SEC201.2 Web-Based Programming

Responsive Web Design & Bootstrap

Outline

- Responsive Web Design (RDW)
- CSS Responsive
  - RDW Viewport
  - RDW Grid View
  - RDW Media Queries
  - RDW Images
  - RDW Videos
  - RDW Frameworks
- Bootstrap
  - What is Bootstrap?
  - How to Install
  - Create First Web Page
  - Bootstrap Containers
  - Bootstrap Grid System Basics
  - Sample Web Site with Bootstrap
- How to check if a website is Mobile Responsive?
- Summary

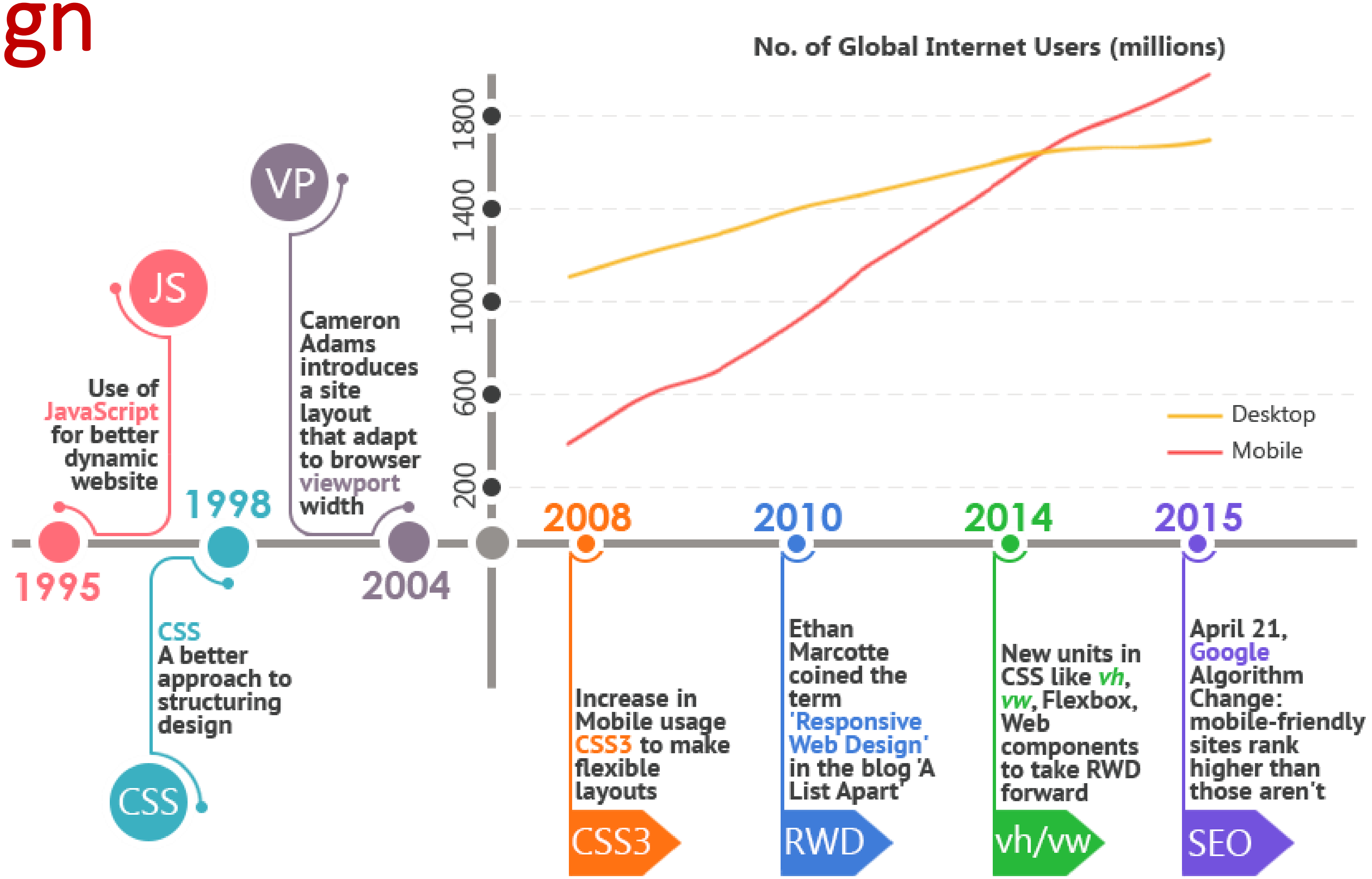
Responsive Web Design

- ***Responsive web design*** is an approach that is widely implemented in the web designing process that enables the web pages to render well across various mobile devices (***desktops***, ***smartphones***, ***tablets***, ***laptops***) having different window or screen sizes and resolutions

- Web developers, especially the front-end developers, focus on developing mobile-friendly websites as the users tend to interact with the website more through smartphones

Responsive Web Design

History of
Responsive Web Design



1995

Use of JavaScript for better dynamic website

1998

CSS A better approach to structuring design

2004

Cameron Adams introduces a site layout that adapt to browser viewport width

2008

Increase in Mobile usage CSS3 to make flexible layouts

2010

Ethan Marcotte coined the term 'Responsive Web Design' in the blog 'A List Apart'

2014

New units in CSS like *vh*, *vw*, Flexbox, Web components to take RWD forward

2015

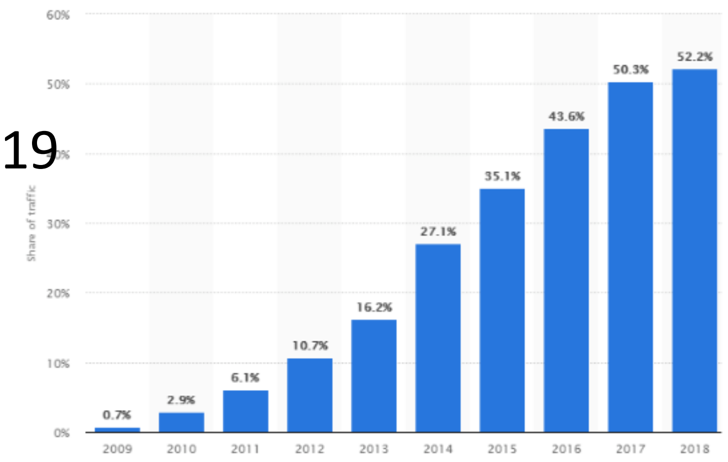
April 21, Google Algorithm Change: mobile-friendly sites rank higher than those aren't

https://marutitech.com/responsive-web-design-key-element-of-web-strategy/

Why is it important to have a Responsive Web Design for your website?

- Given the growth in smartphone users, there has been a remarkable increase in website traffic
- This drastic change drives us to the conclusion that people’s expectations regarding how a website should appear on a mobile device have changed too
- Few stats that denote the importance of having mobile responsiveness
  - Nearly 60% of total internet access happens on mobiles in 2019
  - Mobile phones drive 50% of total e-commerce revenue
  - Total number of adults making use of smartphones is 77%
  - 94% of online visitors judge a website based on the responsiveness of a specific website

Percentage of all global web pages served to mobile

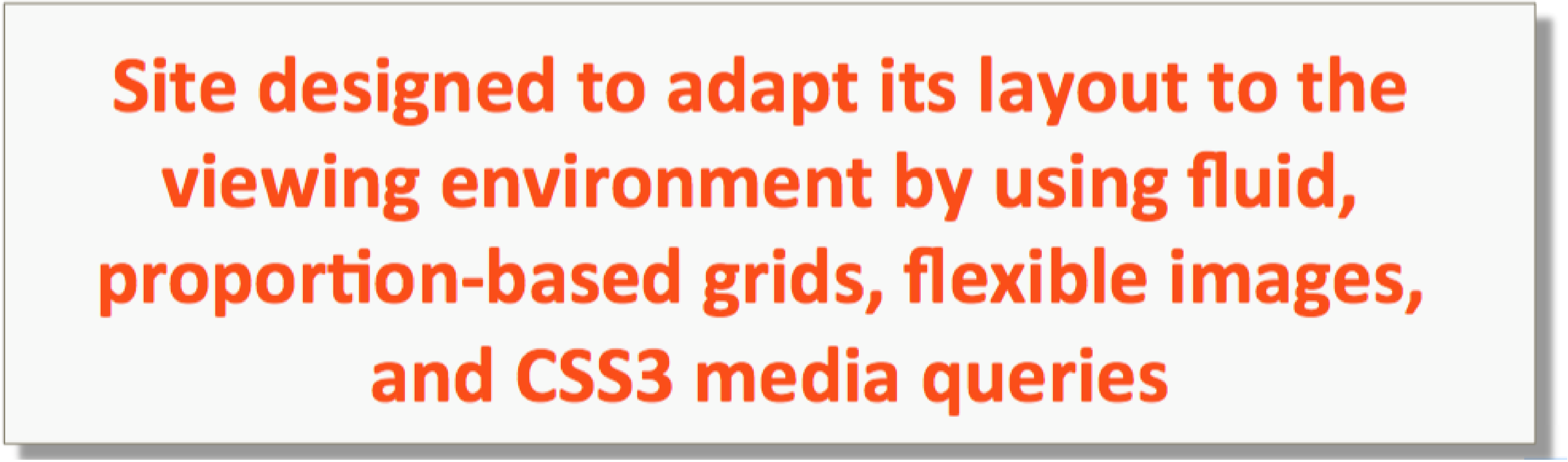


Why is it important to have a Responsive Web Design for your website?

- The above stats, shows **how crucial it is to have a responsive web design for a website** to cater to the rapidly growing online users through smartphones
- A web developer builds a user-friendly website to reach the maximum audience
- **Google** recommends all websites to be **mobile friendly** and assigns an extra weight for **responsiveness**, at the time of indexing a page, thus responsive webpages have superior search positions
- As a result, organizations need to invest resources for **cross browser compatibility testing** and **testing across a variety of real devices**

What device do we need to support?

What is a Responsive Web Site?



Site designed to adapt its layout to the viewing environment by using fluid, proportion-based grids, flexible images, and CSS3 media queries

Understanding Responsive Web Design with an Example

Let’s consider water as an example

- If water is poured into a jar, it adopts the shape of a jar
- If water is poured into a bottle, it takes the form of a bottle

Similarly, think of the content on a website as water, where the content adjusts and renders itself as per the screen-sizes across various devices just like water adopting the shape of various vessels it gets poured in

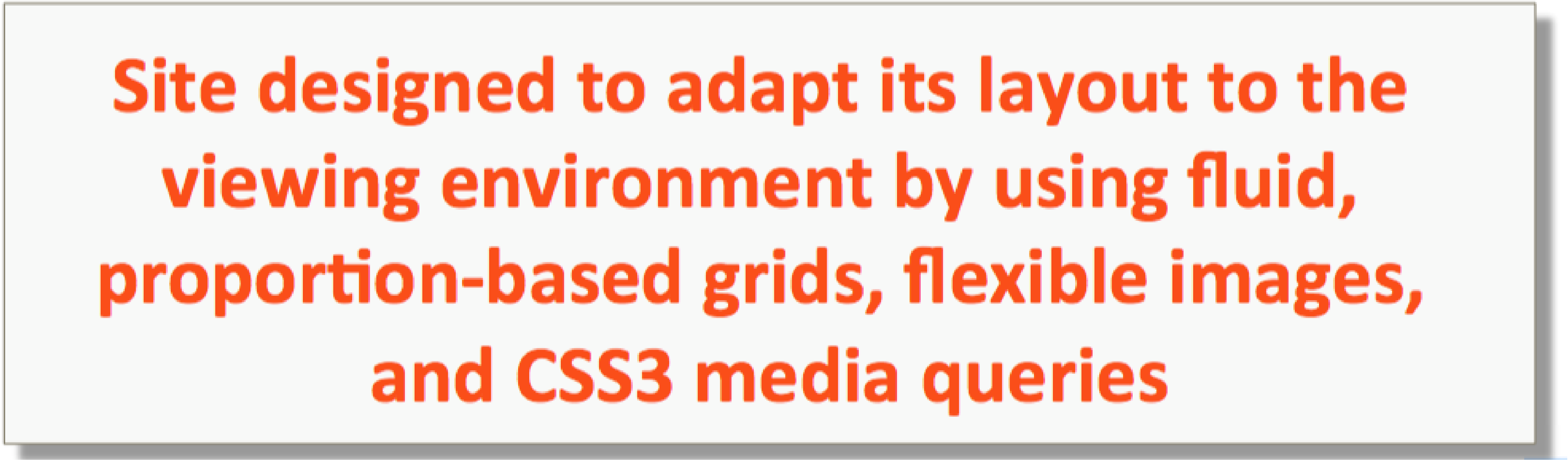
CONTENT IS LIKE WATER

> “You put water into a cup it becomes the cup.
> You put water into a bottle it becomes the bottle.
> You put it in a teapot, it becomes the teapot.”

Josh Clark *(originally Bruce Lee)* - Seven deadly mobile myths

Illustration by Stéphanie Walter

What is a Responsive Web Site?



Site designed to adapt its layout to the viewing environment by using fluid, proportion-based grids, flexible images, and CSS3 media queries

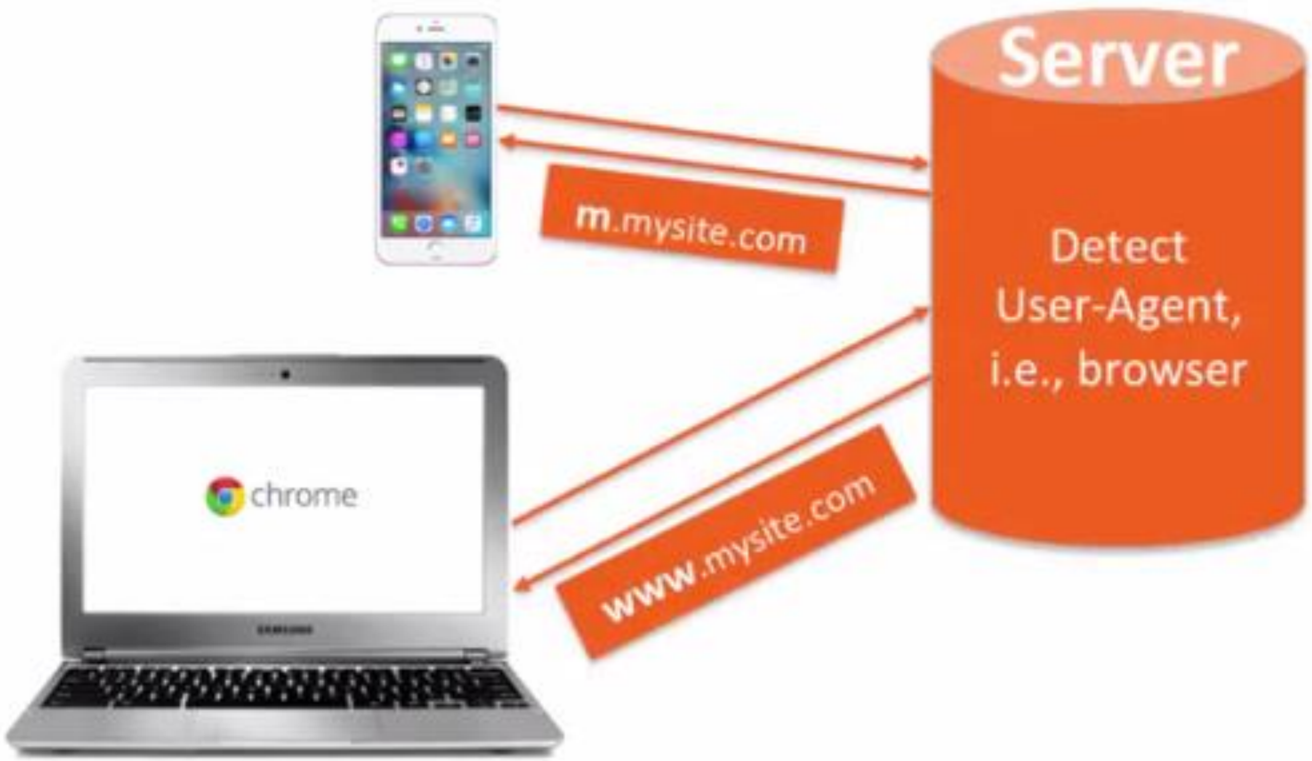
- Site’s layout adapts to the size of the device
- Content verbosity or its visual delivery may change

Alternative to Responsive Design

What you could do is you could have a service high technology that detects your user agent, in other words, the type of browser that you are using, and figure out whether you're on a mobile device or on a desktop device.

Then based on the information either serve up a regular desktop version of the website or serve up to the client the mobile version of that website.

This is basically what the websites that wanted to have a mobile version as well as desktop version are used to do before the responsive design.



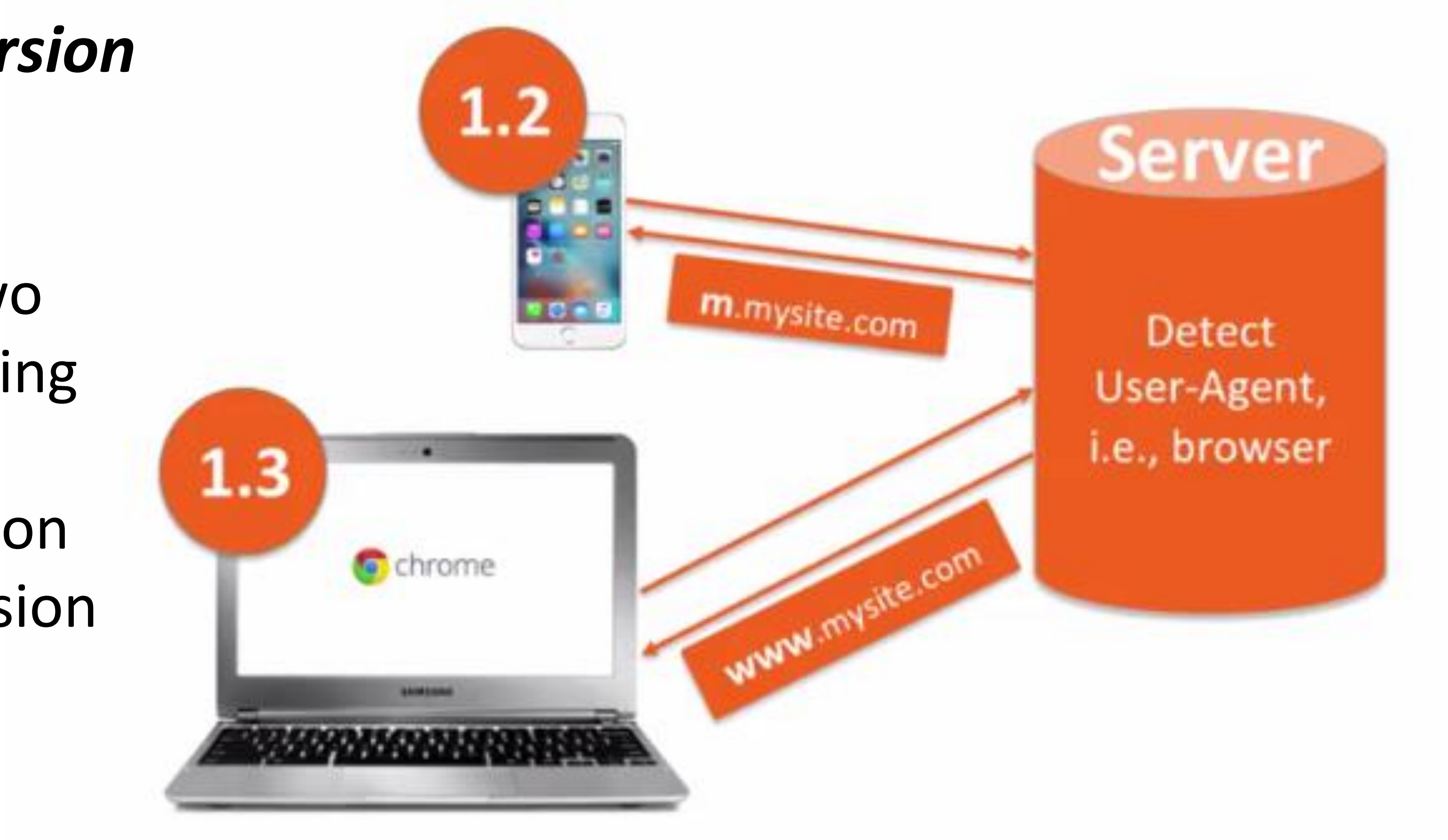
But there are a couple of issues/problems with this type of approach!!!

Alternative to Responsive Design: Problem1

High risk of feature diversion between apps

Because you have really two different applications running your website:

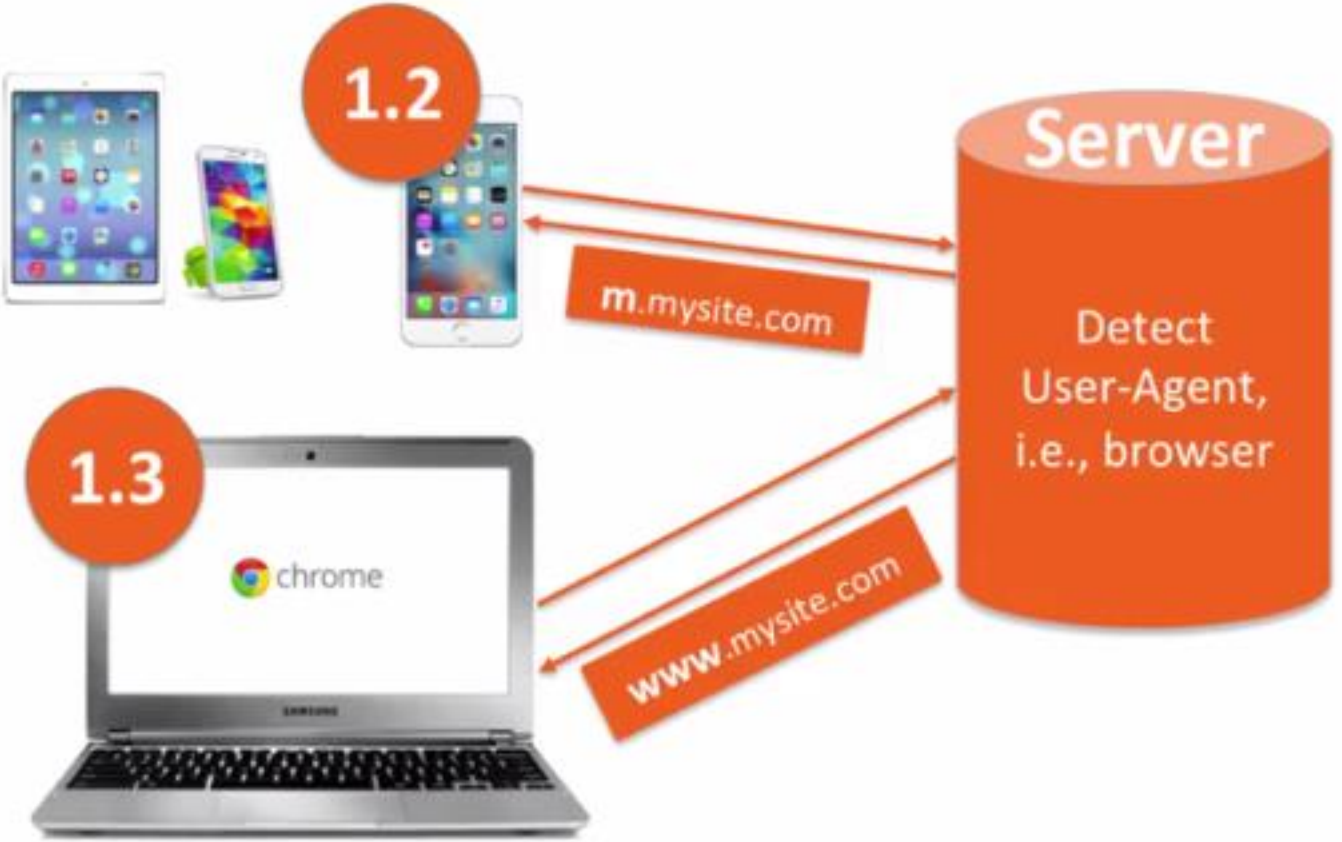
- One is a mobile version
- One is a desktop version



Alternative to Responsive Design: Problem2

Mobile devices are just too varied in size among themselves

- It's hard to make a mobile site that satisfies every client
- What ends up happening is that you end up sacrificing user experience because you are trying to kind of make one mobile website that fits them all and that, basically, never works



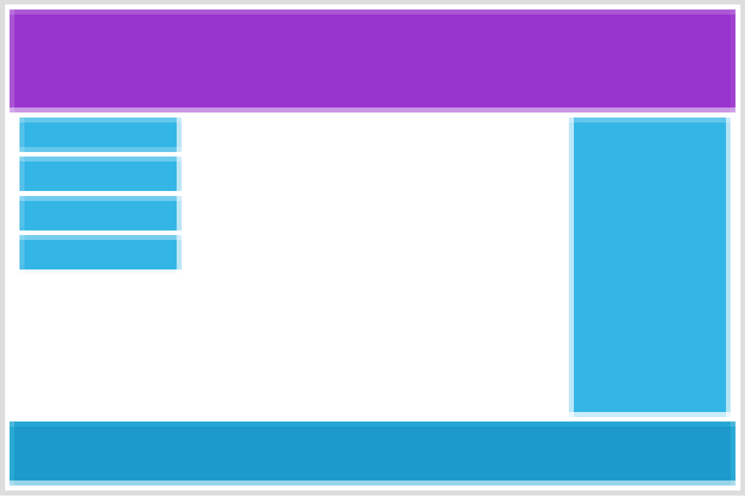
Nowadays, because of these two issues, people really stay away from this approach

Outline

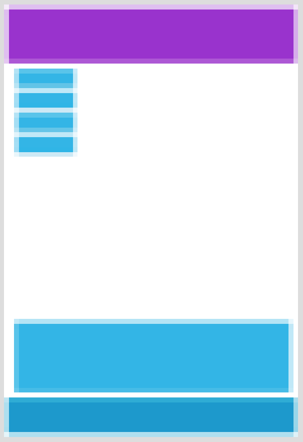
- Responsive Web Design (RDW)
- **CSS Responsive**
  - RDW Viewport
  - RDW Grid View
  - RDW Media Queries
  - RDW Images
  - RDW Videos
  - RDW Frameworks
- Bootstrap
  - What is Bootstrap?
  - How to Install
  - Create First Web Page
  - Bootstrap Containers
  - Bootstrap Grid System Basics
  - Sample Web Site with Bootstrap
- How to check if a website is Mobile Responsive?
- Summary

CSS Responsive

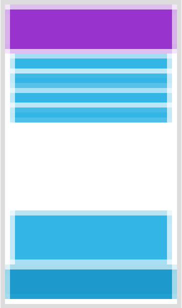
- Responsive web design makes your web page look good on all devices
- Responsive web design uses only HTML and CSS
- Responsive web design is not a program or a JavaScript
- Web pages should not leave out information to fit smaller devices, but rather adapt its content to fit any device:



Desktop



Tablet



Phone

Responsive Web Design - 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
- Before tablets and mobile phones, web pages were designed only for computer screens, and it was common for web pages to have a static design and a fixed size
- Then, when we started surfing 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
- This was not perfect!! But a quick fix

Outline

- Responsive Web Design (RDW)
- **CSS Responsive**
  - **RDW Viewport**
  - RDW Grid View
  - RDW Media Queries
  - RDW Images
  - RDW Videos
  - RDW Frameworks
- Bootstrap
  - What is Bootstrap?
  - How to Install
  - Create First Web Page
  - Bootstrap Containers
  - Bootstrap Grid System Basics
  - Sample Web Site with Bootstrap
- How to check if a website is Mobile Responsive?
- Summary

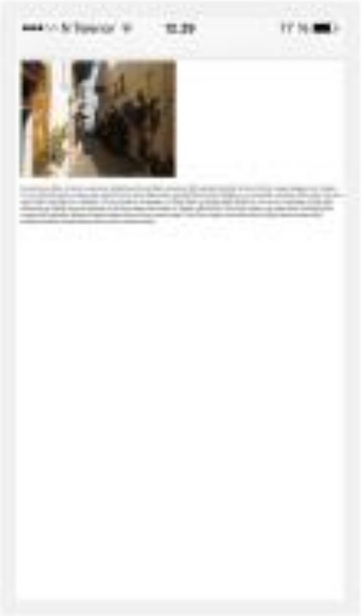
Setting the Viewport

- HTML5 introduced a method to let web designers take control over the viewport, through the **<meta>** tag
- You should include the following **<meta>** viewport element in ***all your web pages***:

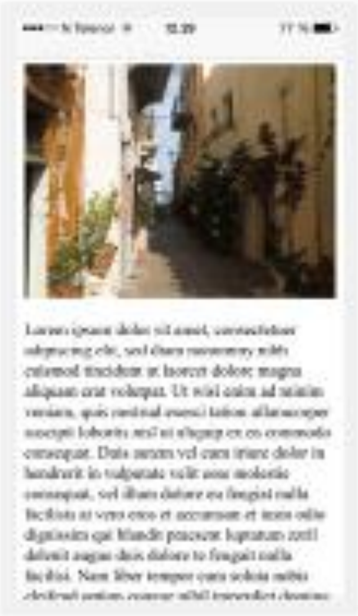
```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

  - A **<meta>** viewport element 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

Example – Viewport Meta Tag



Without the viewport meta tag



Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming

With the viewport meta tag

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

Size Content to The Viewport

- Users are used to scroll websites vertically on both desktop and mobile devices - but not horizontally!

- So, if the user is forced to scroll horizontally, or zoom out, to see the whole web page it results in a poor user experience

- Some additional rules to follow:
    1. *Do NOT use large fixed width elements*
    2. *Do NOT let the content rely on a particular viewport width to render well*
    3. *Use CSS media queries to apply different styling for small and large screens*

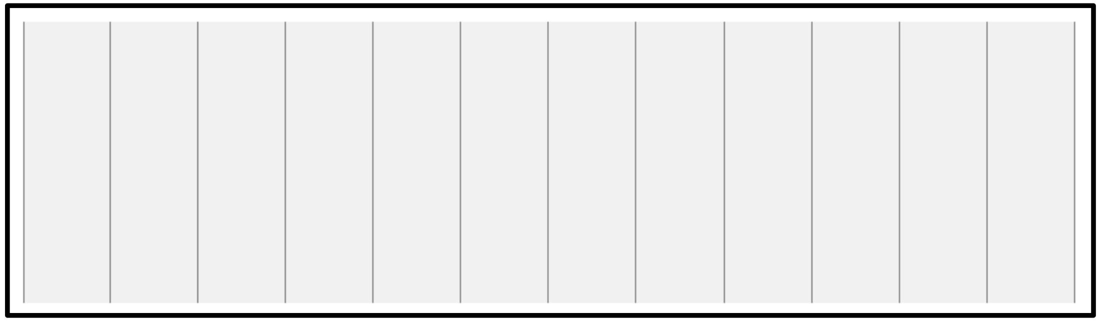
https://www.w3schools.com/css/css_rwd_viewport.asp

Outline

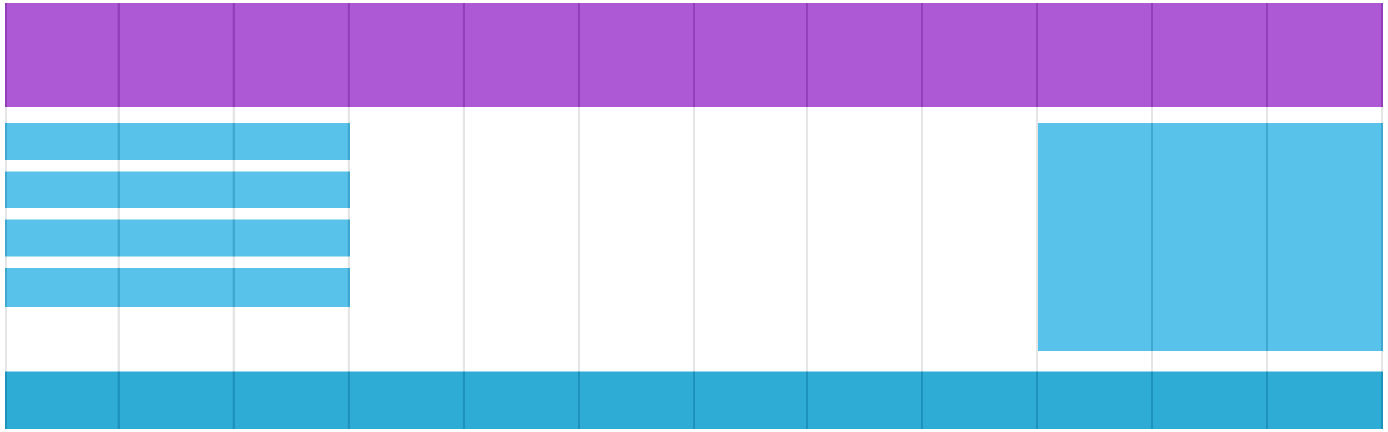
- Responsive Web Design (RDW)
- CSS **Responsive**
  - RDW Viewport
  - **RDW Grid View**
  - RDW Media Queries
  - RDW Images
  - RDW Videos
  - RDW Frameworks
- Bootstrap
  - What is Bootstrap?
  - How to Install
  - Create First Web Page
  - Bootstrap Containers
  - Bootstrap Grid System Basics
  - Sample Web Site with Bootstrap
- How to check if a website is Mobile Responsive?
- Summary

Responsive Web Design - Grid-View

- Many web pages are based on a **grid-view**, which means that **the page is divided into columns**



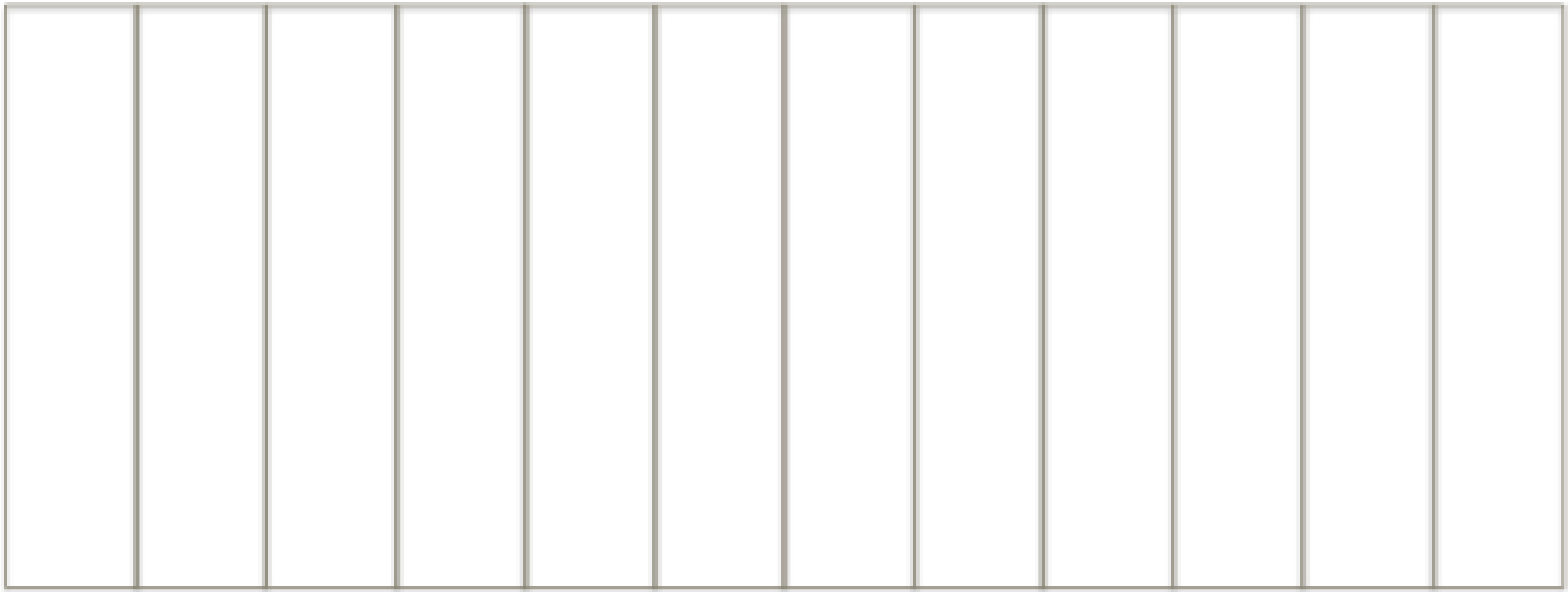
- Using a grid-view is very helpful when designing web pages
  - It makes it easier to place elements on the page



https://www.w3schools.com/css/tryresponsive_grid.htm

12-Column Grid Responsive Layout

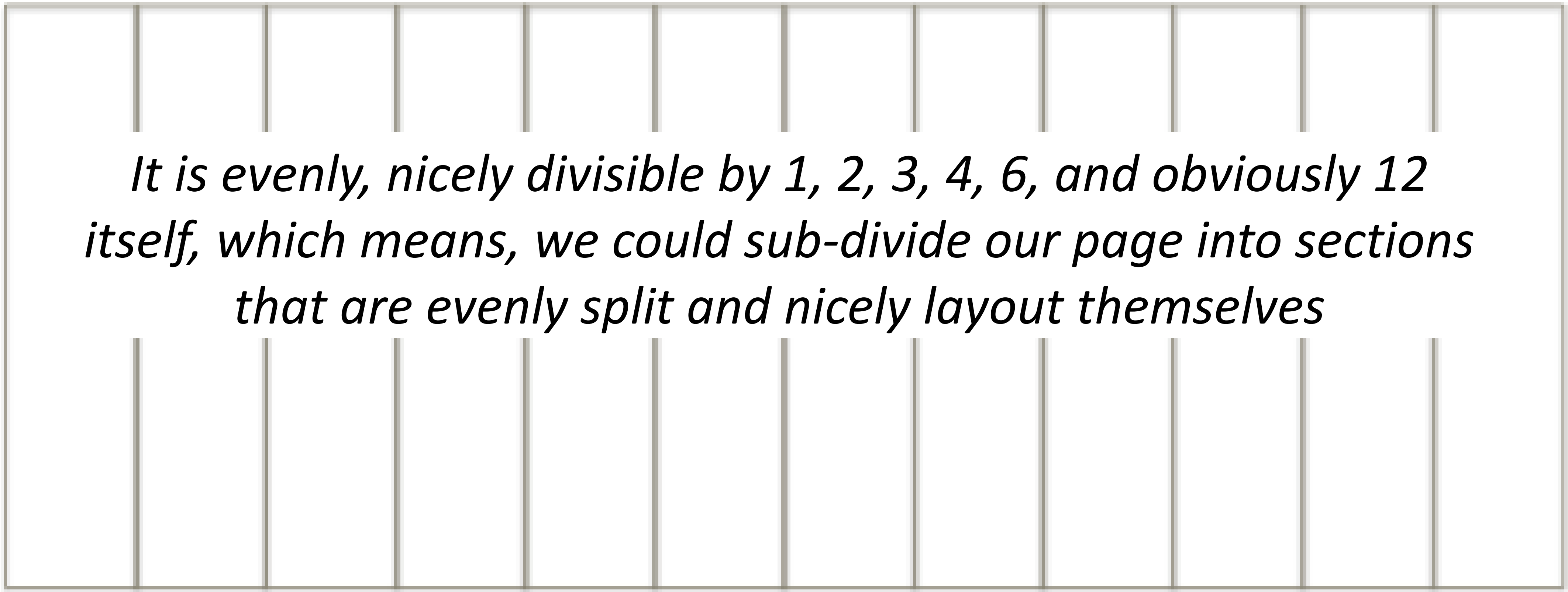
A responsive grid-view often has **12 columns**, and has a **total width of 100%**, and will shrink and expand as you resize the browser window



12-Column Grid Responsive Layout

The reason why **12** is used, because of the **factors of 12**

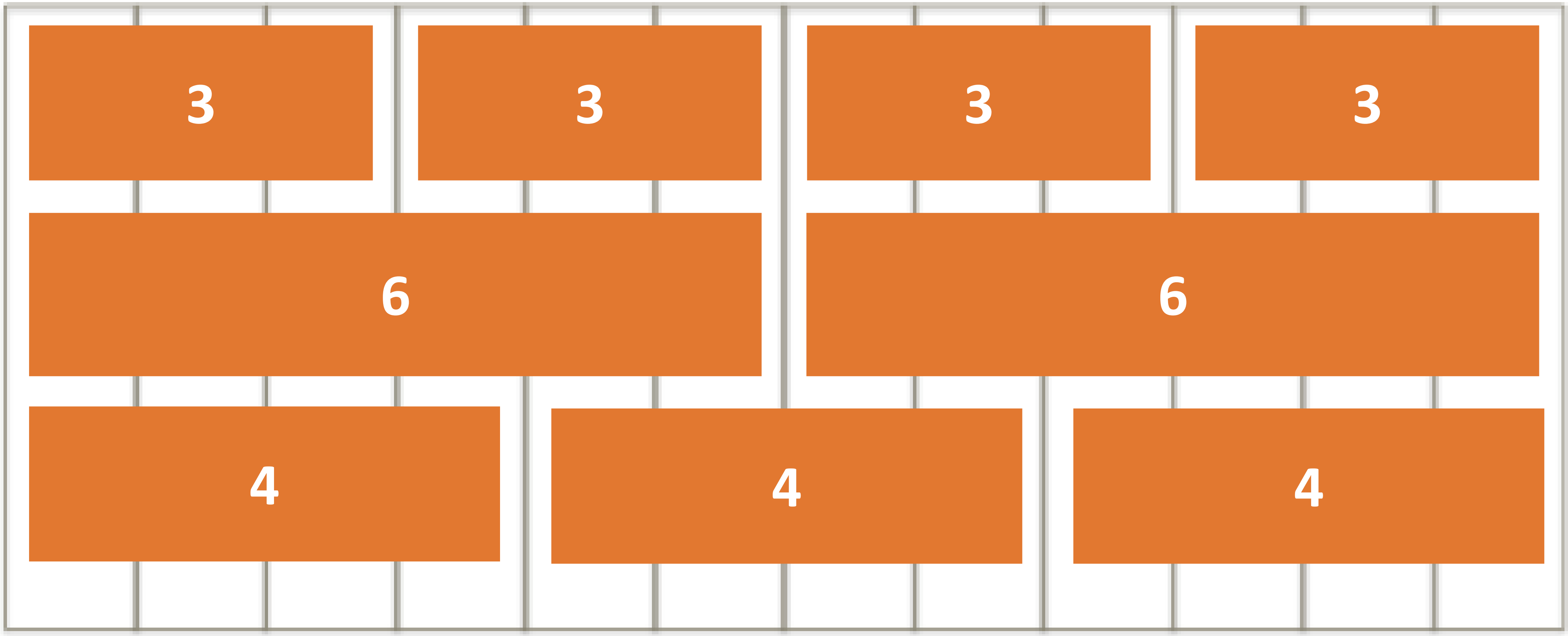
Factors of 12: 1, 2, 3, 4, 6, 12



It is evenly, nicely divisible by 1, 2, 3, 4, 6, and obviously 12 itself, which means, we could sub-divide our page into sections that are evenly split and nicely layout themselves

12-Column Grid Responsive Layout

Factors of 12: **1, 2, 3, 4, 6, 12**

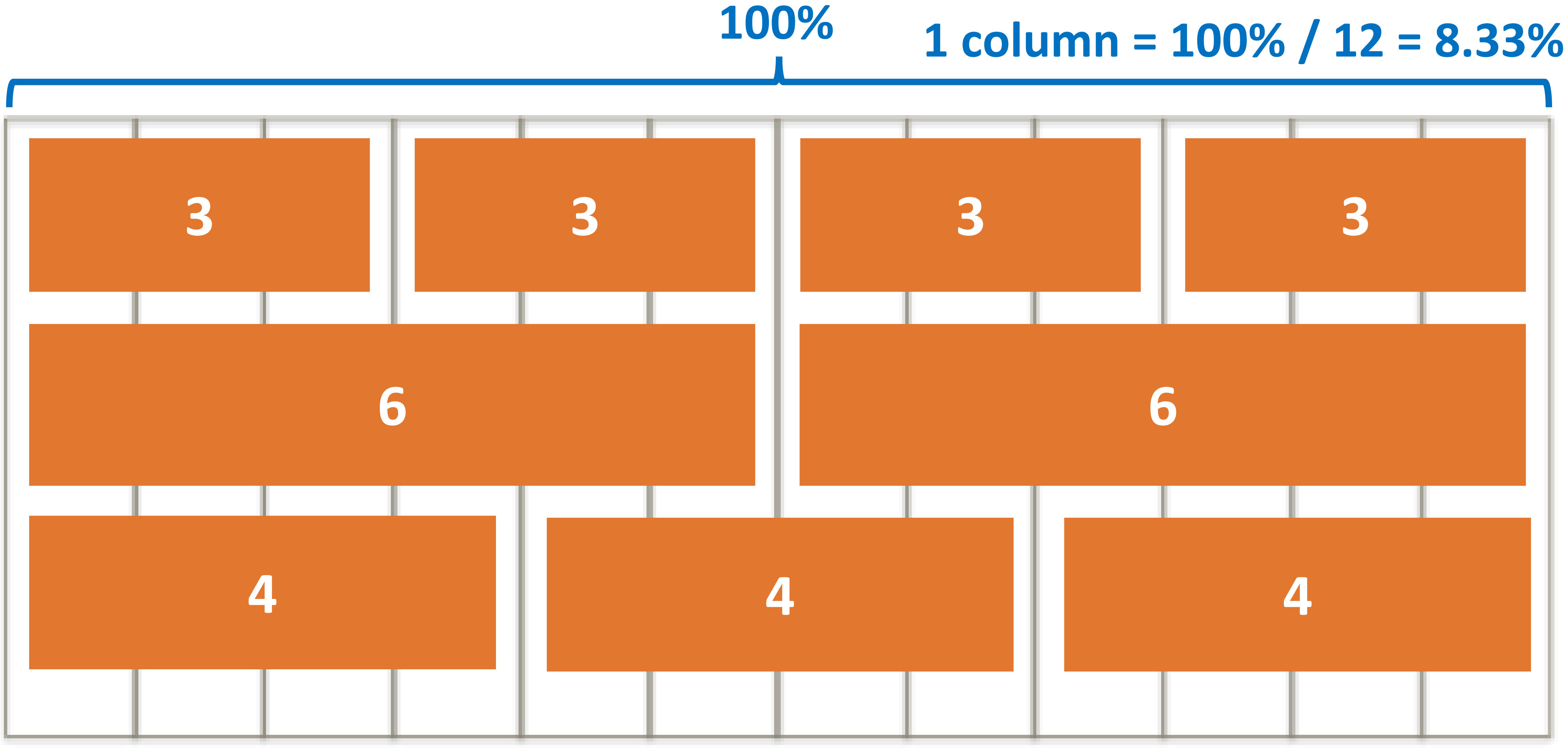


3 3 3 3

6 6

4 4 4

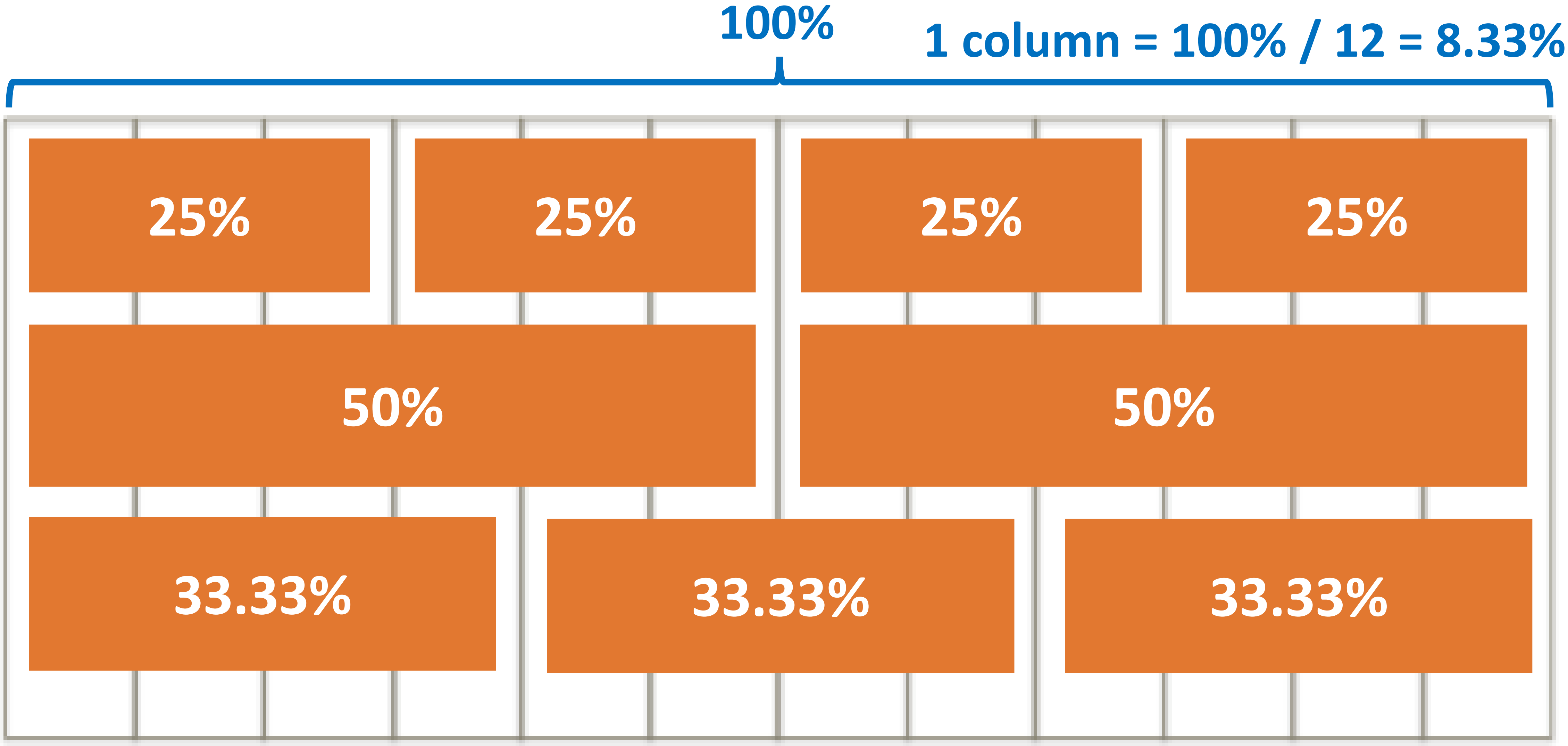
12-Column Grid Responsive Layout



100%

1 column = 100% / 12 = 8.33%

12-Column Grid Responsive Layout



100%

1 column = 100% / 12 = 8.33%

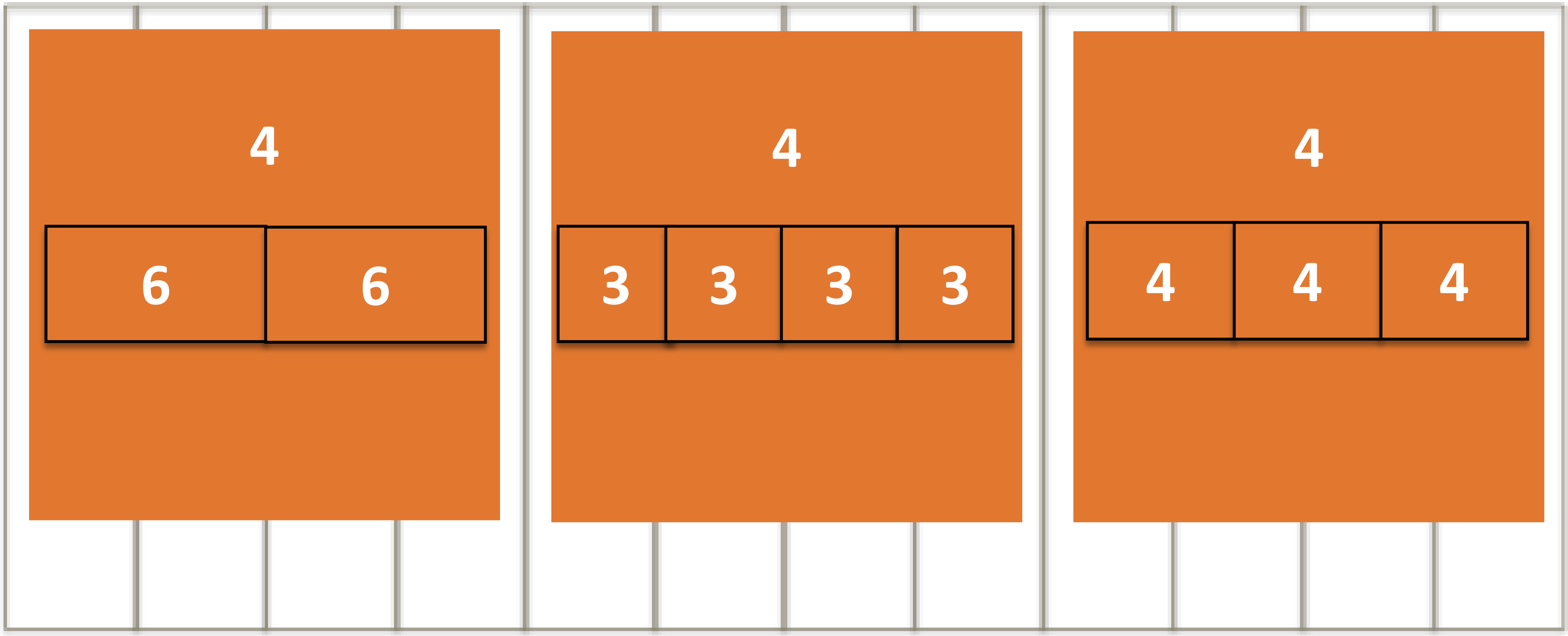
25% 25% 25% 25%

50% 50%

33.33% 33.33% 33.33%

12-Column Grid Responsive Layout

Nested Grids



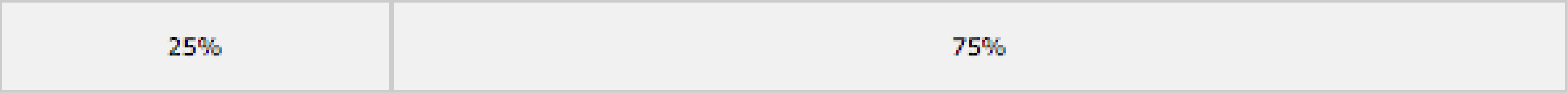
Building a Responsive Grid-View

- 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;
  }
```

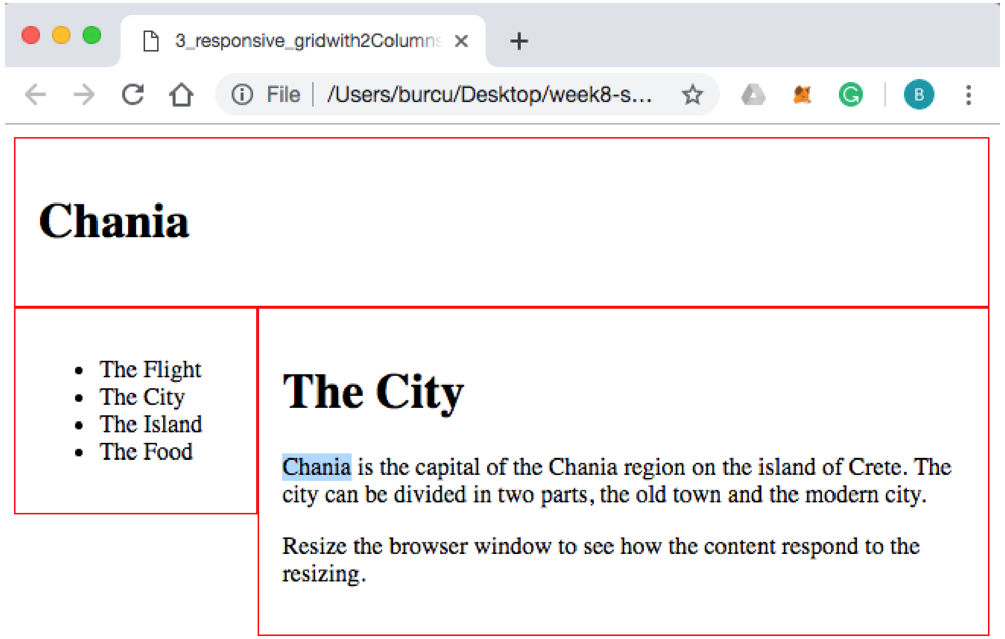
Example1: Building a Responsive Grid-View

A simple responsive web page, with two columns



25%	75%

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


3_responsive_gridwith2Columns
File | /Users/burcu/Desktop/week8-s...
B

Chania

- The Flight
- The City
- The Island
- The Food

The City

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.

Resize the browser window to see how the content respond to the resizing.

week9-sampleCodes/cssResponsive/3_responsive_gridwith2Columns.html

Example2: Building a Responsive Grid-View

- Previous example is fine if the web page only contains two columns
- However, we want to use a responsive grid-view with 12 columns, to have more control over the web page
- First, we must calculate the percentage for one column:

  100% / 12 columns = 8.33%
- Then we make one class for each of the 12 columns, **class="col-"** and a number defining how many columns the section should span →

css

```
.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%;}
```

- All these columns should be floating to the left, and

```
[class*="col-"] {
  float: left;
  padding: 15px;
  border: 1px solid red;
}
```

← have a padding of 15px

Example2: Building a Responsive Grid-View

- Each row should be wrapped in a **<div>**
- The number of columns inside a row should always add up to 12:

HTML

```
<div class="row">
  <div class="col-3">...</div> <!-- 25% -->
  <div class="col-9">...</div> <!-- 75% -->
</div>
```

- The columns inside a row are all floating to the left and are therefore taken out of the flow of the page, and other elements will be placed as if the columns do not exist
  - To prevent this, we will add a style that clears the flow:

CSS

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

week9-sampleCodes/cssResponsive/4_responsive_gridGeneral.html

Example2: Building a Responsive Grid-View

- We may also want to add some styles and colors to make it look better:

```
html {font-family: "Lucida Sans", sans-serif;}

.header {background-color: #9933cc; color: #ffffff; padding: 15px;}

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

.menu li {padding: 8px; margin-bottom: 7px; background-color :#33b5e5; color: #ffffff;
          box-shadow: 0 1px 3px rgba(0,0,0,0.12), 0 1px 2px rgba(0,0,0,0.24);}

.menu li:hover {background-color: #0099cc;}
```

***Notice** that the webpage in the example does not look good when you resize the browser window to a very small width*

week9-sampleCodes/cssResponsive/4_responsive_gridGeneral.html

Outline

- Responsive Web Design (RDW)
- **CSS Responsive**
  - RDW Viewport
  - RDW Grid View
  - **RDW Media Queries**
  - RDW Images
  - RDW Videos
  - RDW Frameworks
- Bootstrap
  - What is Bootstrap?
  - How to Install
  - Create First Web Page
  - Bootstrap Containers
  - Bootstrap Grid System Basics
  - Sample Web Site with Bootstrap
- How to check if a website is Mobile Responsive?
- Summary

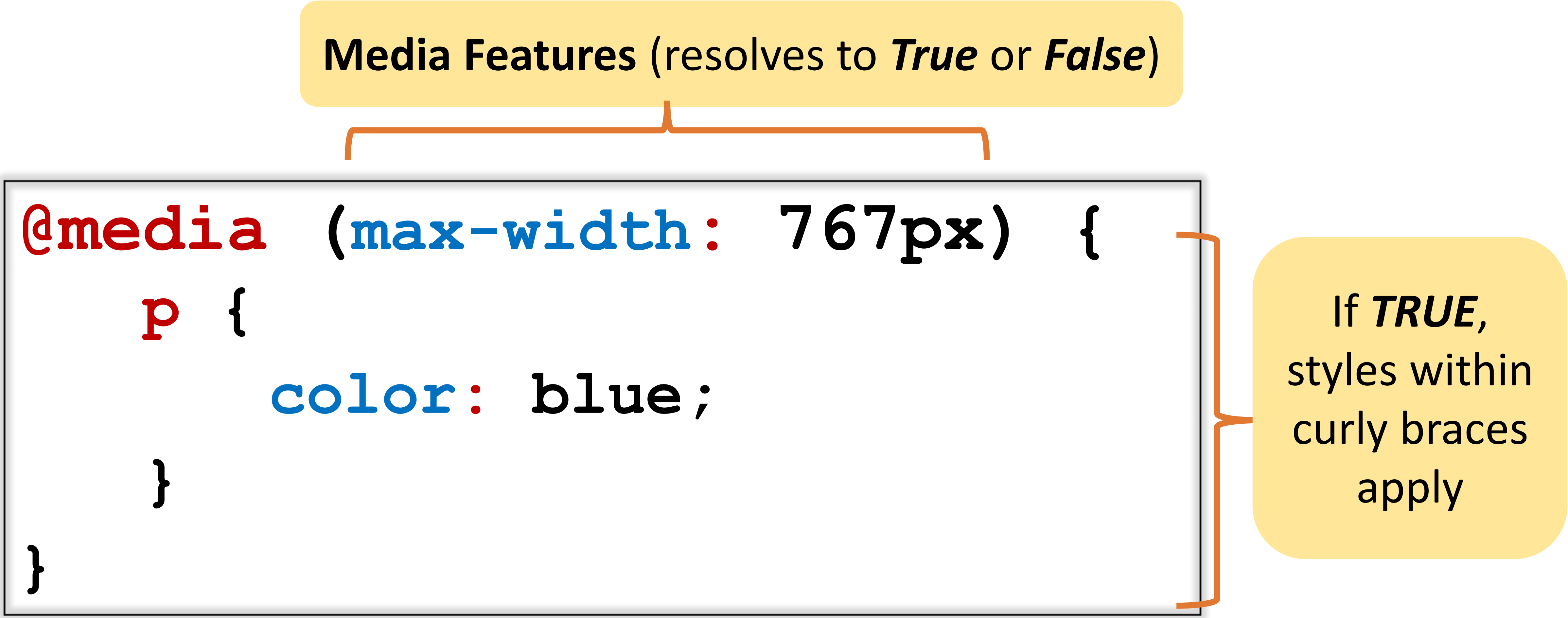
CSS Media Queries

- Media query is a CSS technique introduced in CSS3
- Media queries can be used to check many things, such as:
  - width and height of the viewport
  - width and height of the device
  - orientation (is the tablet/phone in landscape or portrait mode?)
  - resolution
- Allow us to group styles together and target them to devices based on some criteria

  Ex: Target a device by its **width**, its **height**, or **orientation** like ***landscape*** or ***portrait***
- It uses the **@media** rule to include a block of CSS properties ***only if a certain condition is true***

https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries

Media Query Syntax



Media Features (resolves to ***True*** or ***False***)

```
@media (max-width: 767px) {
    p {
        color: blue;
    }
}
```

If ***TRUE***, styles within curly braces apply

Media Query Common Features

```
@media (max-width: 800px) {…}
```

```
@media (min-width: 800px) {…}
```

```
@media (orientation: portrait) {…}
```

```
@media (screen) {…}
```

```
@media (print) {…}
```

Media Query Common Logical Operators

Devices with width within a range

```
@media (min-width: 768px) and (max-width: 991px) {…}
```

Comma is equivalent to **OR**

```
@media (max-width: 767px) , (min-width: 992px) {…}
```

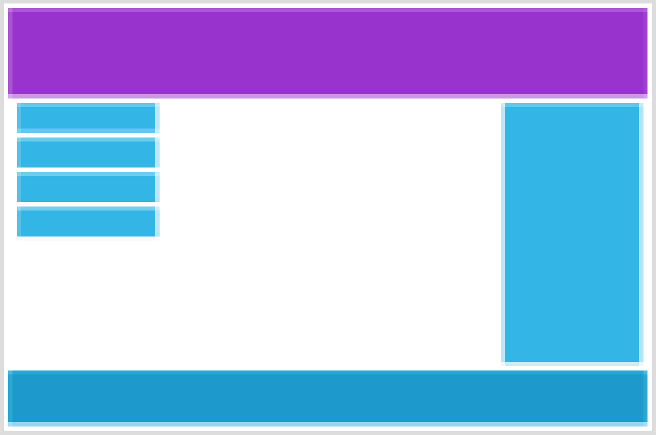
Media Query Common Approach

```
p { color: blue; } /* base styles */
...
@media (min-width: 1200px) {
...
}
@media (min-width: 992px) and (max-width: 1199px)
{
...
}
...
```

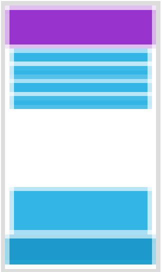
Careful not to overlap range boundaries!

Example2: Building a Responsive Grid-View – (continue)

- In an earlier example, we made a web page with rows and columns, and it was responsive, but it did not look good on a small screen
- Media queries can help with that
- We can **add a breakpoint** where certain parts of the design will behave differently on each side of the breakpoint



Desktop



Phone

Example2: Building a Responsive Grid-View – (continue) Add a Breakpoint

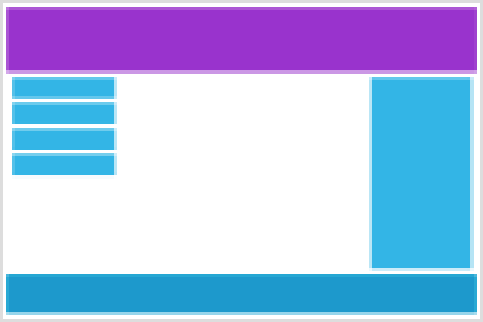
```
<style>
   /* For mobile phones: */
  [class*="col-"] { width: 100%; }

    @media only screen and (min-width: 768px)
  {  /* For desktop: */
      .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%;} }
</style>
```

week9-sampleCodes/cssResponsive/5_responsive_addBreakpoint.html

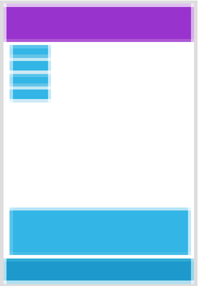
```
<body>
  <div class="row">
    <div class="col-3 menu">
      <ul>
        <li>The Flight</li>  <li>The City</li>
        <li>The Island</li> <li>The Food</li>
      </ul>
     </div>

    <div class="col-6">
      <h1>The City</h1>
      <p>Chania…. .</p>
    </div>

    <div class="col-3 right">
      <div class="aside">
        <h2>What?</h2> <p>Chania is … .</p>
         <h2>Where?</h2> <p>Crete is … .</p>
         <h2>How?</h2> <p>You can … .</p>
       </div>
    </div>
  </div>
</body>
```

Example2: Building a Responsive Grid-View – (continue) Add an Other Breakpoint

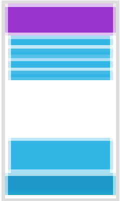
- We can also insert a breakpoint between tablets and mobile phones



Desktop



Tablet



Phone

- We can do this by adding one more media query (at 600px), and a set of new classes for devices larger than 600px (but smaller than 768px)

week9-sampleCodes/cssResponsive/6_responsive_after.html
https://www.w3schools.com/css/css_rwd_mediaqueries.asp

Typical Device Breakpoints

- There are tons of screens and devices with different heights and widths, so it is hard to create an exact breakpoint for each device

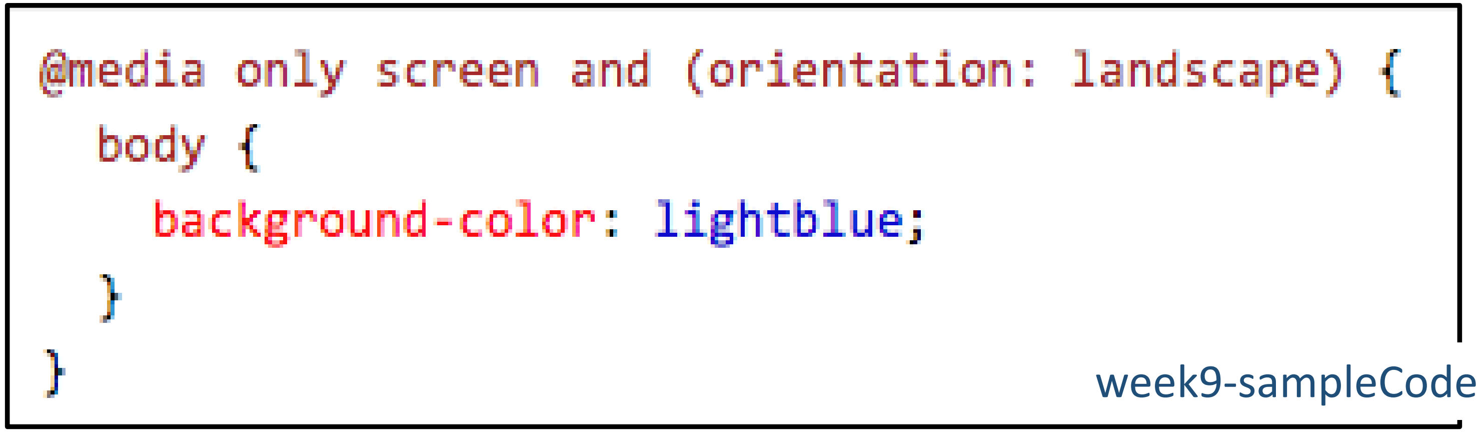
- To keep things simple you could target five groups →

```
/* 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) {...}
```

week9-sampleCodes/cssMediaQueries/typicalDeviceBreakpoints.html

Orientation: Portrait / Landscape

- Media queries can also be used to change layout of a page depending on the orientation of the browser.
- You can have a set of CSS properties that will only apply when the browser window is wider than its height, a so called "Landscape" orientation:
  - *Ex*: The web page will have a light-blue background if the orientation is in landscape mode:



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

week9-sampleCodes/cssMediaQueries/orientation.html

Outline

- Responsive Web Design (RDW)
- **CSS Responsive**
  - RDW Viewport
  - RDW Grid View
  - RDW Media Queries
  - **RDW Images**
  - RDW Videos
  - RDW Frameworks
- Bootstrap
  - What is Bootstrap?
  - How to Install
  - Create First Web Page
  - Bootstrap Containers
  - Bootstrap Grid System Basics
  - Sample Web Site with Bootstrap
- How to check if a website is Mobile Responsive?
- Summary

Responsive Web Design - Images

- **Using The width Property**
  - If the **width** property is set to a **percentage** and the **height** is set to "**auto**", the image will be responsive and scale up and down
  - ***Notice*** that the image can be scaled up to be larger than its original size by this way → A better solution, in many cases, will be to use the **max-width** property instead

```
img {
  width: 100%;
  height: auto
;
}
```

- **Using The max-width Property**
  - If the **max-width** property is set to **100%**, the image will scale down if it has to, but never scale up to be larger than its original size

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

```
img {
  max-width: 100%;
  height: auto;
}
```

- **Example2: Building a Responsive Grid-View –(continue) Add an Image to the Example Web Page**

week9-sampleCodes/cssResponsive/7_responsive_after_withImage.html

Outline

- Responsive Web Design (RDW)
- **CSS Responsive**
  - RDW Viewport
  - RDW Grid View
  - RDW Media Queries
  - RDW Images
  - **RDW Videos**
  - RDW Frameworks
- Bootstrap
  - What is Bootstrap?
  - How to Install
  - Create First Web Page
  - Bootstrap Containers
  - Bootstrap Grid System Basics
  - Sample Web Site with Bootstrap
- How to check if a website is Mobile Responsive?
- Summary

Responsive Web Design - Videos

- **Using The width Property**
  - If the **width** property is set to 100%, the video player will be responsive and scale up and down
  - ***Notice*** that the video can be scaled up to be larger than its original size by this way → A better solution, in many cases, will be to use the **max-width** property instead

```
video {
  width: 100%;
  height: auto
;
}
```

- **Using The max-width Property**

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

  - If the **max-width** property is set to **100%**, the video player will scale down if it has to, but never scale up to be larger than its original size

```
video {
  max-width: 100%;
  height: auto;
}
```

- **Example2: Building a Responsive Grid-View –(continue) Add a Video to the Example Web Page**

week9-sampleCodes/cssResponsive/8_responsive_after_withVideo.html

Outline

- Responsive Web Design (RDW)
- CSS Responsive
  - RDW Viewport
  - RDW Grid View
  - RDW Media Queries
  - RDW Images
  - RDW Videos
  - **RDW Frameworks**
- Bootstrap
  - What is Bootstrap?
  - How to Install
  - Create First Web Page
  - Bootstrap Containers
  - Bootstrap Grid System Basics
  - Sample Web Site with Bootstrap
- How to check if a website is Mobile Responsive?
- Summary

Responsive Web Design - Frameworks

- There are many existing CSS Frameworks that offer Responsive Design
- They are free, and easy to use

- **Bootstrap**
  - Bootstrap is one of the popular frameworks, which uses HTML, CSS and jQuery to make responsive web pages

- **W3.CSS**
  - An other popular way to create a responsive design, is to use a responsive style sheet, like W3.CSS

Outline

- Responsive Web Design (RDW)
- CSS Responsive
  - RDW Viewport
  - RDW Grid View
  - RDW Media Queries
  - RDW Images
  - RDW Videos
  - RDW Frameworks
- **Bootstrap**
  - **What is Bootstrap?**
  - How to Install
  - Create First Web Page
  - Bootstrap Containers
  - Bootstrap Grid System Basics
  - Sample Web Site with Bootstrap
- How to check if a website is Mobile Responsive?
- Summary

What is Bootstrap?

Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.

What is Bootstrap?

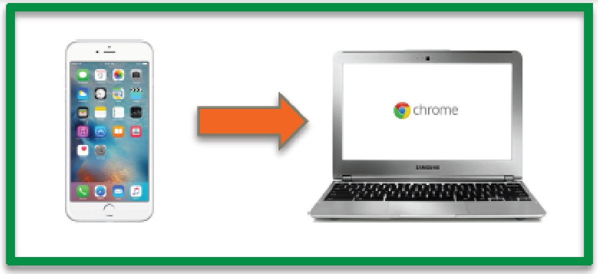
Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.

CSS

Mostly CSS classes

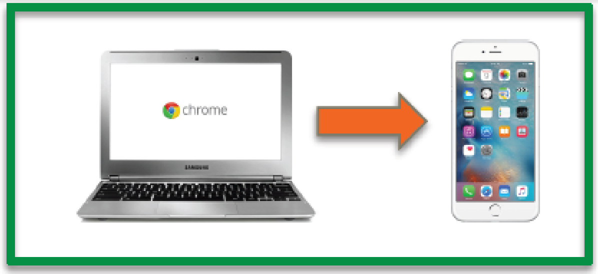
What is Bootstrap?

Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.



✔

Which to code first?



✔

What is Bootstrap?

✓ Mobile First == PLAN mobile from the start

✓ CSS Framework is mobile ready

Why Use Bootstrap?

Advantages of 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, mobile-first styles are part of the core framework
- **Browser compatibility:** Bootstrap 4 is compatible with all modern browsers (Chrome, Firefox, Internet Explorer 10+, Edge, Safari, and Opera)

Disadvantages of Bootstrap (vs. Writing Your Own)?

Too big, too bloated

- A lot of features you will probably never use
  - Use selective download

- You can write your own that's more smaller
  - But it will take you a LOT longer to write it as well

Outline

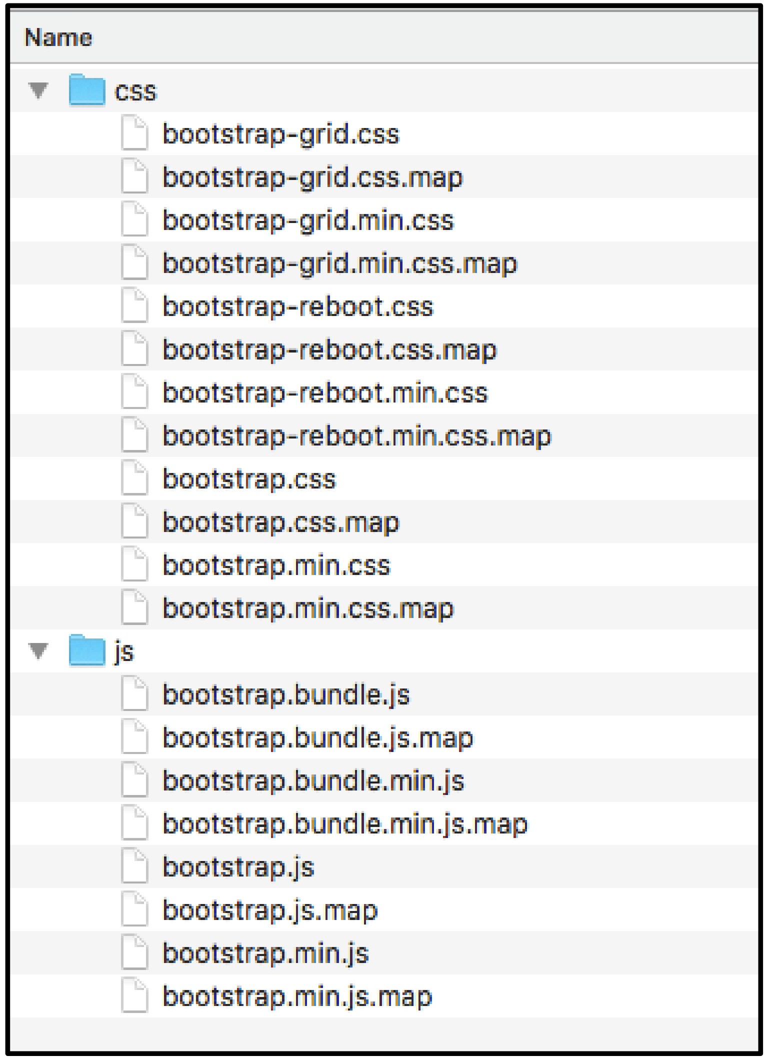
- Responsive Web Design (RDW)
- CSS Responsive
  - RDW Viewport
  - RDW Grid View
  - RDW Media Queries
  - RDW Images
  - RDW Videos
  - RDW Frameworks
- **Bootstrap**
  - What is Bootstrap?
  - **How to Install**
  - Create First Web Page
  - Bootstrap Containers
  - Bootstrap Grid System Basics
  - Sample Web Site with Bootstrap
- How to check if a website is Mobile Responsive?
- Summary

How to Install: Basic Bootstrap Setup

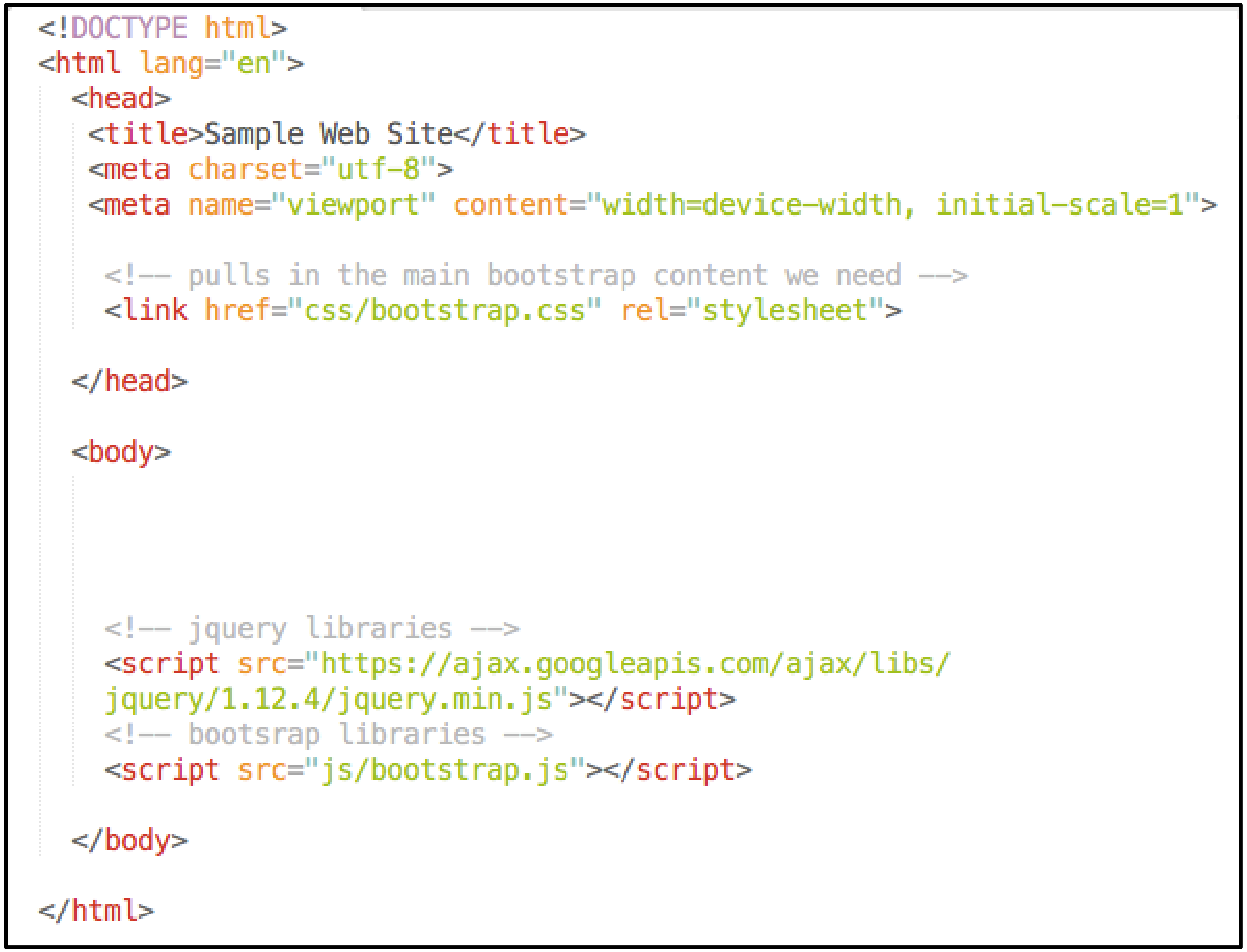
- You can download bootstrap for yourself from these addresses
  - Download Bootstrap from getbootstrap.com
    http://getbootstrap.com/getting-started/
  - Download Bootstrap from Barebones Bootstrap web page
    https://startbootstrap.com/templates/bare/
- You can include Bootstrap from a CDN
  - If you don't want to download and host Bootstrap X yourself, you can include it from a CDN (Content Delivery Network)
- A zip file containing basic starter code to get you up and running with bootstrap, can be found under week9-sampleCodes → "bootstrap-4.4.1-dist.zip"

https://www.w3schools.com/bootstrap4/bootstrap_get_started.asp

Basic Bootstrap Setup



```
Name
▼ css
      bootstrap-grid.css
      bootstrap-grid.css.map
      bootstrap-grid.min.css
      bootstrap-grid.min.css.map
      bootstrap-reboot.css
      bootstrap-reboot.css.map
      bootstrap-reboot.min.css
      bootstrap-reboot.min.css.map
      bootstrap.css
      bootstrap.css.map
      bootstrap.min.css
      bootstrap.min.css.map
▼ js
      bootstrap.bundle.js
      bootstrap.bundle.js.map
      bootstrap.bundle.min.js
      bootstrap.bundle.min.js.map
      bootstrap.js
      bootstrap.js.map
      bootstrap.min.js
      bootstrap.min.js.map
```



```
<!DOCTYPE html>
<html lang="en">
  <head>
   <title>Sample Web Site</title>
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- pulls in the main bootstrap content we need -->
    <link href="css/bootstrap.css" rel="stylesheet">

  </head>

  <body>




    <!-- jquery libraries -->
    <script src="https://ajax.googleapis.com/ajax/libs/
jquery/1.12.4/jquery.min.js"></script>
    <!-- bootsrap libraries -->
    <script src="js/bootstrap.js"></script>

  </body>

</html>
```

Outline

- Responsive Web Design (RDW)
- CSS Responsive
  - RDW Viewport
  - RDW Grid View
  - RDW Media Queries
  - RDW Images
  - RDW Videos
  - RDW Frameworks
- **Bootstrap**
  - What is Bootstrap?
  - How to Install
  - **Create First Web Page**
  - Bootstrap Containers
  - Bootstrap Grid System Basics
  - Sample Web Site with Bootstrap
- How to check if a website is Mobile Responsive?
- Summary

Create First Web Page With Bootstrap

1. Add the HTML5 doctype

- Bootstrap uses HTML elements and CSS properties that require the HTML5 doctype
- Always include the HTML5 doctype at the beginning of the page, along with the **lang** attribute and the correct **character set**

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
  </head>
</html>
```

Create First Web Page With Bootstrap

2. Bootstrap is mobile-first

- Bootstrap is designed to be responsive to mobile devices
- Mobile-first styles are part of the core framework
- To ensure proper rendering and touch zooming, add the following **<meta>** tag inside the **<head>** element:

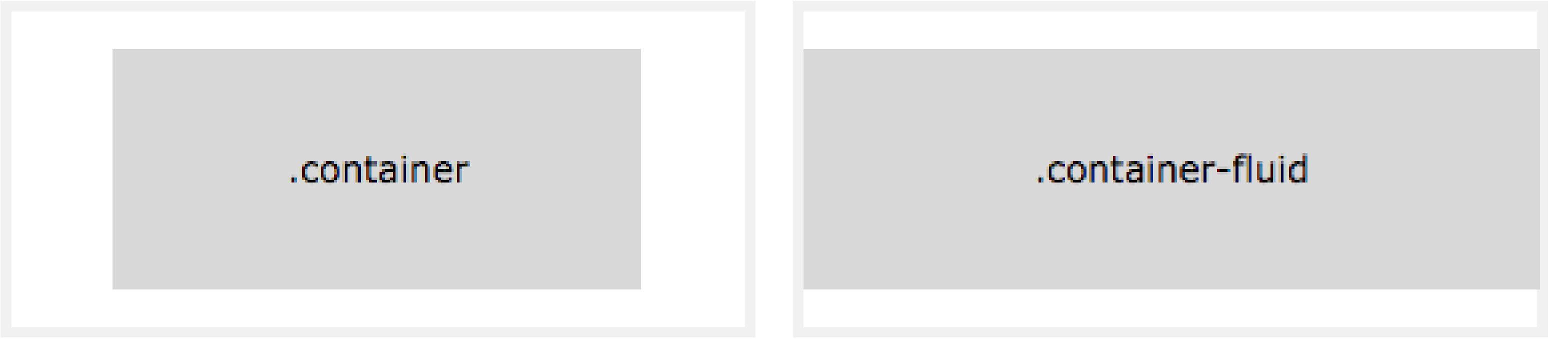
```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

  - 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** part sets the initial zoom level when the page is first loaded by the browser

Create First Web Page With Bootstrap

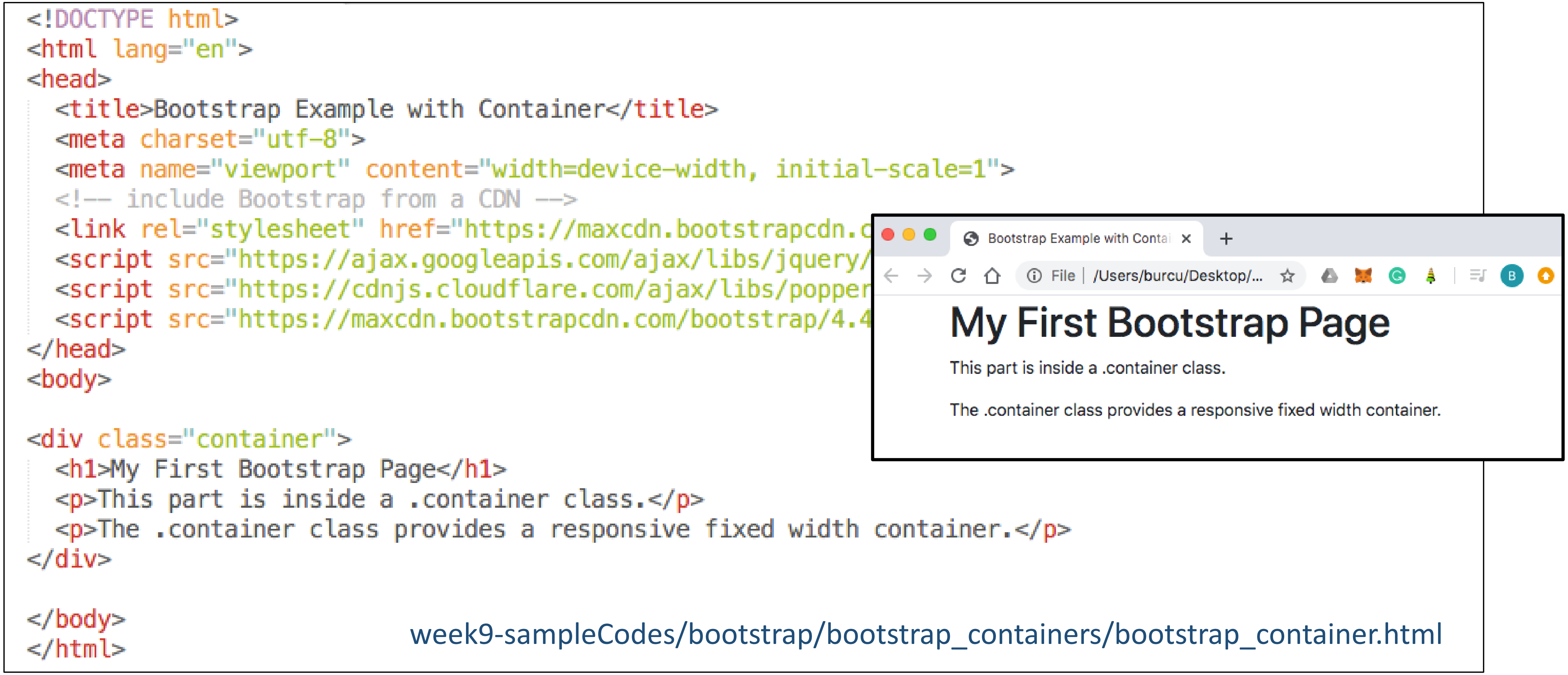
3. Containers

- Bootstrap requires a containing element to wrap site contents
- There are two container classes to choose from:
    1. The **.container** class provides a responsive **fixed width container**
    2. The **.container-fluid** class provides a **full width container**, spanning the entire width of the viewport



https://www.w3schools.com/bootstrap4/bootstrap_get_started.asp

Container Example: A basic Bootstrap page (with fixed width container)



```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example with Container</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!-- include Bootstrap from a CDN -->
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.c
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4
</head>
<body>

<div class="container">
  <h1>My First Bootstrap Page</h1>
  <p>This part is inside a .container class.</p>
  <p>The .container class provides a responsive fixed width container.</p>
</div>

</body>
</html>
```

Bootstrap Example with Contai ×
File | /Users/burcu/Desktop/...

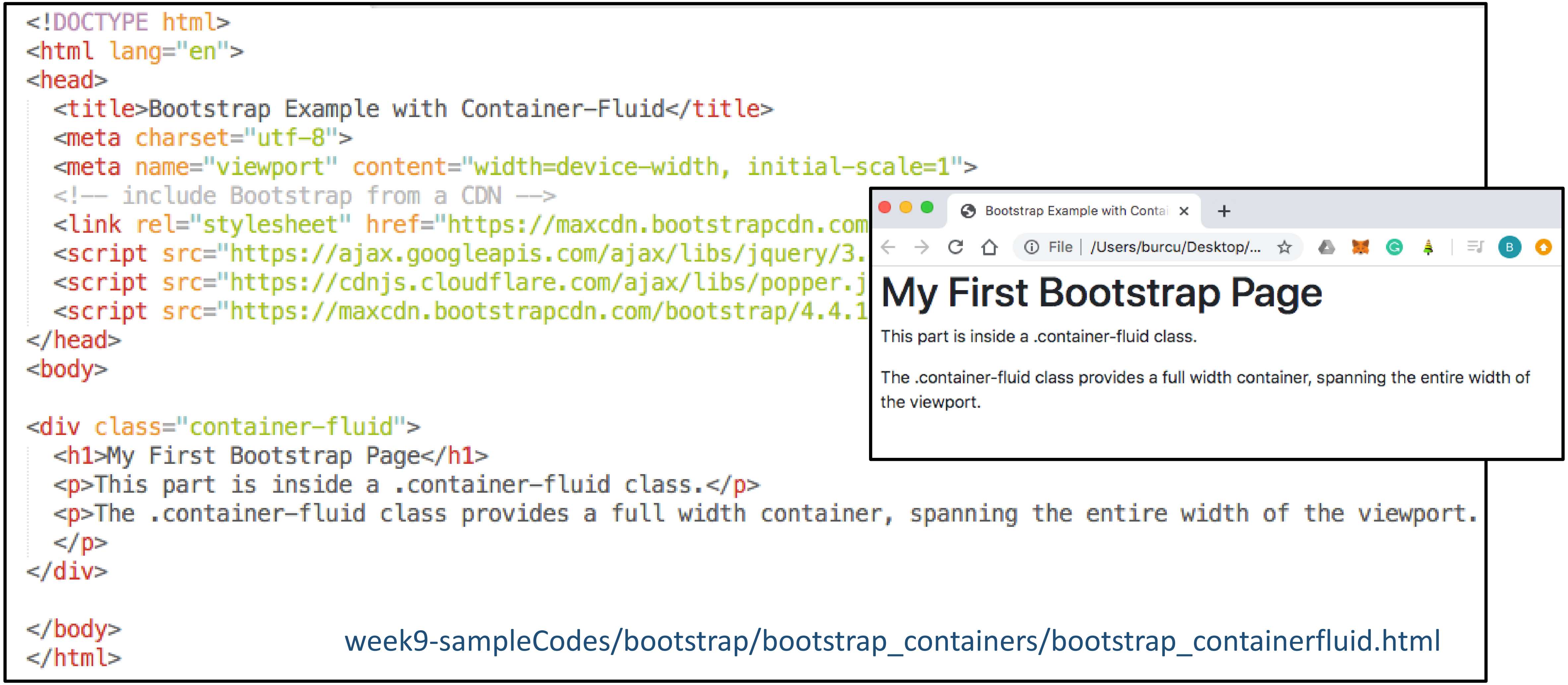
My First Bootstrap Page

This part is inside a .container class.

The .container class provides a responsive fixed width container.

week9-sampleCodes/bootstrap/bootstrap_containers/bootstrap_container.html

Container Fluid Example: A basic Bootstrap page (with a responsive full width container)



```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example with Container-Fluid</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!-- include Bootstrap from a CDN -->
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.j
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1
</head>
<body>

<div class="container-fluid">
  <h1>My First Bootstrap Page</h1>
  <p>This part is inside a .container-fluid class.</p>
  <p>The .container-fluid class provides a full width container, spanning the entire width of the viewport.
  </p>
</div>

</body>
</html>
```

Bootstrap Example with Contai
File | /Users/burcu/Desktop/...

My First Bootstrap Page

This part is inside a .container-fluid class.

The .container-fluid class provides a full width container, spanning the entire width of the viewport.

week9-sampleCodes/bootstrap/bootstrap_containers/bootstrap_containerfluid.html

Outline

- Responsive Web Design (RDW)
- CSS Responsive
  - RDW Viewport
  - RDW Grid View
  - RDW Media Queries
  - RDW Images
  - RDW Videos
  - RDW Frameworks
- **Bootstrap**
  - What is Bootstrap?
  - How to Install
  - Create First Web Page
  - **Bootstrap Containers**
  - Bootstrap Grid System Basics
  - Sample Web Site with Bootstrap
- How to check if a website is Mobile Responsive?
- Summary

Bootstrap Containers

Fixed Container

- Use the **.container** class to create a responsive, fixed-width container
- Note that its width (**max-width**) will change on different screen sizes:

	Extra Small	Small	Medium	Large	Extra Large
	$< 576px$	$\geq 576px$	$\geq 768px$	$\geq 992px$	$\geq 1200px$
max-width	100%	540px	720px	960px	1140px

Fluid Container

- Use the **.container-fluid** class to create a full width container, that will always span the entire width of the screen (**width** is always **100%**)

Bootstrap Containers

Container Padding

- By default, containers have 15px left and right padding, with no top or bottom padding
- Therefore, we often use **spacing utilities**, such as extra padding and margins to make them look even better
- *Ex*:, **.pt-3** means "add a top padding of 16px"

```
<div class="container pt-3"></div>
```

Container Border and Color

- Other utilities, such as borders and colors, are also often used together with containers

week9-sampleCodes/bootstrap/bootstrap_containers/bootstrap_containerpadding.html
week9-sampleCodes/bootstrap/bootstrap_containers/bootstrap_containerbordercolor.html

Bootstrap Containers

Responsive Containers

- You can also use the **.container-sm|md|lg|xl** classes to create responsive containers
- The **max-width** of the container will change on different screen sizes/viewports week9-sampleCodes/bootstrap/bootstrap_containers/bootstrap_responsivecontainer.html

Class	Extra Small < 576px	Small $\geq 576px$	Medium $\geq 768px$	Large $\geq 992px$	Extra Large $\geq 1200px$
.container-sm	100%	540px	720px	960px	1140px
.container-md	100%	100%	720px	960px	1140px
.container-lg	100%	100%	100%	960px	1140px
.container-xl	100%	100%	100%	100%	1140px

Outline

- Responsive Web Design (RDW)
- CSS Responsive
  - RDW Viewport
  - RDW Grid View
  - RDW Media Queries
  - RDW Images
  - RDW Videos
  - RDW Frameworks
- **Bootstrap**
  - What is Bootstrap?
  - How to Install
  - Create First Web Page
  - Bootstrap Containers
  - **Bootstrap Grid System Basics**
  - Sample Web Site with Bootstrap
- How to check if a website is Mobile Responsive?
- Summary

Bootstrap Grid System Basics

Must be inside ***container*** (fixed-width) or (***container-fluid*** (full-width))

```
<div class=“container”>
    <div class=“row”>
        <div class=“col-md-4”>Col 1</div>
        ...
    </div>
</div>
```

Bootstrap Grid System Basics

Creates horizontal groups of columns

```
<div class="container">
    <div class="row">
        <div class="col-md-4">Col 1</div>
        ...
    </div>
</div>
```

Applies negative left/right margins

Bootstrap Grid System Basics

```
<div class=“container”>
    <div class=“row”>
        <div class=“col-md-4”>Col 1</div>
        ...
    </div>
</div>
```

Bootstrap Grid System Basics

SIZE identifier identifies at which breakpoint specified column spans will be ignored and all elements will collapse

```
col-SIZE-SPAN
```

- How many columns element should span (out of 12)
- Values: 1 through 12

Example:

.col-sm-1 spans 1 column,

.col-sm-4 spans 4 columns,

.col-sm-6 spans 6 columns, etc.

Note: Make sure that the sum always adds up to 12!

- Screen width range identifier

.col-	(extra small devices - screen width < 576px)
.col-sm-	(small devices - screen width ≥ 576px)
.col-md-	(medium devices - screen width ≥ 768px)
.col-lg-	(large devices - screen ≥ 992px)
.col-xl-	(xlarge devices - screen width ≥ 1200px)

- Columns will **collapse** (i.e., **stack**) below that width
  - Unless another rule applies

https://www.w3schools.com/bootstrap4/bootstrap_grid_examples.asp

Bootstrap Grid System Basics

```
<header class=“container”>
    <nav class=“row”>
        <div class=“col-md-4”>Col 1</div>
        ...
    </nav>
</header>
```

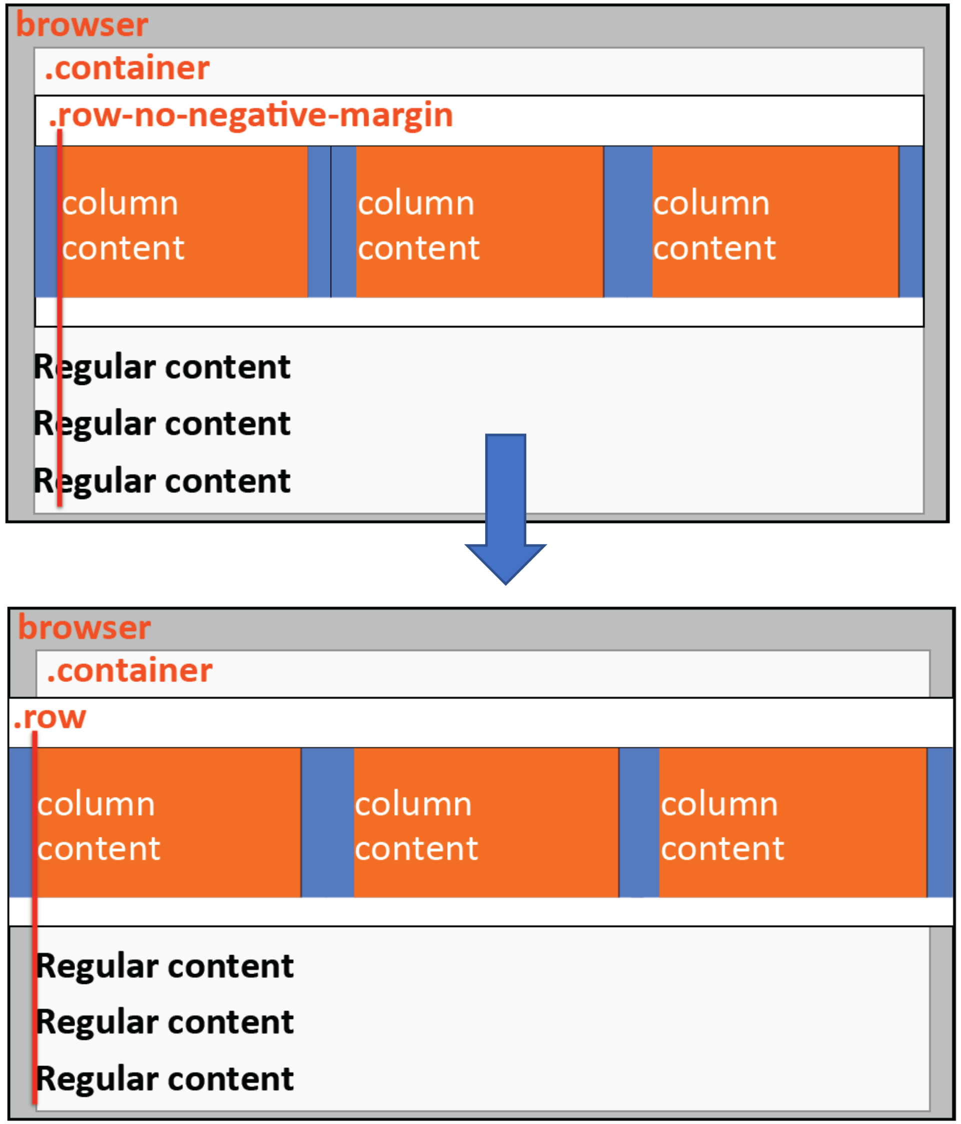
Bootstrap Grid System Rules

Some Bootstrap 4 grid system rules:

- Rows must be placed within a **.container** (fixed-width) or **.container-fluid** (full-width) for proper alignment and padding
- Use rows to create horizontal groups of columns
- Content should be placed within columns, and only columns may be immediate children of rows
- Predefined classes like **.row** and **.col-sm-4** are available for quickly making grid layouts
- Columns create gutters (gaps between column content) via padding. That padding is offset in rows for the first and last column via negative margin on **.rows**

https://www.w3schools.com/bootstrap4/bootstrap_grid_system.asp

Why Negative Row Margin?



Bootstrap Grid System Rules

Some Bootstrap 4 grid system rules (continue):

- Grid columns are created by specifying the number of 12 available columns you wish to span
  - *Ex*: Three equal columns would use three **.col-sm-4**
- Column widths are in percentage, so they are always fluid and sized relative to their parent element
- The biggest **difference between Bootstrap 3 and Bootstrap 4** is that Bootstrap 4 now uses **flexbox**, instead of floats
  - One big advantage with flexbox is that grid columns without a specified width will automatically layout as "equal width columns" (and equal height)
  - *Ex*: Three elements with **.col-sm** will each automatically be 33.33% wide from the small breakpoint and up

https://www.w3schools.com/css/css3_flexbox.asp

Outline

- Responsive Web Design (RDW)
- CSS Responsive
  - RDW Viewport
  - RDW Grid View
  - RDW Media Queries
  - RDW Images
  - RDW Videos
  - RDW Frameworks
- **Bootstrap**
  - What is Bootstrap?
  - How to Install
  - Create First Web Page
  - Bootstrap Containers
  - Bootstrap Grid System Basics
  - **Sample Web Site with Bootstrap**
- How to check if a website is Mobile Responsive?
- Summary

Sample Web Site with Bootstrap

In order to see the change of a simple web site with the use of Bootstrap, let’s add necessary code step by step

week9-sampleCodes/bootstrap/ bootstrap_sample_website/step1_add_nav_bar/index.before.html

Step 1: Adding a navigational bar

week9-sampleCodes/bootstrap/ bootstrap_sample_website/step1_add_nav_bar/index.after.html

Step 2: Adding collapsible navigational bar

week9-sampleCodes/bootstrap/ bootstrap_sample_website/step2_add_collapsible_nav_bar

Sample Web Site with Bootstrap

Step 3: Build your site by adding new web pages

week9-sampleCodes/bootstrap/bootstrap_sample_website/step3_buildTheSiteByAddingPages

Step 4: Add breadcrumb to your pages

week9-sampleCodes/bootstrap/bootstrap_sample_website/step4_add_breadcrumbs

Step 5: Add responsive grid to scale the pictures on different screen

week9-sampleCodes/bootstrap/bootstrap_sample_website/step5_add_responsiveGrid

Step 6: Change the entire look of your web site (without any code change) by using different Bootstrap Themes

week9-sampleCodes/bootstrap/bootstrap_sample_website/step6_change_bootstrapTheme

Outline

- Responsive Web Design (RDW)
- CSS Responsive
  - RDW Viewport
  - RDW Grid View
  - RDW Media Queries
  - RDW Images
  - RDW Videos
  - RDW Frameworks
- Bootstrap
  - What is Bootstrap?
  - How to Install
  - Create First Web Page
  - Bootstrap Containers
  - Bootstrap Grid System Basics
  - Sample Web Site with Bootstrap
- **How to check if a website is Mobile Responsive?**
- Summary

How to check if a website is Responsive?

- Manually testing the website across various devices for responsiveness using the developer tools in the browser can prove to be a very tedious task
- As test automation is slowly gaining importance, there are tools available in the market using which one can quickly determine the responsiveness of a particular website across multiple devices and browsers
- **Responsive by BrowserStack** (https://www.browserstack.com/responsive)
  - Makes mobile responsive testing simple
  - Provides a combination of real devices to test on the fly!

Summary

- Traffic from mobile devices is up to 60% of all website traffic
- Thus, creating a website that works seamlessly on mobile devices is critical
- This is achieved by adopting a responsive design
- Making the website responsive is not straightforward!