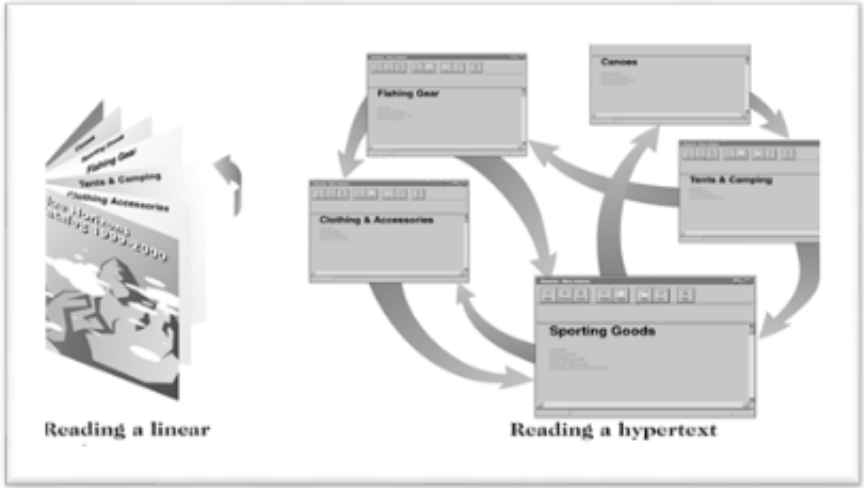
Cognizant Digital Experience

HTML & CSS Overview

Cognizant Digital Experience

What is HTML

- **Hypertext Markup Language** (HTML) is the language for specifying the content of Web pages
- Hypertext refers to the fact that Web pages are more than just text
- They can contain multimedia, provide links for jumping within & beyond
- Markup refers to the fact that it works by augmenting text with special symbols (tags) that identify structure and content type




```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Source Code

My First Heading

My first paragraph.

Browser View

HTML Page Structure

Element	Description
<html> </html>	Surrounds the entire page
<head> </head>	Contains header information (metadata, CSS styles, Java Script code)
<title> </title>	Holds the page title normally displayed in the title bar and used in search results
<body> </body>	Contains the main body text. All parts of the page normally visible are in the body

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Source Code

Browser View

HTML Tags & Attributes

Tags	A tag is a piece of text inside angle bracket (<>). Tags typically have a beginning and an end, with some text inside them.
Attributes	Tags are sometimes enhanced by name – value pairs that modify the tag. For example, img tag (used to embed image in a page) can have src attribute to describe where the image file can be found.

```
<!DOCTYPE html>
<html>
<body>

<h2>HTML Images</h2>
<p>HTML images are defined with the img tag:</p>

<img
src="https://upload.wikimedia.org/wikipedia/commons/thumb/6/61/HTML5_logo_and_w
ordmark.svg/120px-HTML5_logo_and_wordmark.svg.png" alt="HTML Logo" width="174"
height="142">

</body>
</html>
```

Source Code

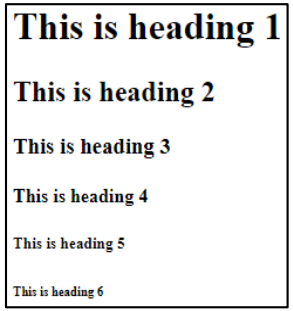
Browser View

HTML Headings

	Heading 1	Reserved to Strongest emphasis
	Heading 2	Secondary level heading. Headings go dow n to level 6.
	Paragraph	Surrounds a paragraph

```
<!DOCTYPE html>
<html>
<body>

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>

</body>
</html>
```

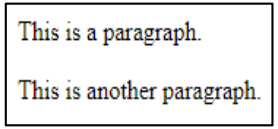
Source Code



Browser View

```
<!DOCTYPE html>
<html>
<body>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

</body>
</html>
```

Source Code



Browser View

HTML Links

<a>	Defines hyperlink	The href attribute indicates the link's destination. The link text is the part that will be visible to the reader.

```
<!DOCTYPE html>
<html>
<body>

<h1>HTML Links</h1>

<p><a href="https://www.w3schools.com/">Visit W3Schools.com!</a></p>

</body>
</html>
```

Source Code

Browser View

HTML Lists

	Unordered List	Bullets with list
	Ordered List	Numbered List

```
<!DOCTYPE html>
<html>
<body>

<h2>An unordered HTML list</h2>

<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>

</body>
</html>
```

Source Code

An unordered HTML list

- Coffee
- Tea
- Milk

Browser View

```
<!DOCTYPE html>
<html>
<body>

<h2>An ordered HTML list</h2>

<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

</body>
</html>
```

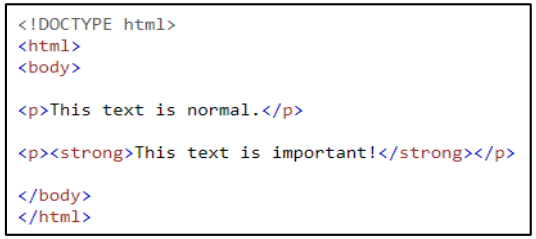
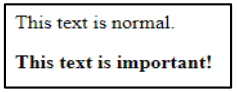
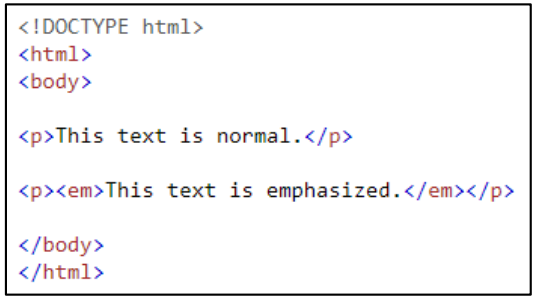
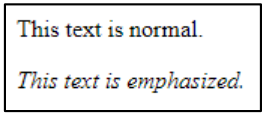
Source Code

An ordered HTML list

1. Coffee
2. Tea
3. Milk

Browser View

HTML Formatting

	Bold	Defines text with importance	 Source Code	 Browser View
	Italic	Defines emphasized text	 Source Code	 Browser View

HTML – Adding Images

<img>	Attach an image	Associated w ith SRC (defines the location of the image) and ALT (describes the content of the image)

```
<!DOCTYPE html>
<html>
<body>

<h2>HTML Image</h2>
<img src="pic_trulli.jpg" alt="Trulli" width="500" height="333">

</body>
</html>
```

Source Code

Browser View

HTML – Adding Audio, Video

<audio controls>	Attach an audio	Provides functionalities like Play, Pause, Seeking, Volume	Source Code / Browser View
<video>	Attach a video	Provides video controls, like play, pause, and volume	Source Code / Browser View

```
<!DOCTYPE html>
<html>
<body>

<h1>The audio controls attribute</h1>

<p>Click on the play button to play a sound:</p>

<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
</audio>

</body>
</html>
```

Source Code

Browser View

```
<!DOCTYPE html>
<html>
<body>

<video width="400" controls>
  <source src="mov_bbb.mp4" type="video/mp4">
  <source src="mov_bbb.ogg" type="video/ogg">
  Your browser does not support HTML video.
</video>

<p>
Video courtesy of
<a href="https://www.bigbuckbunny.org/" target="_blank">Big Buck Bunny</a>.
</p>

</body>
</html>
```

Source Code

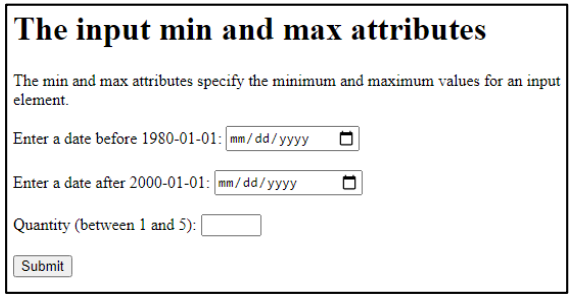
Browser View

HTML Forms

<form>	Collect user input	Container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc.

```
<!DOCTYPE html>
<html>
<body>

<h1>The input min and max attributes</h1>

<p>The min and max attributes specify the minimum and maximum values for an input element.</p>

<form action="/action_page.php">
  <label for="datemax">Enter a date before 1980-01-01:</label>
  <input type="date" id="datemax" name="datemax" max="1979-12-31"><br><br>

  <label for="datemin">Enter a date after 2000-01-01:</label>
  <input type="date" id="datemin" name="datemin" min="2000-01-02"><br><br>

  <label for="quantity">Quantity (between 1 and 5):</label>
  <input type="number" id="quantity" name="quantity" min="1" max="5"><br><br>

  <input type="submit" value="Submit">
</form>

</body>
</html>
```

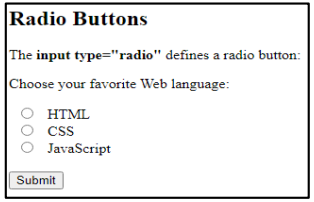
Source Code



Browser View

```
<!DOCTYPE html>
<html>
<body>

<h2>Radio Buttons</h2>

<p>The <strong>input type="radio"</strong> defines a radio button:</p>

<p>Choose your favorite Web language:</p>
<form action="/action_page.php">
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language" value="JavaScript">
  <label for="javascript">JavaScript</label><br><br>
  <input type="submit" value="Submit">
</form>

</body>
</html>
```

Source Code



Browser View

HTML Table

<table>	Displays tabular data into rows and columns	Consists a series of table rows (tr). Each table row has a number of table data (td) elements.

```
<!DOCTYPE html>
<html>
<body>

<h2>Basic HTML Table</h2>

<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
  </tr>
</table>

</body>
</html>
```

Source Code

Basic HTML Table

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94
John	Doe	80

Browser View

What is CSS

CSS stands for **Cascading Style Sheets**

CSS describes how HTML elements are to be displayed on screen, paper, or in other media

CSS makes website more flexible because CSS is reusable and change of stylesheet can change design of many pages

With CSS, it is easier to maintain as it leads to cleaner HTML code since it separates styles from HTML tags and page content. Also, consistent look across entire website can be achieved by changing styles in one place.

If different styles are specified for HTML elements, the styles will cascade into new styles with the following priority:

Priority 1: Inline styles

Priority 2: External and internal style sheets

Priority 3: Browser default

If different styles are defined on the same priority level, the last one has the highest priority.

Without CSS

```
<font size="14px">
My First Header
</font>
<font size="12px" color="red" face="Verdana">
My information 1 goes here.
</font>
<font size="14px">
My Second Header
</font>
<font size="12px" color="red" face="Verdana">
Different information goes here.
</font>
```

With CSS

```
<p class="header">My First Header</p>
<p class="info">My Information 1 goes here</p>
<p class="header">My Second Header</p>
<p class="info">Different  Information goes here</p>
```

Content

```
.header { font-size:14px;}
.info {          font-family: verdana;
                 font-color: blue;
                 font-size: 12px; }
```

Style

CSS Syntax

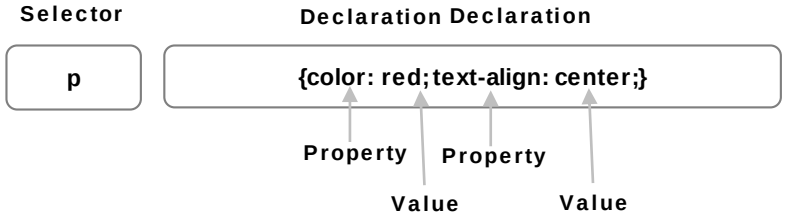
```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  color: red;
  text-align: center;
}
</style>
</head>
<body>

<p>Hello World!</p>
<p>These paragraphs are styled with CSS.</p>

</body>
</html>
```

Source Code

Hello World!

These paragraphs are styled with CSS.

Browser View



The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

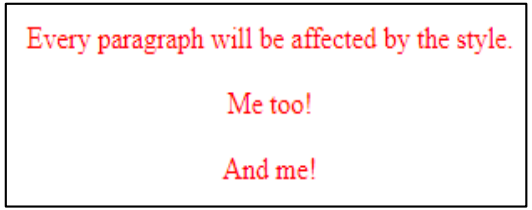
Each declaration includes a CSS property name and a value, separated by a colon.

Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

CSS Selectors – Type Selector

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<p>Every paragraph will be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>

</body>
</html>
```

Source Code



Browser View

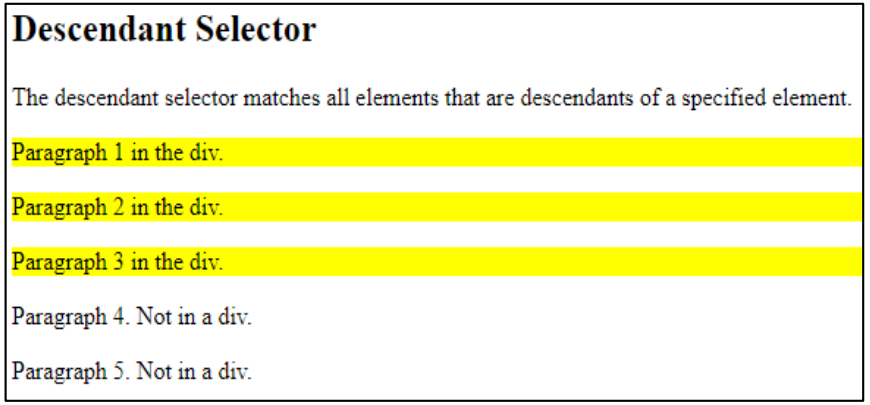
Type Selector

The type selector selects HTML elements based on the element name.

CSS Selectors - Descendant Selector

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

<h2>Descendant Selector</h2>
<p>The descendant selector matches all elements that
are descendants of a specified element.</p>
<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
  <section><p>Paragraph 3 in the div.</p></section>
</div>
<p>Paragraph 4. Not in a div.</p>
<p>Paragraph 5. Not in a div.</p>
</body>
</html>
```

Source Code



Browser View

Descendant Selector

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

CSS Selectors - Class Selector

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<h1 class="center">Red and center-aligned heading</h1>
<p class="center">Red and center-aligned paragraph.</p>

</body>
</html>
```

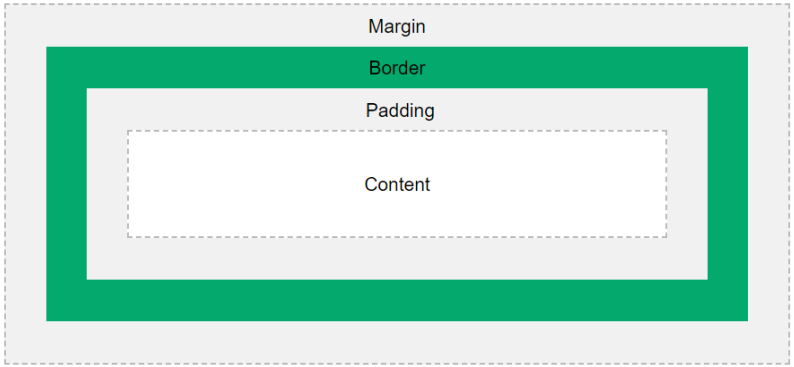
Source Code

Browser View

Class Selector

The class selector selects HTML elements with a specific class attribute.

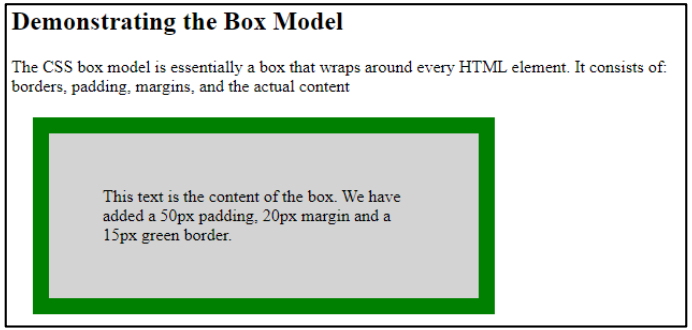
CSS Box Model



The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The box model allows us to add a border around elements, and to define space between elements.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: lightgrey;
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
</style>
</head>
<body>
<h2>Demonstrating the Box Model</h2>
<p>The CSS box model is essentially a box that wraps around every HTML
element. It consists of: borders, padding, margins, and the actual content
</p>
<div>This text is the content of the box. We have added a 50px padding, 20px
margin and a 15px green border. </div>
</body>
</html>
```

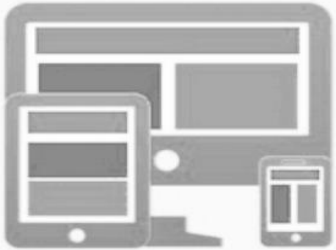
Source Code



Browser View

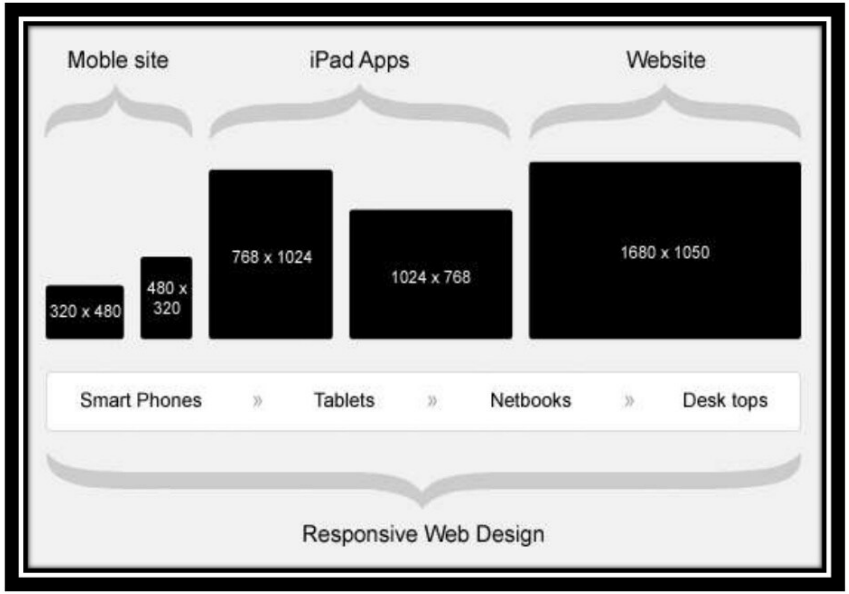
Responsive Web Design

Responsive Web Design is about using HTML and CSS to automatically resize, hide, shrink, or enlarge, a website, to make it look good on all devices (desktops, tablets, and phones)

Fluid Grids	Flexible Images	Media Queries
		
A percentage based design that adapts to the screen size accordingly	Images sized in relative units to prevent them from displaying outside their containing element	A way to apply CSS rules to the page based on the size of the displaying browser

- Smooth Navigation
- Easy Reading
- Reduced scrolling & zooming
- Excellent User Experience

Setting Viewport



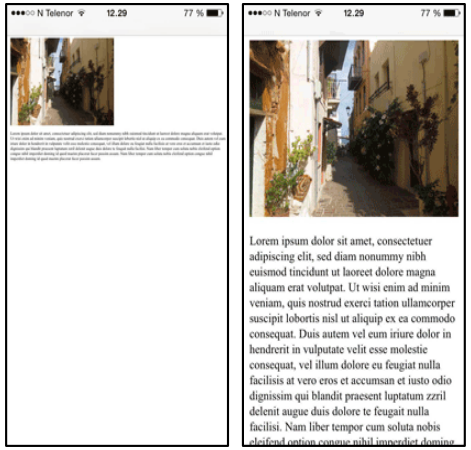

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>

<h2>Setting the Viewport</h2>
<p>This example does not really do anything, other than showing you how to add
the viewport meta element.</p>

</body>
</html>
```

Source Code

Setting the Viewport

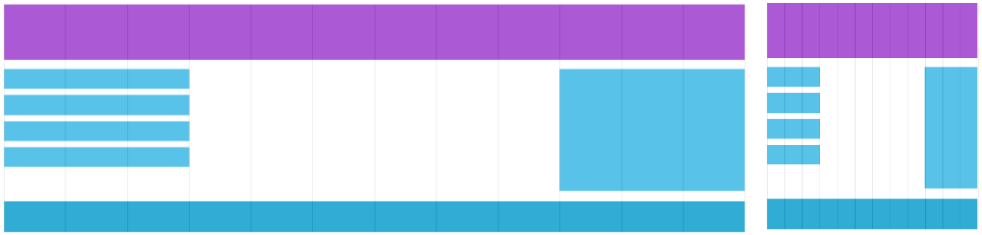
This example does not really do anything, other than showing you how to add the viewport meta element.

Browser View



Without Viewport With Viewport

Grid View



A responsive grid-view has 12 columns, and has a total width of 100%, and will shrink and expand as you resize the browser window.

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content=
"width=device-width, initial-scale=1.0">
<style>
* {
  box-sizing: border-box;
}

.row::after {
  content: "";
  clear: both;
  display: table;
}

[class*="col-"] {
  float: left;
  padding: 15px;
}
```

- Set box-sizing property to border-box
- All columns should be floating to the left, and have a padding of 15px

```
.col-1 {width: 8.33%;}
.col-2 {width: 16.66%;}
.col-3 {width: 25%;}
.col-4 {width: 33.33%;}
.col-5 {width: 41.66%;}
.col-6 {width: 50%;}
.col-7 {width: 58.33%;}
.col-8 {width: 66.66%;}
.col-9 {width: 75%;}
.col-10 {width: 83.33%;}
.col-11 {width: 91.66%;}
.col-12 {width: 100%;}
```

- Make one class for each of the 12 columns, class="col-" and a number defining how many columns the section should span

```
<div class="row">
  <div class="col-3 menu">
    <ul>
      <li>The Flight</li>
      <li>The City</li>
      <li>The Island</li>
      <li>The Food</li>
    </ul>
  </div>

  <div class="col-9">
    <h1>The City</h1>
    <p>Chania is the capital of the Chania
    region on the island of Crete. </p>
    <p>Resize the browser window to see
    how the content respond to the resizing.
    </p>
  </div>
</div>
```

- Each row should be wrapped in a . The number of columns inside a row should always add up to 12

Browser View

Image Adjustments

```html
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>

<h2>Responsive Image</h2>
<p>When the CSS width property is set in a percentage value, the image will scale up and down when resizing the browser window. Resize the browser window to see the effect.</p>

<img src="img_girl.jpg" style="width:100%;">

</body>
</html>
```

Set image width as 100%

```html
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>

<h2>Show Different Images Depending on Browser Width</h2>
<p>Resize the browser width and the image will change at 600px and 1500px.</p>

<picture>
  <source srcset="img_smallflower.jpg" media="(max-width: 600px)">
  <source srcset="img_flowers.jpg" media="(max-width: 1500px)">
  <source srcset="flowers.jpg">
  <img src="img_flowers.jpg" alt="Flowers" style="width:auto;">
</picture>

</body>
</html>
```

Use different images based on browser width

```html
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
/* For width smaller than 400px: */
body {
  background-repeat: no-repeat;
  background-image: url('img_smallflower.jpg');
}

/* For width 400px and larger: */
@media only screen and (min-width: 400px) {
  body {
    background-image: url('img_flowers.jpg');
  }
}
</style>
</head>
<body>

<p style="margin-top:360px;">Resize the browser width and the background image will change at 400px.</p>

</body>
</html>
```

Use different images based on browser width

(Media Queries based solution)

Media Queries

Media Query uses the @media rule to include a block of CSS properties only if a certain condition is true

```
/* If the screen size is 601px or more, set the font-size of <div> to 80px */
@media only screen and (min-width: 601px) {
  div.example {
    font-size: 80px;
  }
}

/* If the screen size is 600px or less, set the font-size of <div> to 30px */
@media only screen and (max-width: 600px) {
  div.example {
    font-size: 30px;
  }
}
```

Change Font Size With Media Queries

```
/* Extra small devices (phones, 600px and down) */
@media only screen and (max-width: 600px) {...}

/* Small devices (portrait tablets and large phones, 600px and up) */
@media only screen and (min-width: 600px) {...}

/* Medium devices (landscape tablets, 768px and up) */
@media only screen and (min-width: 768px) {...}

/* Large devices (laptops/desktops, 992px and up) */
@media only screen and (min-width: 992px) {...}

/* Extra large devices (large laptops and desktops, 1200px and up) */
@media only screen and (min-width: 1200px) {...}
```

Handling typical Breakpoints

```
@media only screen and (orientation: landscape) {
  body {
    background-color: lightblue;
  }
}
```

Make adjustments based on orientation: Portrait / Landscape

```
/* If the screen size is 600px wide or less, hide the element */
@media only screen and (max-width: 600px) {
  div.example {
    display: none;
  }
}
```

Hide Elements With Media Queries

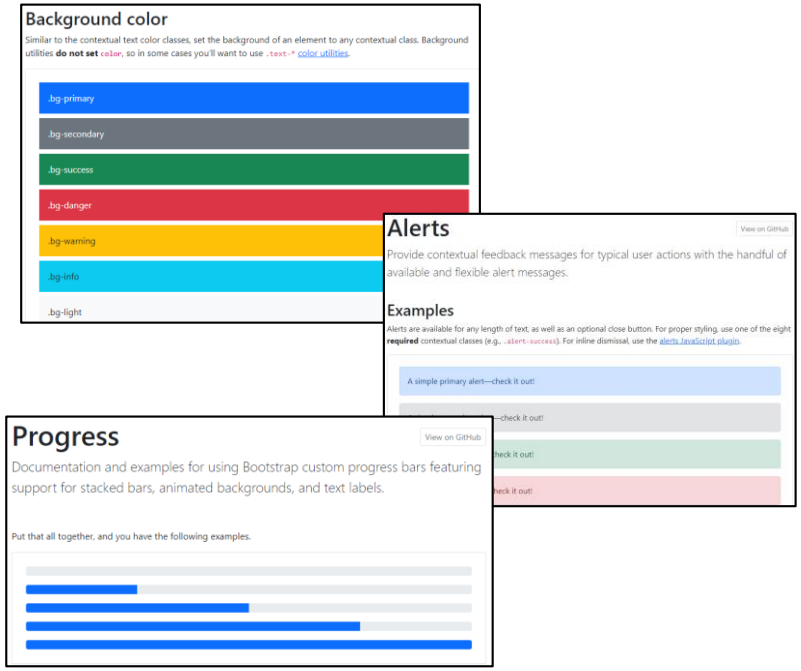
Bootstrap

What is Bootstrap

- Bootstrap is a free front-end framework for faster and easier web development
- Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins
- Bootstrap also gives you the ability to easily create responsive designs

Why Bootstrap

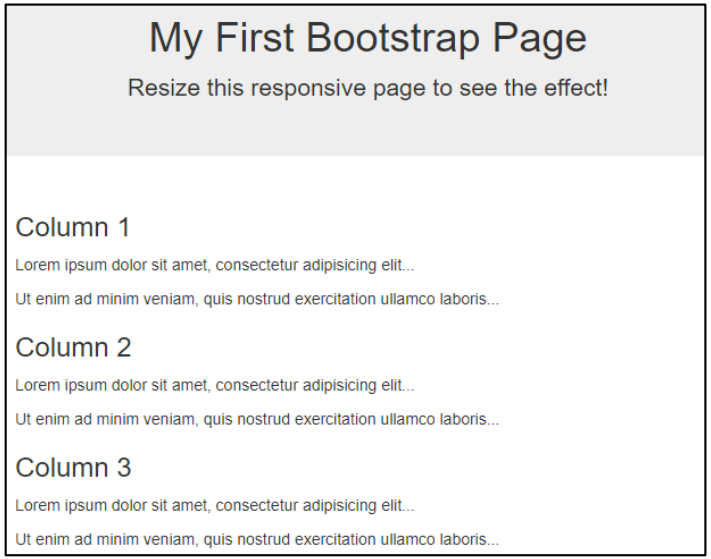
- Easy to use: Anybody with just basic knowledge of HTML and CSS can start using Bootstrap
- Responsive features: Bootstrap's responsive CSS adjusts to phones, tablets, and desktops
- Mobile-first approach: In Bootstrap 3, mobile-first styles are part of the core framework
- Browser compatibility: Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Edge, Safari, and Opera)



Refer https://getbootstrap.com/docs

Bootstrap Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css"
>
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js">
</script>
</head>
<body>

<div class="jumbotron text-center">
  <h1>My First Bootstrap Page</h1>
  <p>Resize this responsive page to see the effect!</p>
</div>

<div class="container">
  <div class="row">
    <div class="col-sm-4">
      <h3>Column 1</h3>
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit...</p>
      <p>Ut enim ad minim veniam, quis nostrud exercitation ullamco
laboris...</p>
    </div>
    <div class="col-sm-4">
      <h3>Column 2</h3>
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit...</p>
      <p>Ut enim ad minim veniam, quis nostrud exercitation ullamco
laboris...</p>
    </div>
    <div class="col-sm-4">
      <h3>Column 3</h3>
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit...</p>
      <p>Ut enim ad minim veniam, quis nostrud exercitation ullamco
laboris...</p>
    </div>
  </div>
</div>
</body>
</html>
```

Sass

What is Sass

- Sass stands for Syntactically Awesome Stylesheet
- Sass is a CSS pre-processor
- Sass reduces repetition of CSS and therefore saves time

Why Sass

- Stylesheets are getting larger, more complex, and harder to maintain. This is where a CSS pre-processor can help.
- Sass lets you use features that do not exist in CSS, like variables, nested rules, mixins, imports, inheritance, built-in functions, and other stuff.

```scss
/* Define standard variables and values for website */
$bgcolor: lightblue;
$textcolor: darkblue;
$fontsize: 18px;

/* Use the variables */
body {
  background-color: $bgcolor;
  color: $textcolor;
  font-size: $fontsize;
}
```

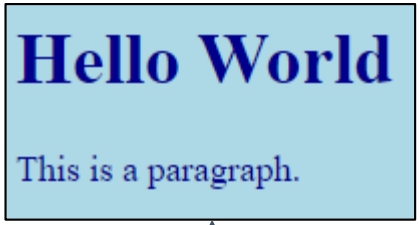
.scss

```css
body {
  background-color: lightblue;
  color: darkblue;
  font-size: 18px;
}
```

.css

```html
<!DOCTYPE html>
<html>
<link rel="stylesheet" href="mystyle.css">
<body>

<h1>Hello World</h1>

<p>This is a paragraph.</p>

</body>
</html>
```

.html



Browser View

Continued Learning

Course Title	Course URL	Course Duration	Learning Goals
Web Design for Beginners: Real World Coding in HTML & CSS	https://cognizant.udemy.com/course/web-design-for-beginners-real-world-coding-in-html-css/	10h 54m	• Create any website layout you can imagine • Support any device size with Responsive (mobile-friendly) Design • Add tasteful animations and effects with CSS3 • Use common vocabulary from the design industry

Section Name	# of Lectures	Duration (in minutes)	Mandatory / Optional
Welcome	2	4	Mandatory
HTML Essentials	8	43	Mandatory
Adding Media to a Web Page	4	30	Mandatory
Text Basics	4	20	Mandatory
Semantics & Organization	5	25	Mandatory
Forms	3	25	Mandatory
Tables	2	9	Mandatory
CSS Essentials	7	59	Mandatory
Intermediate CSS	2	23	Mandatory
Less Exciting Yet Still Necessary CSS Tasks	2	30	Mandatory
CSS Typography	2	22	Mandatory

Section Name	# of Lectures	Duration (in minutes)	Mandatory / Optional
Developer Timeout!	1	6	Mandatory
CSS Backgrounds	4	39	Mandatory
Responsive Web Design	4	43	Mandatory
What is Flexbox in CSS?	2	28	Mandatory
CSS3 Special Effects	5	58	Mandatory
Leveraging JavaScript without Writing JavaScript	5	61	Mandatory
What is Bootstrap?	2	34	Mandatory
Sass	3	54	Mandatory
Pushing a Website Live Up Onto The Web	1	15	Mandatory
Cross Browser Compatibility	1	13	Mandatory
Beyond This Course	2	13	Mandatory

Thank You