16px.

```
h1 {                          device font size: 16px
  font-size: 2em;             2 x 16 = 32px font size
  margin-bottom: 1em;         32px bottom margin
}
```

- Here size of font-size = 2em = 2*device-size =32px
  But margin-bottom =1em=1 * font-size of that element =32px

NOTE- other relative units see video

Percentages

Percentage values are calculated relative to their parent element. A div with a width of 80% would use 80% of its parent element, while a paragraph with its font-size set to 80% would size the text at 80% of the size of its parent text.

- Percentage always work according to it parents.

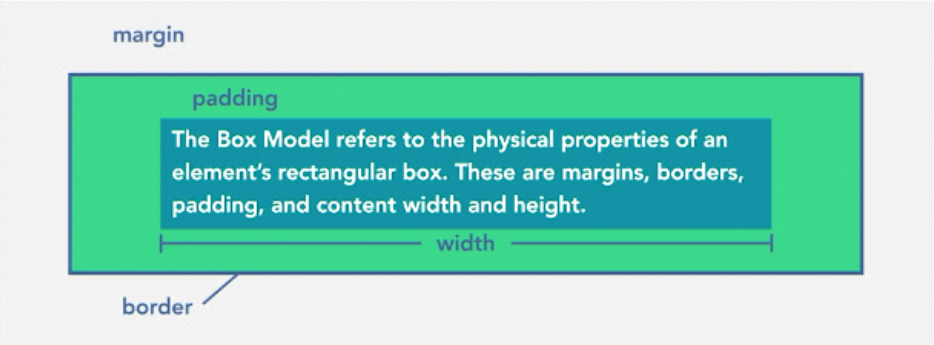
Introduction

Thursday, January 26, 2017 3:30 PM

The Box Model

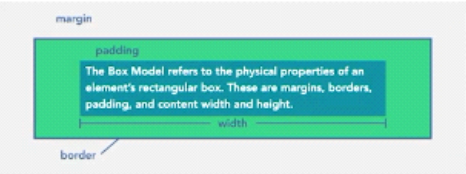
The Box Model refers to the physical properties of an element's rectangular box. These are margins, borders, padding, and content width and height.

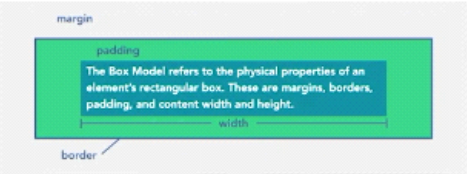
- Every element is considered to be a box and have following property corresponding to it.
  - Width
  - Height
  - Border
  - Padding
  - Margin



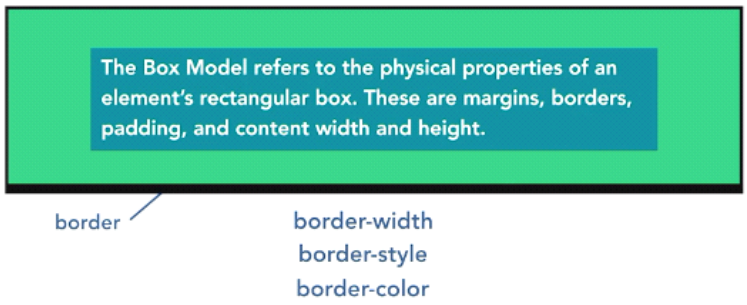
Margin -

- Margin represents space around an element.
- Margin do not indicate final width of an element but they are useful when two elements placed adjacent To each other, the space between the elements is decided by margin.





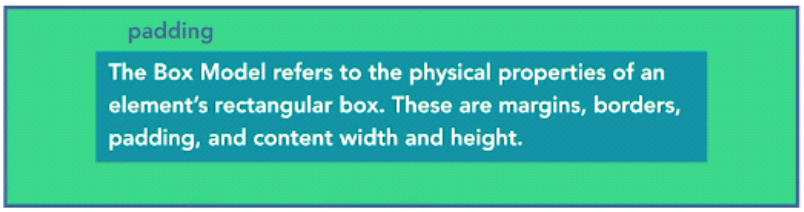
Borders



- Width of an element includes everything except margin, so borders included in final width of an element.

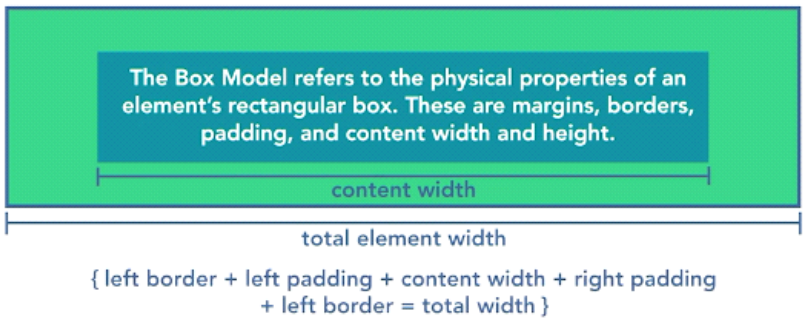
Padding

- Padding is space added to element inside border, same as soft material added around fragile equipment to keep if safe from breaking.



Width and Height

- Total width or final width is different from width of an element, same for height.
- Width refers to content width of element



Box-sizing

- For removing the confusion of width we can set 'Box-sizing' CSS property to
  - Border-box - if this is **set then width means total-width**
  - Content-box (default)- width means content-width.

```
.box {
width: 300px;
padding: 10px;
border: 1px solid black;
box-sizing: content-box;
}
```

total width = 322px

```
.box {
width: 300px;
padding: 10px;
border: 1px solid black;
box-sizing: border-box;
}
```

total width = 300px

Border-box => include everything up to border in width.
Content-box => include only content(element) in width

Box Model Considerations

- If a property is not declared, you can't assume the value is 0.
- Elements often have default margins that you need to account for.

- We shouldn't always assume the default value to be 0, because they may be different.

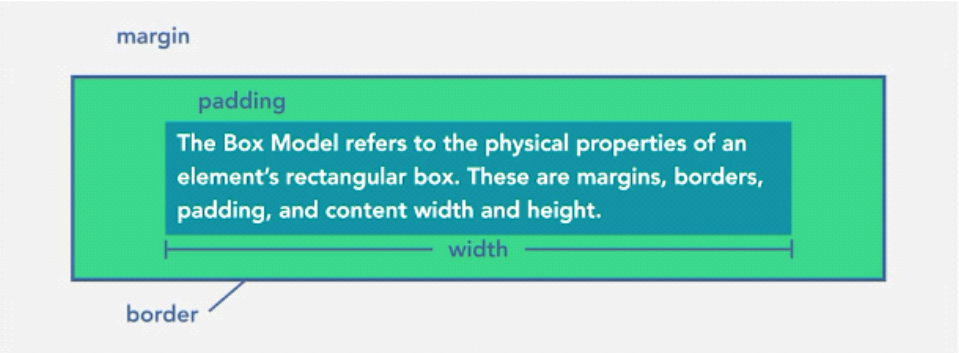
- A 100% width property when combined with padding and borders can create elements that are larger than their parent.

- If we set 100% property and we have not used 'border-box' property then ,
  totalsize= 100% + padding(left and right) + border(left and right)
  Which may increase the size of parent, and CSS will break;

padding

Thursday, January 26, 2017 3:58 PM

- Padding is space between content of element and border.



Padding SYNTAX

Individual properties

- padding-top: //values;
- padding-right://values;
- padding-bottom://values;
- padding-left://values;

Shorthand notation

- padding: 10px 20px 15px 10px;
- padding: 10px 20px 15px;
- padding: 10px 20px;

Short hand Notation syntax-

- **FOUR VALUES** - In shorthand notation we go from top and in clockwise direction.
  - Syntax - padding: **top right bottom left**
- **THREE VALUES** - first will be applied to **top** , second to **left-right** and third to **bottom**
- **TWO VALUES** - first will be applied to **top-bottom,** and second value will be applied to **left-right.**
- **ONE VALUE-** value will be applied to all four sides.

Padding Usage

"This is a single element with a background color and border applied. Padding is used to keep the text away from the edge of the element creating a pull quote"

- They are used to keep the actual content away from the edges, like in example shown. here we apply padding so the content do not touch with edges.

Margin

Thursday, January 26, 2017 4:13 PM

Margins

- Margins represent the space between elements.
- Margin values are not calculated as part of an element's total width.
- Most elements have a default margin you must account for.
  - Unlike, padding most elements have default margins.

Margin SYNTAX

Individual properties

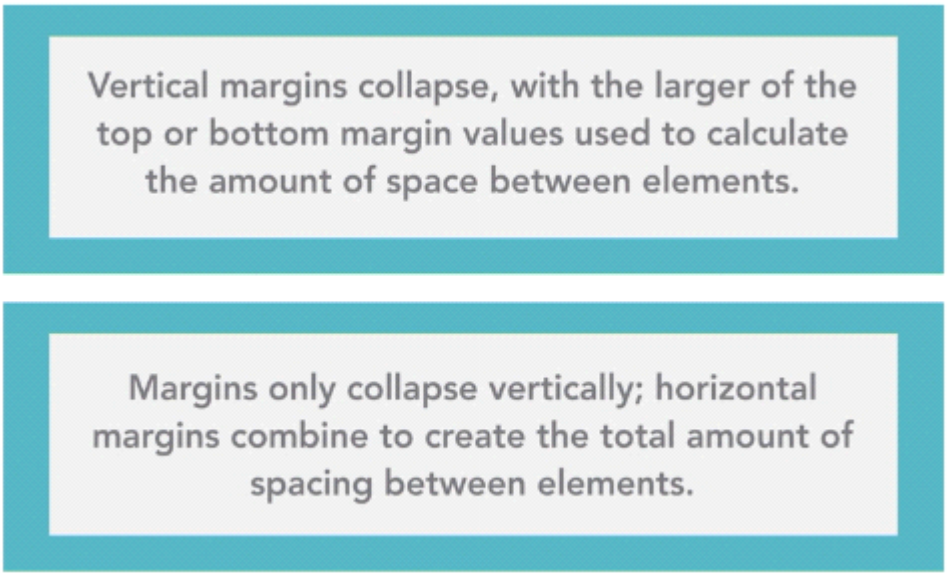
- margin-top: //values;
- margin-right://values;
- margin-bottom://values;
- margin-left://values;

Shorthand notation

- margin: 10px 20px 15px 10px;
- margin: 10px 20px 15px;
- margin: 10px 20px;
- margin: 10px;

- Syntax is exactly same as **padding**.

Margin Consideration



- It means if have upper element with bottom-margin:60px and bottom element with top-maring:80px, total margin=max(60,80) and not 60+80.
- This happens with vertical margins only.

Border

Thursday, January 26, 2017 7:33 PM

Border SYNTAX

Name: Border
Value: <border-style> || <border-width> || <border-color>

Individual properties

```
border-top-style: solid;
border-right-style: solid;
border-bottom-style: solid;
border-left-style: solid;

border-top-weight: 1px;
border-right-weight: 1px;
border-bottom-weight: 1px;
border-left-weight: 1px;

border-top-color: black;
border-right-color: black;
border-bottom-color: black;
border-left-color: black;
```

Shorthand notation

```
border-top: 1px solid black;
border-right: 1px solid black;
border-bottom: 1px solid black;
border-left: 1px solid black;

border: 1px solid black;
```

- If have used **border** only then it will apply on all side borders.

Borders Width

```
p { padding: 20px;
    border: 1px solid black;
    width: 200px;}
```

Total Width:

1px + 20px + 200px + 20px + 1px

Border Styles

Border Color

The border-color property can be specified independently of foreground color and background color.

```
border-color: green;
```

Backgrounds Intro

Thursday, January 26, 2017 7:45 PM

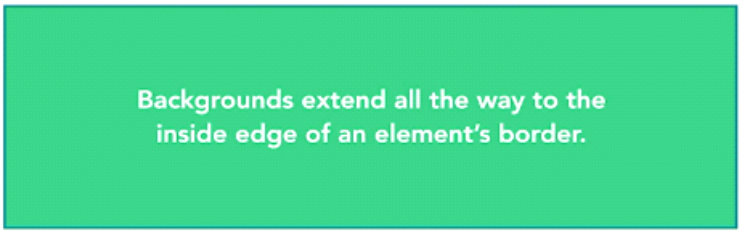
Backgrounds

- Background property allows us to modify default system backgrounds.

Background SYNTAX

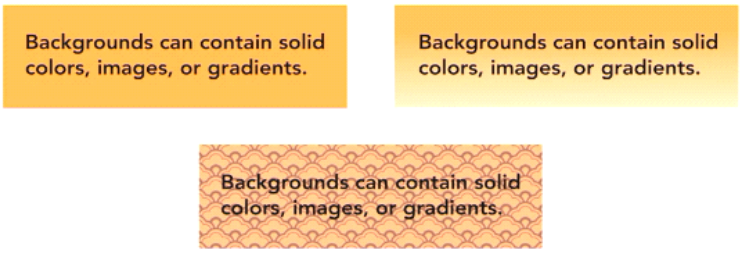
```
div {
background-color: #ff0;
background-image: url(flower.png), url(ball.png), url(grass1.png);
background-position: center center, 20% 80%, top left;
background-origin: border-box, content-box, border-box;
background-repeat: no-repeat, no-repeat, no-repeat;
}
```

Background Area



- Backgrounds include everything inside borders.

Background Options



Background Options

- Take this as sample and now we look at the all possible properties we can apply to change look of this page.

background-color

background-image

- We can apply solid background color.
- Element-box is the box of an element(up to border

Element box

background-position

Element box

background-position

Element box

background-repeat

Element box

background-size

Element box

background-clip

Element box

background-color
background-image 1

- If both background-color and background-image used , then background image comes above background-color.

Element box

background-color
background-image 1
background-image 2

- We can use the place where other backgroud image is not there.

Positioning

Sunday, January 29, 2017 6:47 PM

- Positioning allows us to position an element in page relative to
  - Original starting positon
  - Other element

CSS Positioning Schemes

- Normal flow
- Element floating
- Absolute positioning

Normal Document Flow

- Normal Flows is same as default positioning but it is powerful and can be used sometimes.
- Normal Document Flow takes all elements in the order they found and stack them.
  Stack means place them in order they are in document.
- Normal Flow do most of our work automatically and we do not have to do anything.

```
<html>
<head>....</head>
<body>
  <h1>Welcome to lynda.com!</h1>
  <h2>CSS Fundamentals</h2>
  <h3>James Williamson</h3>
<p>In CSS Fundamentals, author James
Williamson gives web authors a high level overview
of the basic concepts of CSS, its syntax, and how
find online resources that will help write CSS as
well as deepen their understanding of it.</p>
  <p>Course topics include basic CSS concepts
and syntax, a look at the history of CSS and its
current state, an overview of common CSS
properties and terms, and a look at online CSS
resources designed specifically for web
authors.</p>
  <p class="skill">Skill level: All</p>
</body>
</html>
```

CHANGING POSITION OF AN ELEMENT

- We can use Position property to change position of an element dramatically.

Position

Name: position
Value: static | relative | absolute | fixed | inherit

Static - position a/c to normal flow(Default value)

Relative Positioning

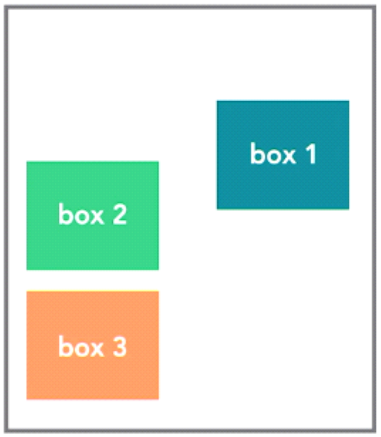
- Relative Positioning Allows us to tweak element position according to offset value given
- Relative position is a part of normal Flow, so the space for that element is left behind.

An element with `position: relative;` is positioned relative to its normal position.

Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.

- JAHA PAR HOTA USSE RELATIVE ME

```
.box1 {position: relative;
      left: 100px;
      top: 50px;
}
```



- Here position is relative to the offset(top-left) corner of the page.
  - Left means x axis.
  - Top means y axis.
- In relative positioning the space left by the positioned element is kept empty .

Absolute Positioning

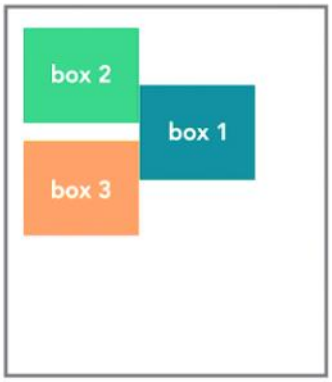
- Here position is relative to the offset(top-left) corner of the page.
  - Left means x axis.
  - Top means y axis.
- In absolute positioning remove the element from normal Flow, so this element is not given space by browser , so no space is left for it and element after it comes upward.

An element with `position: absolute;` is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).

However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

- APNE NEAREST KE RELATIVE ME

```
.box1 {position: absolute;
      left: 100px;
      top: 50px;
}
```

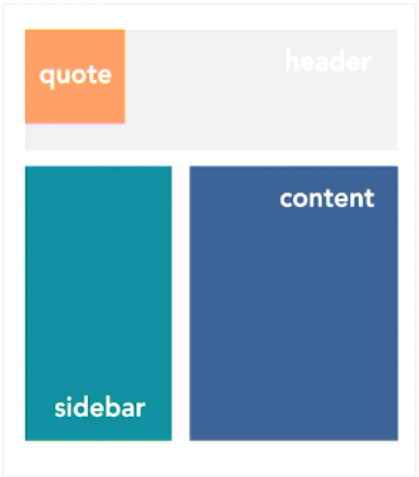


- **NOTE - For absolute Positioning a child can be positioned inside parent, only when parent is also positioned(in any way), so if parent is not positioned then child will not positioned properly.**

Example - 1 if parent is not positioned , quote is child of content.

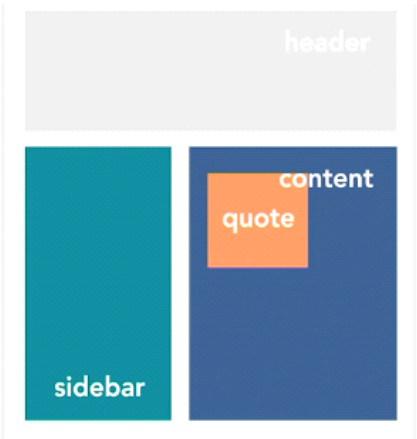
```
.content {

}
.quote {
    position: absolute;
    top: 20px;
    left: 40px;
}
```



Example - 2 if parent is positioned , quote is child of content.

```
.content {
    position: relative;
  }
.quote {
    position: absolute;
    top: 20px;
    left: 40px;
}
```



Note - We will see that relative-absolute parent-child relationship technique a lot in positioning.

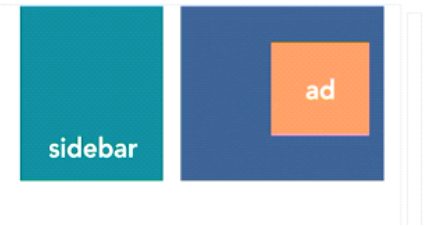
Fixed Positioning

- In Fixed positioning elements are absolutely positioned , but they are always positioned relative

to active viewport(viewport means what we see).

- So the element will be fixed always.

```
.ad {
    position: fixed;
    top: 50px;
    right: 100px;
}
```



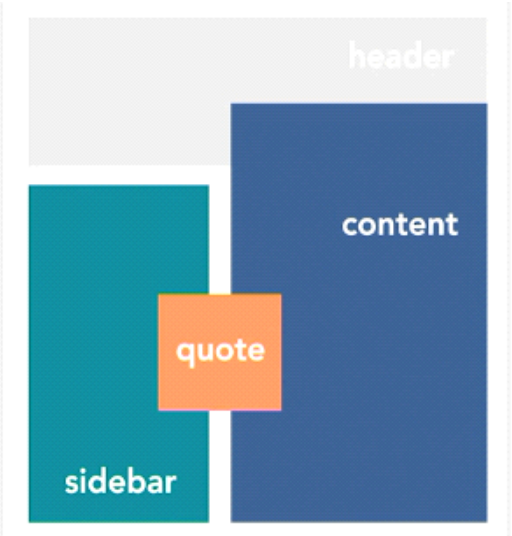
Element Stacking

Monday, January 30, 2017 12:13 AM

- By default positioned element(by position property) will be come over unpositioned.

- Stacking means placing elements one over another.
- Stacking done controlled by Z-index, so more z-index element will be shown up.

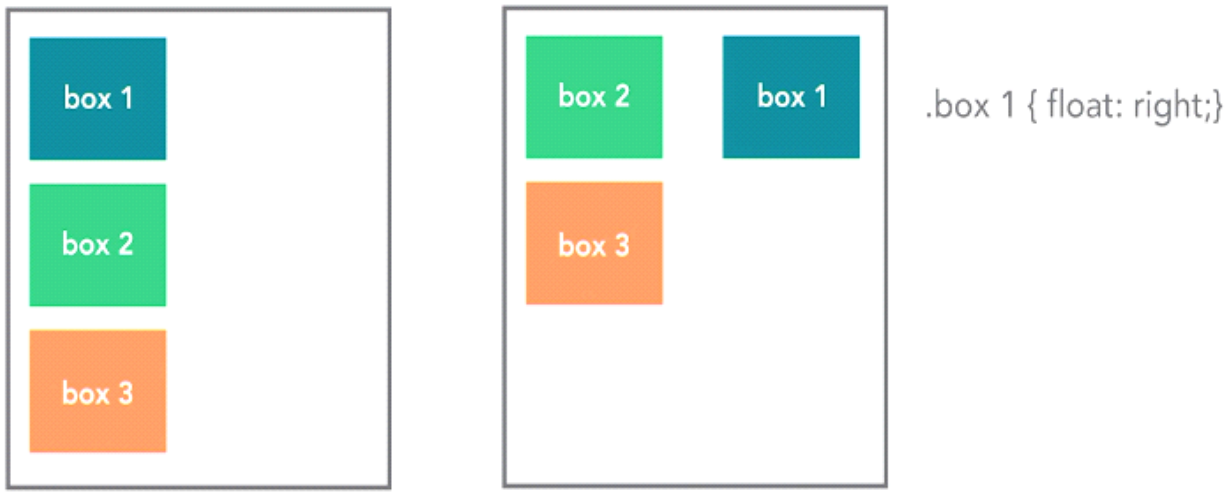
```
.content {
    z-index: 1;
}
.quote {
    z-index: 2;
}
```



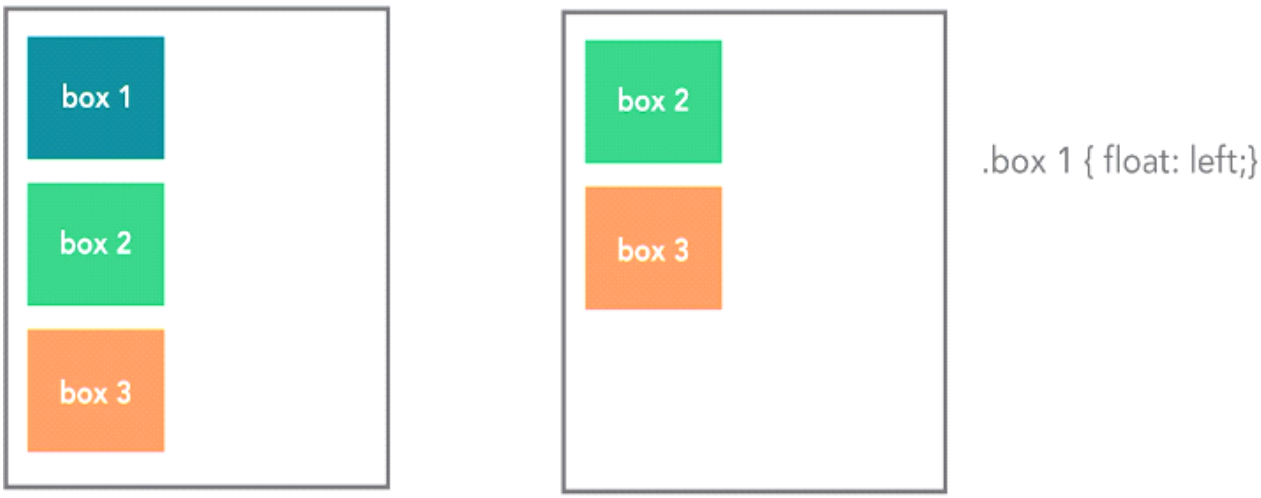
Element Floating

Monday, January 30, 2017 12:13 AM

- Float element is element which is being shifted to left or right.
- Float property is used to assign float
- Default is left float.
- When a float is assigned to an element it comes out of normal flow(like absolute positioning and other element takes it position.



- So, when we float an element to left, it gets out of normal flow and overlapped by others.



FLOATING with Image and TEXT

In CSS Fundamentals, author James Williamson gives web authors a high level overview of the basic concepts of CSS, its syntax, and how find online resources that will help write CSS as well as deepen their understanding of it.

In CSS Fundamentals, author James Williamson gives web authors a high level overview of the basic concepts of CSS, its syntax, and how find online resources that will help write CSS as well as deepen their understanding of it.

```
img { float: left;}
```

- An important property of float is that when it is done with image and having text beneath it then text try to wrap around it and not override it.
  this property is used many developers to style text around an image.
- This property works in both float(left and right)

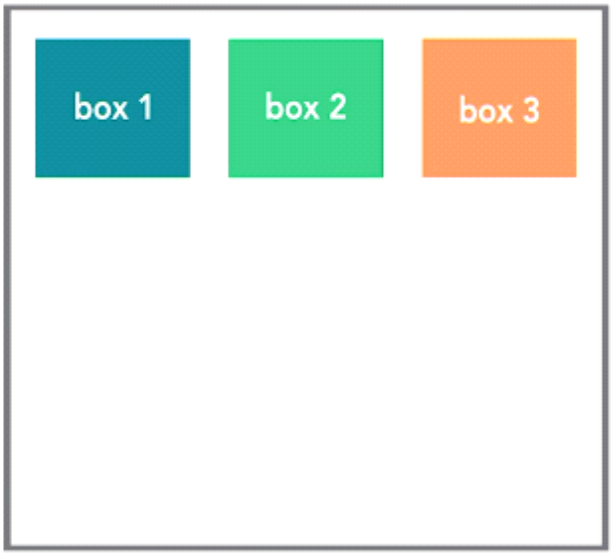
Floating techniques

Monday, January 30, 2017 12:32 AM

Stacking Horizontally

- We can stack element horizontally by apply float on all of them.

```
div {float: left;}
```



Clearing Floats

- Sometime we do not wanted elements beneath a floating to come up and take its space.
- We can do this by clearing that element's **FLOAT**;

The clear Property

The `clear` property is used to control the behavior of floating elements.

Elements after a floating element will flow around it. To avoid this, use the `clear` property.

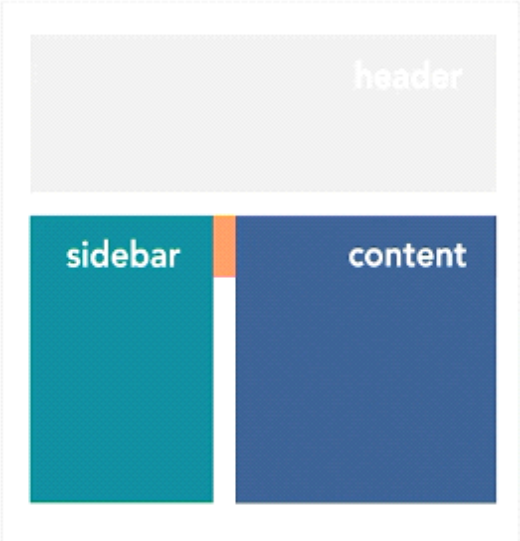
The `clear` property specifies on which sides of an element floating elements are not allowed to float:

- Possible value
  - Left
  - Right
  - both

Example 1 - without float clear

```
.content { float: right; }

.sidebar { float: left; }
```

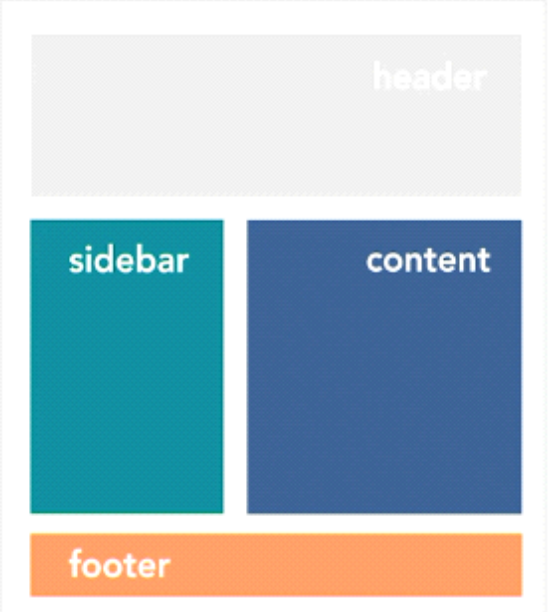


Example 2 - with float clear.

```
.content { float: right; }

.sidebar { float: left; }

.footer { clear: both; }
```



Media Type and Queries

Monday, January 30, 2017 12:42 AM

- One of the most interesting feature of HTML and CSS is that we control which type of style sheet must be supplied to a media type.
- So we can supply different types of style sheets to different devices.

Media Types

Media types allow you to serve different styles to multiple devices based on device type.

- By so defining correct media type we can tell supply corrent style to different device a/c to what they understands.

Media Query Syntax

link element syntax

```
<link rel="stylesheet" type="text/css"
href="styles.css" media="sceen">
```

@import syntax

```
@import url("styles.css") screen;
```

@media rule

```
@media screen {
 //styles
}u
```

Css Reset

Monday, January 30, 2017 1:16 AM

CSS Reset

- A set of styles that nullify a browser's default style sheet, allowing designers to avoid conflicts between their styles and browser default styles.

- Designer can nullify browser's default styles and work everything of their own.

CSS Reset Properties

- Margins
- Padding
- Borders
- Font-size
- Line-height

- Every CSS reset provides following properties.

CSS Reset Placement

- To use CSS reset we just need to place our baseline style to the top of our style sheet.
- We can overwrite these baseline style later on.

NOTE - we can find meyer's CSS reset File at **meyerweb.com**