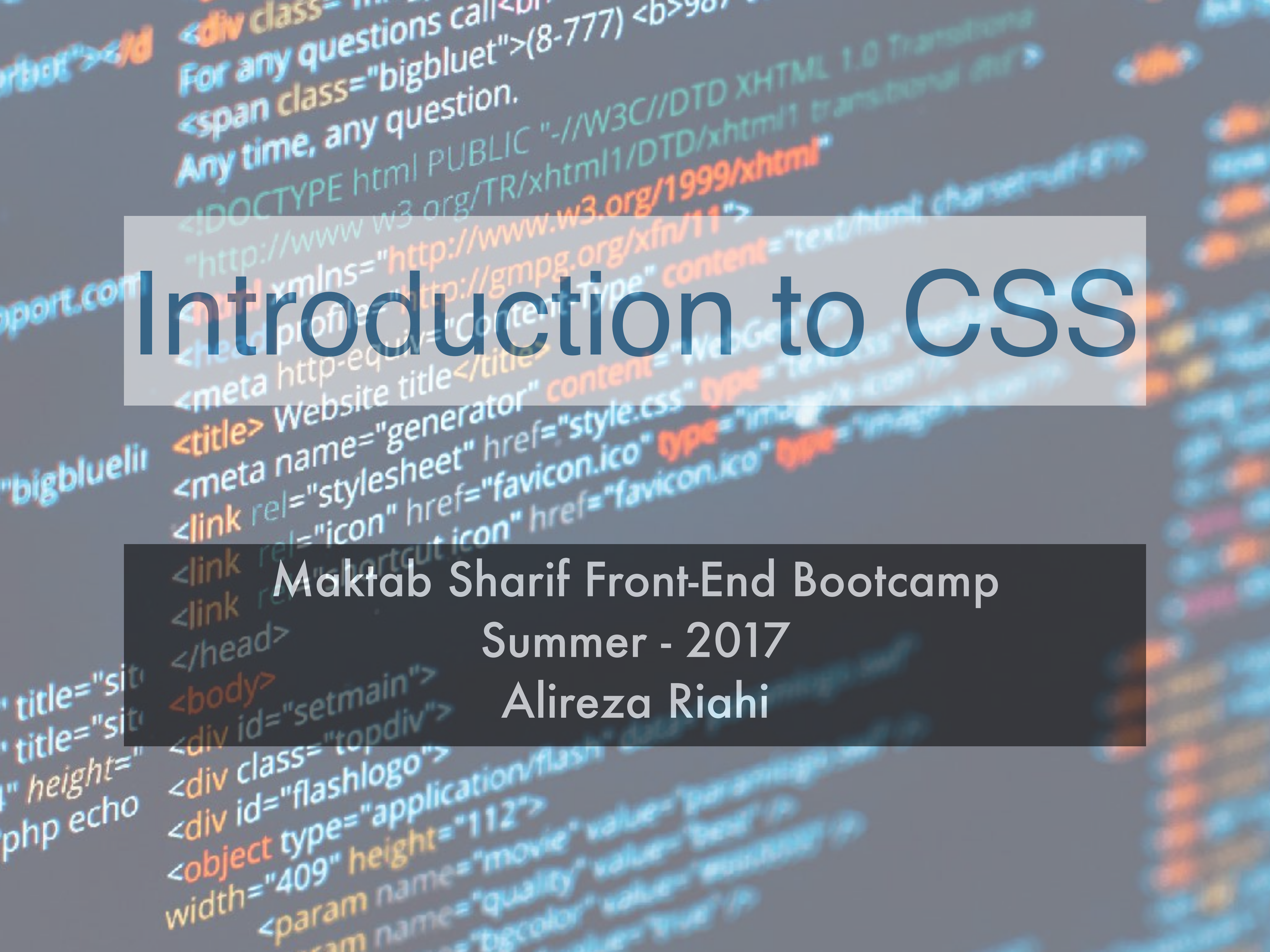
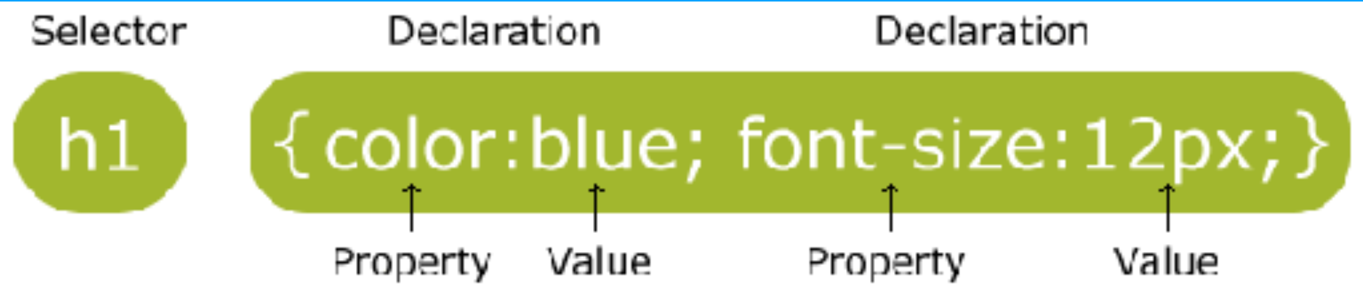


Introduction to CSS

Maktab Sharif Front-End Bootcamp
Summer - 2017
Alireza Riahi

CSS Syntax



CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and more.

The element Selector:

You can select all <p> elements on a page like this

```
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.

```
#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.

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

CSS Syntax

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:

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

HTML elements can also refer to more than one class.

```
<p class="center large">This paragraph refers to two classes.</p>
```

Grouping Selectors

If you have elements with the same style definitions, It will be better to group the selectors, to minimize the code. To group selectors, separate each selector with a comma.

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

CSS Comments

Comments are ignored by browsers. A CSS comment starts with /* and ends with */. Comments can also span multiple lines:

```
p {
   color: red;
   /* This is a single-line comment */
   text-align: center;
}
```

Insert CSS

There are three ways of inserting a style sheet:

External Style Sheet

With an external style sheet, you can change the look of an entire website by changing just one file! Each page must include a reference to the external style sheet file inside the <link> element. The <link> element goes inside the <head> section:

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

Internal Style Sheet

An internal style sheet may be used if one single page has a unique style. Internal styles are defined within the <style> element, inside the <head> section of an HTML page:

```
<head>
<style>
h1 {
   color: maroon;
   margin-left: 40px;
}
</style>
</head>
```

Inline Styles

An inline style may be used to apply a unique style for a single element. To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

```
<h1 style="color:blue;margin-left:30px;">This is a heading</h1>
```

CSS Backgrounds

The background-color property specifies the background color of an element.

```
body {
   background-color: lightblue;
}
```

The background-image property specifies an image to use as the background of an element. By default, the image is repeated so it covers the entire element.

```
body {
   background-image: url("paper.gif");
}
```

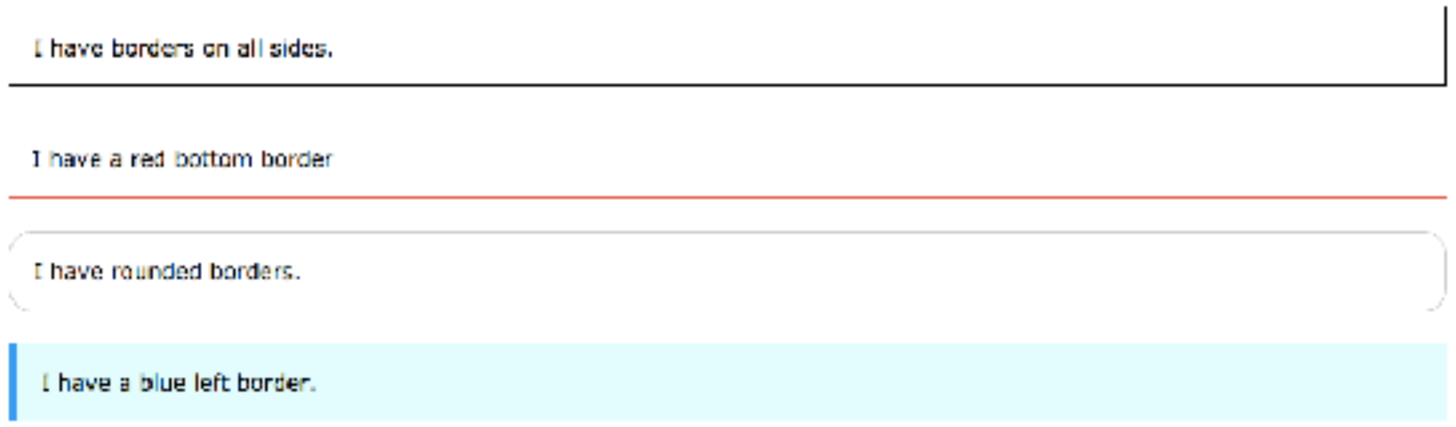
By default, the background-image property repeats an image both horizontally and vertically. If the image above is repeated only horizontally (background-repeat: repeat-x;). The position of the image is specified by the background-position property. the background image is shown in the same place as the text. To specify that the background image should be fixed (will not scroll with the rest of the page), use the background-attachment property.

```
body {
   background-image: url("img_tree.png");
   background-repeat: no-repeat;
   background-position: right top;
   background-attachment: fixed;
}
```

Background - Shorthand property

```
body {
   background: #ffffff url("img_tree.png") no-repeat right top;
}
```

CSS Borders


I have borders on all sides.

I have a red bottom border

I have rounded borders.

I have a blue left border.

The border-style property specifies what kind of border to display. The border-width property specifies the width of the four borders. The border-color property is used to set the color of the four borders. The border-radius property is used to add rounded borders to an element.

```
p {
   border-style: solid;
   border-width: 5px;
   border-color: green;
   border-radius: 5px;
}
```

The border property is a shorthand property for the following individual border properties:

- border-width
- border-style (required)
- border-color

```
p {
   border: 5px solid red;
}
```

Border Style property:

- dotted
- dashed
- solid
- double
- groove
- ridge
- inset
- outset
- none
- hidden

Border color:
border-color: red green blue yellow;

Border width:
border-width: 2px 10px 4px 20px;

Border style:
border-style: dotted solid double dashed;

CSS Margins and Padding

Margins

The CSS margin properties are used to generate space around elements. The margin properties set the size of the white space outside the border.

```
p {
   margin-top: 100px;
   margin-bottom: 100px;
   margin-right: 150px;
   margin-left: 80px;
}
```

The margin property is a shorthand property for the following individual margin properties:

```
p {
   margin: 100px 150px 100px 80px;
}
```

You can set the margin property to auto to horizontally center the element within its container.

Padding

The CSS padding properties are used to generate space around content. The padding clears an area around the content (inside the border) of an element.

The padding property is a shorthand property for the following individual padding properties:

```
p {
   padding: 50px 30px 50px 80px;
}
```

CSS Height and Width

The height and width properties are used to set the height and width of an element.

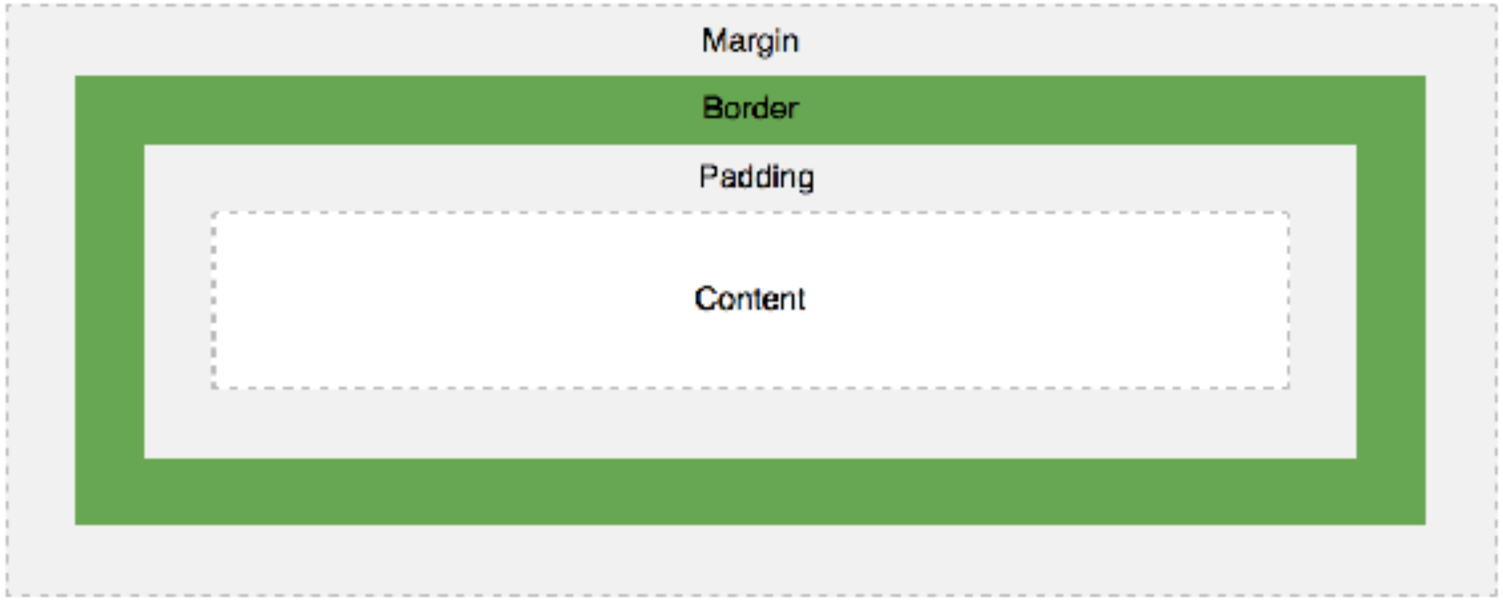
```
div {
   height: 200px;
   width: 50%;
   background-color: powderblue;
}
```

The max-width property is used to set the maximum width of an element.

```
div {
   max-width: 500px;
   height: 100px;
   background-color: powderblue;
}
```

max-height	Sets the maximum height of an element
max-width	Sets the maximum width of an element
min-height	Sets the minimum height of an element
min-width	Sets the minimum width of an element

CSS Box Model



All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout. The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. Explanation of the different parts:

- Content - The content of the box, where text and images appear
- Padding - Clears an area around the content. The padding is transparent
- Border - A border that goes around the padding and content
- Margin - Clears an area outside the border. The margin is transparent

When you set the width and height properties of an element with CSS, you just set the width and height of the content area. To calculate the full size of an element, you must also add padding, borders and margins. Assume we want to style a <div> element to have a total width of 350px:

```
div {
   width: 320px;
   padding: 10px;
   border: 5px solid gray;
   margin: 0;
}
```

320px (width)
+ 20px (left + right padding)
+ 10px (left + right border)
+ 0px (left + right margin)
= 350px

CSS Text

Text Color

The color property is used to set the color of the text.

```
h1 {
   color: green;
}
```

Text Alignment

The text-align property is used to set the horizontal alignment of a text.

```
h1 {
   text-align: center;
}
```

Text Decoration

The text-decoration property is used to set or remove decorations from text.

```
a {
   text-decoration: none;
}
```

Text Indentation

The text-indent property is used to specify the indentation of the first line of a text:

```
p {
   text-indent: 50px;
}
```

Line Height

The line-height property is used to specify the space between lines:

```
p {
   line-height: 0.8;
}
```

Text Direction

The direction property is used to change the text direction of an element:

```
p {
   direction: rtl;
}
```

Text Shadow

The text-shadow property adds shadow to text. The following example specifies the position of the horizontal shadow (3px), the position of the vertical shadow (2px) and the color of the shadow (red):

```
h1 {
   text-shadow: 3px 2px red;
}
```

CSS Fonts

Font Family

The font family of a text is set with the font-family property. If the name of a font family is more than one word, it must be in quotation marks, like: "Times New Roman".

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

Font Style

The font-style property is mostly used to specify italic text.

This property has three values:

- normal - The text is shown normally
- italic - The text is shown in italics
- oblique - The text is "leaning" (oblique is very similar to italic, but less supported)

```
p.italic {
   font-style: italic;
}
```

Font Size

Setting the text size with pixels gives you full control over the text size:

```
h2 {
   font-size: 30px;
}
```

Font Weight

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

CSS Links

Links can be styled with any CSS property (e.g. color, font-family, background, etc.). In addition, links can be styled differently depending on what state they are in.

The four links states are:

- a:link - a normal, unvisited link
- a:visited - a link the user has visited
- a:hover - a link when the user mouses over it
- a:active - a link the moment it is clicked

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

```
a:link, a:visited {
   background-color: #f44336;
   color: white;
   padding: 14px 25px;
   text-align: center;
   text-decoration: none;
   display: inline-block;
}

a:hover, a:active {
   background-color: red;
}
```

CSS Icons

Font Awesome Icons

The simplest way to add an icon to your HTML page, is with an icon library, such as Font Awesome. Add the name of the specified icon class to any inline HTML element (like <i> or <span>). All the icons in the icon libraries below, are scalable vectors that can be customized with CSS (size, color, shadow, etc.)

To use the Font Awesome icons, add the following line inside the <head> section of your HTML page:

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
```

```
<i class="fa fa-cloud"></i>
<i class="fa fa-heart"></i>
<i class="fa fa-car"></i>
<i class="fa fa-file"></i>
<i class="fa fa-bars”  ></i>
```

Bootstrap Icons

To use the Bootstrap glyphicons, add the following line inside the <head> section of your HTML page:

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
```

```
<i class="glyphicon glyphicon-cloud"></i>
<i class="glyphicon glyphicon-remove”  ></i>
<i class="glyphicon glyphicon-user"></i>
<i class="glyphicon glyphicon-envelope"></i>
<i class="glyphicon glyphicon-thumbs-up”  ></i>
```

Google Icons

To use the Google icons, add the following line inside the <head> section of your HTML page:

```
<link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
```

```
<i class="material-icons">cloud</i>
<i class="material-icons">favorite</i>
<i class="material-icons">attachment</i>
<i class="material-icons">computer</i>
<i class="material-icons">traffic</i>
```

CSS Lists

Different List Item Markers

The list-style-type property specifies the type of list item marker.

```
ul.a {
   list-style-type: circle;
}
```

The list-style-image property specifies an image as the list item marker:

```
ul {
   list-style-image: url('sqpurple.gif');
}
```

Remove Default Settings

The list-style-type:none property can also be used to remove the markers/bullets. Note that the list also has default margin and padding. To remove this, add margin:0 and padding:0 to <ul> or <ol>:

```
ul {
   list-style-type: none;
   margin: 0;
   padding: 0;
}
```

CSS Tables

To specify table borders in CSS, use the border property. The border-collapse property sets whether the table borders should be collapsed into a single border. Width and height of a table are defined by the width and height properties. The text-align property sets the horizontal alignment (like left, right, or center) of the content in <th> or <td>. The vertical-align property sets the vertical alignment (like top, bottom, or middle) of the content in <th> or <td>. For zebra-striped tables, use the nth-child() selector and add a background-color to all even (or odd) table rows

```
table {
   border-collapse: collapse;
   width: 100%;
}

table, th, td {
   border: 1px solid black;
}
th {
   height: 50px;
   text-align: center;
   vertical-align: bottom;
}
tr:nth-child(even) {
background-color: #f2f2f2
}
```

CSS Layout

The display property is the most important CSS property for controlling layout. Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is block or inline.

display: none; is commonly used with JavaScript to hide and show elements without deleting and recreating them.

Hiding an element can be done by setting the display property to none. The element will be hidden, and the page will be displayed as if the element is not there.

visibility:hidden; also hides an element. However, the element will still take up the same space as before. The element will be hidden, but still affect the layout:

CSS Position

The position property specifies the type of positioning method used for an element (static, relative, fixed or absolute). Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the position value.

HTML elements are positioned static by default. Static positioned elements are not affected by the top, bottom, left, and right properties.

position: fixed;

An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.

Notice the fixed element in the lower-right corner of the page. Here is the CSS that is used:

```
div. {
   position: fixed;
   bottom: 0;
   right: 0;
}
```

CSS Position

position: relative;

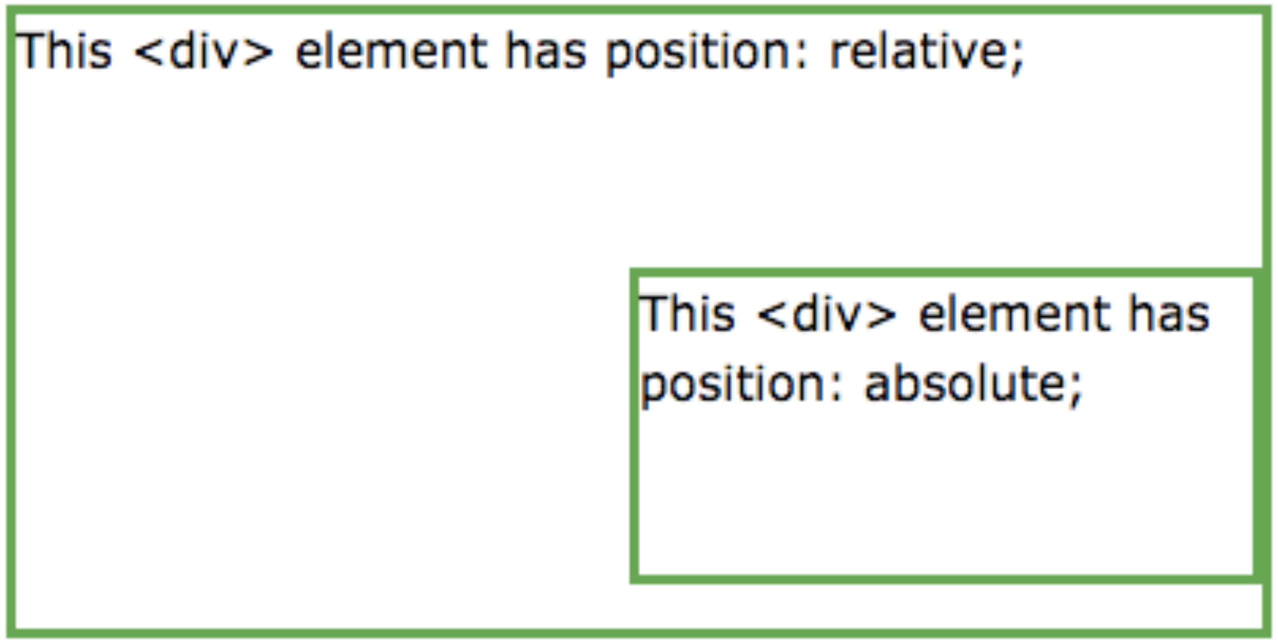
An element with position: relative; is positioned relative to its normal position. Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.

position: absolute;

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.

```
div.relative {
   position: relative;
   width: 400px;
   height: 200px;
   border: 3px solid #73AD21;
}

div.absolute {
   position: absolute;
   top: 80px;
   right: 0;
   width: 200px;
   height: 100px;
   border: 3px solid #73AD21;
}
```



This <div> element has position: relative;

This <div> element has position: absolute;

CSS Overflow

The CSS overflow property specifies whether to clip content or to add scrollbars when the content of an element is too big to fit in a specified area. The overflow property only works for block elements with a specified height.

The overflow property has the following values:

- visible - Default. The overflow is not clipped. It renders outside the element's box
- hidden - The overflow is clipped, and the rest of the content will be invisible
- scroll - The overflow is clipped, but a scrollbar is added to see the rest of the content
- auto - If overflow is clipped, a scrollbar should be added to see the rest of the content

The overflow-x and overflow-y properties specifies whether to change the overflow of content just horizontally or vertically (or both):

overflow-x specifies what to do with the left/right edges of the content.
overflow-y specifies what to do with the top/bottom edges of the content.

CSS Float and Clear

The float Property

The float property specifies whether or not an element should 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

CSS inline-block

The old way - using float (notice that we also need to specify a clear property for the element after the floating boxes):

```
.floating-box {
   float: left;
   width: 150px;
   height: 75px;
   margin: 10px;
   border: 3px solid #73AD21;
}

.after-box {
   clear: left;
}
```

The same effect can be achieved by using the inline-block value of the display property (notice that no clear property is needed):

```
.floating-box {
   display: inline-block;
   width: 150px;
   height: 75px;
   margin: 10px;
   border: 3px solid #73AD21;
}
```

CSS Horizontal & Vertical Align

To horizontally center a block element (like <div>), use margin: auto; Setting the width of the element will prevent it from stretching out to the edges of its container.

To just center the text inside an element, use text-align: center;

To center an image, use margin: auto; and make it into a block element.

There are many ways to center an element vertically in CSS. A simple solution is to use top and bottom padding.

See this link:

https://www.w3schools.com/css/css_align.asp

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 <div> elements:

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

CSS Combinators

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 <div> element:

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

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 <div> elements:

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

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 <div> elements:

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

CSS Pseudo-classes

A pseudo-class is used to define a special state of an element. For example, it can be used to:

- Style an element when a user mouses over it
- Style visited and unvisited links differently
- Style an element when it gets focus

The syntax of pseudo-classes:

```
selector:pseudo-class {
   property:value;
}
```

Links can be displayed in different ways =>

The :first-child Pseudo-class

```
p:first-child {
   color: blue;
}
```

https://www.w3schools.com/cssref/css_selectors.asp

```
/* unvisited link */
a:link {
    color: #FF0000;
}

/* visited link */
a:visited {
    color: #00FF00;
}

/* mouse over link */
a:hover {
    color: #FF00FF;
}

/* selected link */
a:active {
    color: #0000FF;
}
```

CSS Opacity / Transparency

The opacity property specifies the opacity/transparency of an element. The opacity property can take a value from 0.0 - 1.0. The lower value, the more transparent.

Note: IE8 and earlier use filter:alpha(opacity=x). The x can take a value from 0 - 100. A lower value makes the element more transparent.

```
img {
    opacity: 0.5;
    filter: alpha(opacity=50); /* For IE8 and earlier */
}
```

Finished

for deep information you can visit

www.w3schools.com

Alireza Riahi
ali.r.riahi@gmail.com

Maktab Sharif - Summer 2017