Responsive Web Design(RWD)

- Having a responsive website is very important. One must be able to access it from all devices making it available on all platforms. Thus, you must give the developers enough time to work on the website and not rush into creating a website that won't work across multiple devices.

Responsive Web Design(RWD)

- It makes your web page look good on all devices.
- It uses only HTML and CSS.
- It is not a program or a JavaScript.
- Web pages can be viewed using many different devices: **desktops, tablets, and phones.**
  - Your web page should look good, and be easy to use, regardless of the device.
  - Web pages should not leave out information to fit smaller devices, but rather adapt its content to fit any device:
- It is called **responsive web design** when you use CSS and HTML to
  - resize, hide, shrink, enlarge, or move the content to make it look good on any screen.

RWD- The Viewport

- The viewport is the user's visible area of a web page.
- The viewport varies with the device, and will be smaller on a mobile phone than on a computer screen.
- The internet using tablets and mobile phones, fixed size web pages were too large to fit the viewport.
- **To fix this, browsers on those devices scaled down the entire web page to fit the screen.**

Setting The Viewport

- Include the following <meta> viewport element in all your web pages:

<meta name="viewport" content="width=device-width, initial-scale=1.0">

- This gives the browser instructions on how to control the page's dimensions and scaling.
- The width=device-width part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).
- The initial-scale=1.0 part sets the initial zoom level when the page is first loaded by the browser.

Size Content to The Viewport

- Users are used to scroll websites vertically on both desktop and mobile devices - but not horizontally!
- **Some additional rules to follow:**
  - **1. Do NOT use large fixed width elements -** For example, if an image is displayed at a width wider than the viewport it can cause the viewport to scroll horizontally. Remember to adjust this content to fit within the width of the viewport.
  - **2. Do NOT let the content rely on a particular viewport width to render well** - Since screen dimensions and width in CSS pixels vary widely between devices, content should not rely on a particular viewport width to render well.
  - **3. Use CSS media queries to apply different styling for small and large screens** -
    - Setting large absolute CSS widths for page elements will cause the element to be too wide for the viewport on a smaller device.
    - Instead, consider using relative width values, such as width: 100%. Also, be careful of using large absolute positioning values.
    - It may cause the element to fall outside the viewport on small devices.

Responsive Web Design - Grid-View

- Many web pages are based on a grid-view, which means that the page is divided into **columns**.
- Using a grid-view is very helpful when designing web pages. It makes it easier to place elements on the page.
- **A responsive grid-view** often has **12 columns**, and has a **total width of 100%,** and will shrink and expand as you resize the browser window.

Building a Responsive Grid-View

- First ensure that all HTML elements have the box-sizing property set to border-box.
  - This makes sure that the padding and border are included in the total width and height of the elements.
- Add the following code in your CSS:

```
* {
  box-sizing: border-box;
}
```

```
<style>
* {  box-sizing: border-box; }
.header {   border: 1px solid red;  padding: 15px; }
.menu {  width: 25%;  float: left;  padding: 15px;  border: 1px solid red;}
.main {
 width: 75%;  float: left;  padding: 15px;  border: 1px solid red;}
</style>
<div class="header">  <h1>Chania</h1></div>
<div class="menu">
 <ul>   <li>The Flight</li>   <li>The City</li>   <li>The Island</li>   <li>The Food</li>  </ul></div>
<div class="main">
 <h1>The City</h1>  <p>Chania is the capital of the Chania region on the island of Crete. The city can be divided in two parts, the old town and the modern city.</p>
</div>
```

Grid System

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
# of columns	12				
Gutter width	30px (15px on each side of a column)				
Nestable	Yes				
Column ordering	Yes				

RWD – Media Queries

- Media query is a CSS technique introduced in CSS3.
- It uses the @media rule to include a block of CSS properties only if a certain condition is **true.**

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
body {
  background-color: lightgreen;
}

@media only screen and (max-width: 600px) {
  body {
    background-color: lightblue;
  }
}
</style>
```

Always Design for Mobile First

- Mobile First means designing for mobile before designing for desktop or any other device
  - This will make the page display faster on smaller devices
- This means that we must make some changes in CSS.
- Instead of changing styles when the width gets *smaller* than 768px, we should change the design when the width gets *larger* than 768px.
- This will make our design Mobile First

Responsive breakpoints

- // Extra Small devices (landscape phones, 600px and down)
  @media (max-width: 600px) { ... }

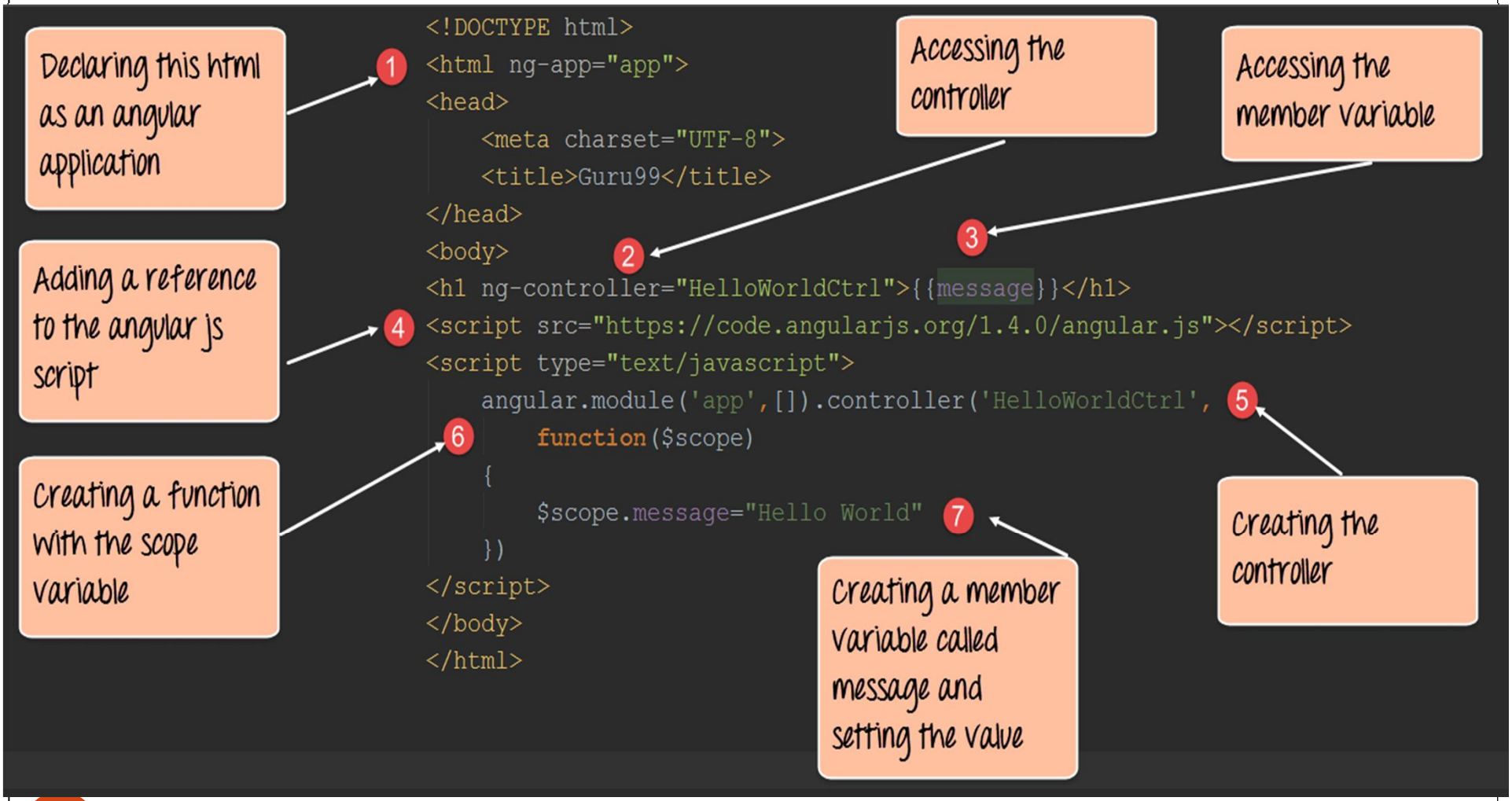
- // Small devices (landscape phones, 576px and up)
  @media (min-width: 576px) { ... }

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

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

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

Summary


Declaring this html as an angular application
Adding a reference to the angular js script
Creating a function with the scope variable
Accessing the controller
Accessing the member variable
Creating the controller
Creating a member variable called message and setting the value
<!DOCTYPE html>
<html ng-app="app">
<head>
<meta charset="UTF-8">
<title>Guru99</title>
</head>
<body>
<h1 ng-controller="HelloWorldCtrl">{{message}}</h1>
<script src="https://code.angularjs.org/1.4.0/angular.js"></script>
<script type="text/javascript">
angular.module('app',[]).controller('HelloWorldCtrl',
function($scope)
{
$scope.message="Hello World"
})
</script>
</body>
</html>
1
2
3
4
5
6
7