Instructor No

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

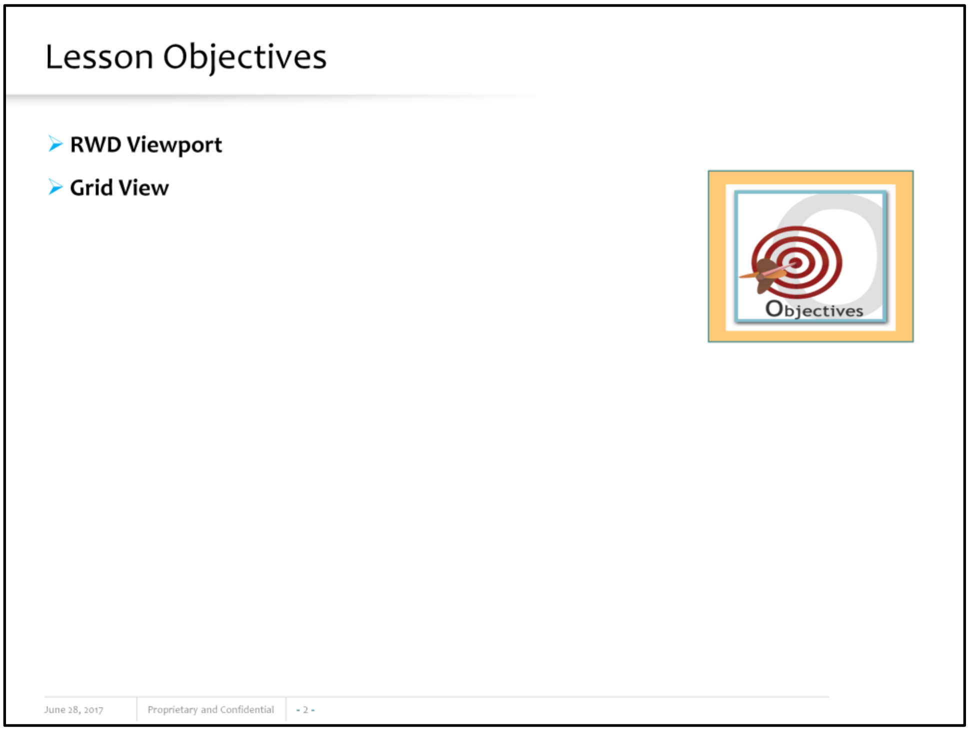
Responsive Web Design

RWD Viewport and Grid view

People matter, results count.

Instructor Notes:

Add instructor notes here.



Lesson Objectives

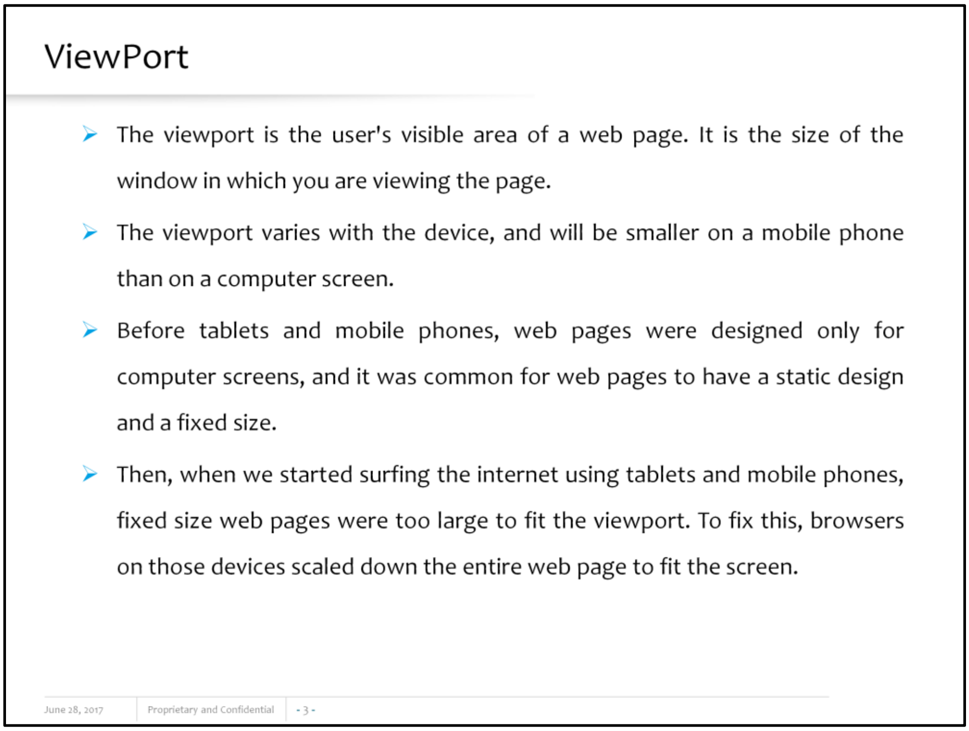
- RWD Viewport
- Grid View

Objectives

June 28, 2017 Proprietary and Confidential - 2 -

Instructor Notes:

Add instructor notes here.

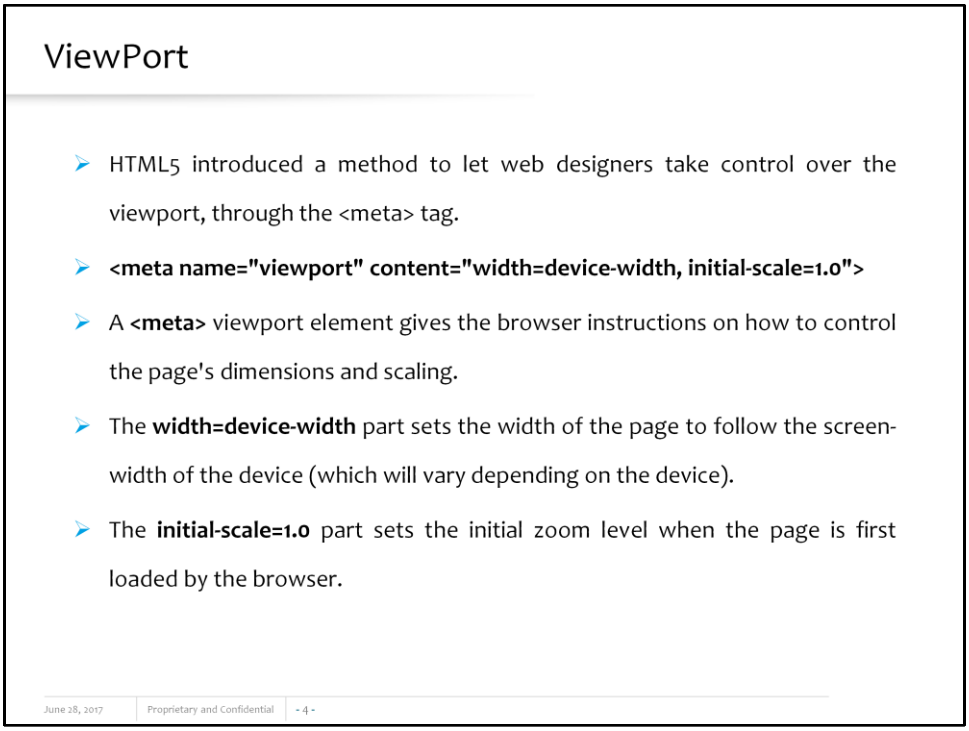


ViewPort

- The viewport is the user's visible area of a web page. It is the size of the window in which you are viewing the 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.

Instructor Notes:

Add instructor notes here.



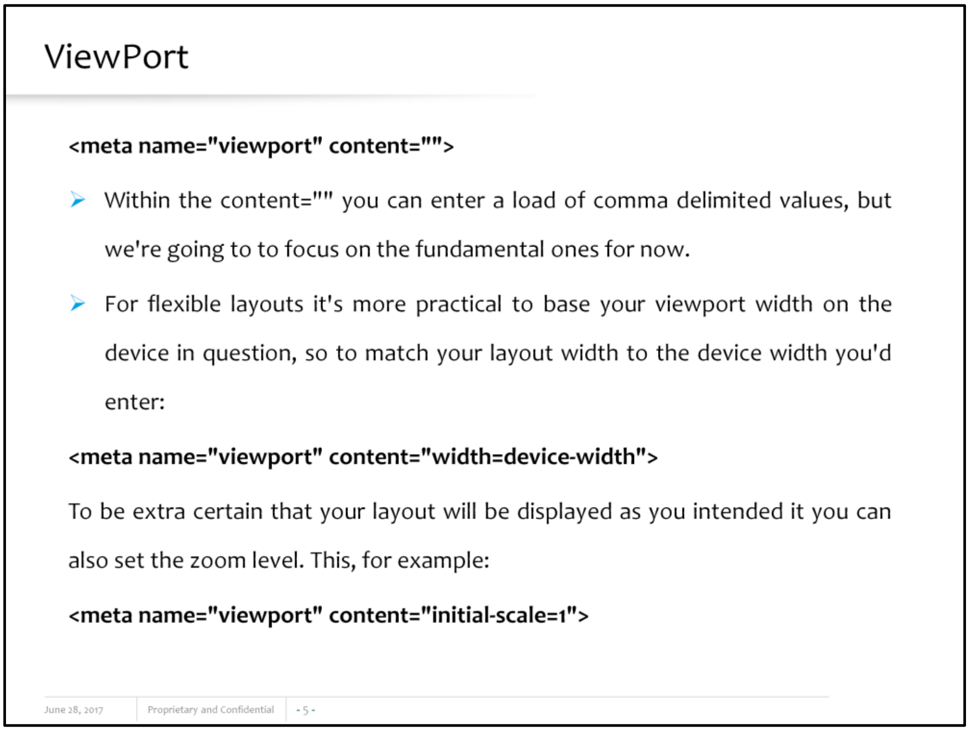
ViewPort

- HTML5 introduced a method to let web designers take control over the viewport, through the <meta> tag.
- **<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.

June 28, 2017 | Proprietary and Confidential | - 4 -

Instructor Notes:

Add instructor notes here.



ViewPort

<meta name="viewport" content="">

- Within the content="" you can enter a load of comma delimited values, but we're going to to focus on the fundamental ones for now.
- For flexible layouts it's more practical to base your viewport width on the device in question, so to match your layout width to the device width you'd enter:

<meta name="viewport" content="width=device-width">

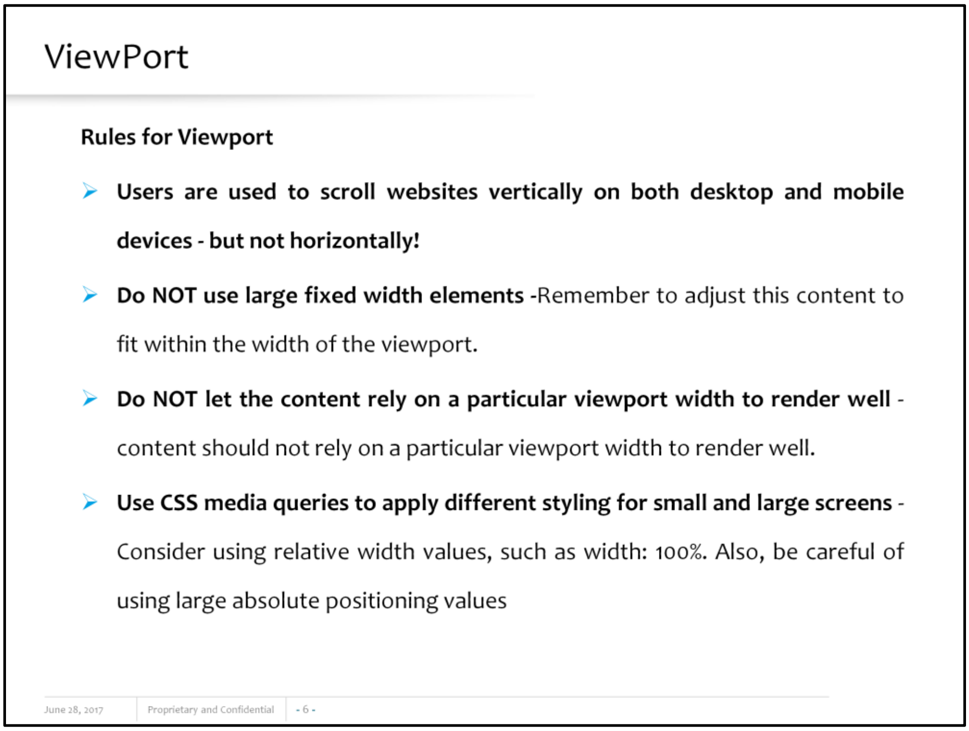
To be extra certain that your layout will be displayed as you intended it you can also set the zoom level. This, for example:

<meta name="viewport" content="initial-scale=1">

June 28, 2017 | Proprietary and Confidential | - 5 -

Instructor Notes:

Add instructor notes here.



ViewPort

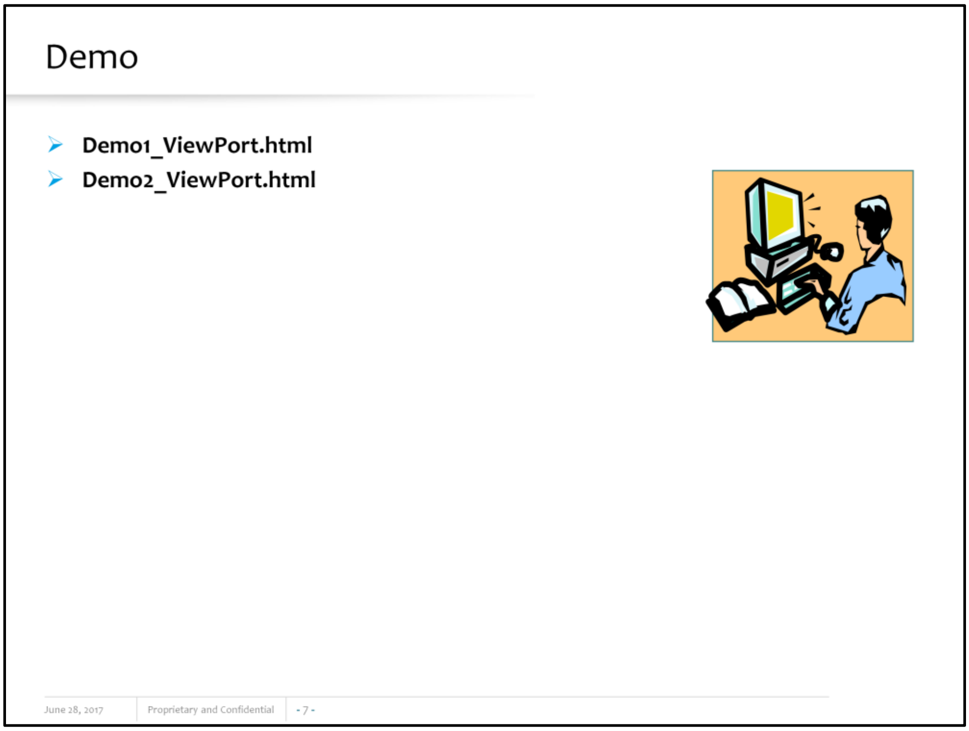
Rules for Viewport

- **Users are used to scroll websites vertically on both desktop and mobile devices - but not horizontally!**
- **Do NOT use large fixed width elements** -Remember to adjust this content to fit within the width of the viewport.
- **Do NOT let the content rely on a particular viewport width to render well** - content should not rely on a particular viewport width to render well.
- **Use CSS media queries to apply different styling for small and large screens** - Consider using relative width values, such as width: 100%. Also, be careful of using large absolute positioning values

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.
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.
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.

Instructor Notes:

Add instructor notes here.



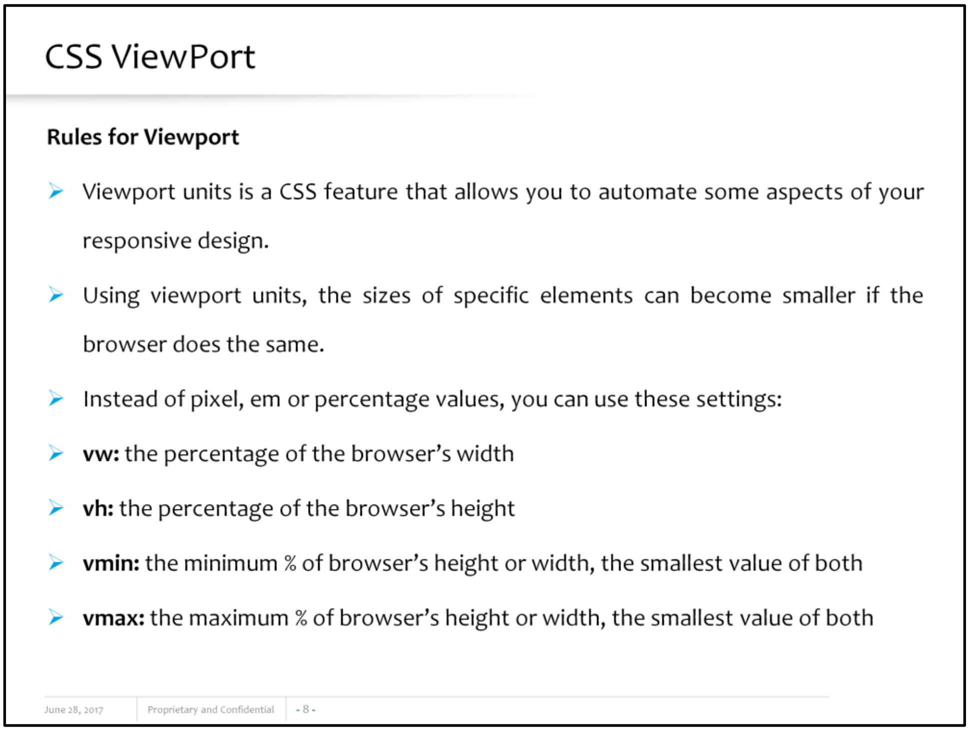
Demo

- Demo1_ViewPort.html
- Demo2_ViewPort.html

June 28, 2017 | Proprietary and Confidential | - 7 -

Instructor Notes:

Add instructor notes here.



CSS ViewPort

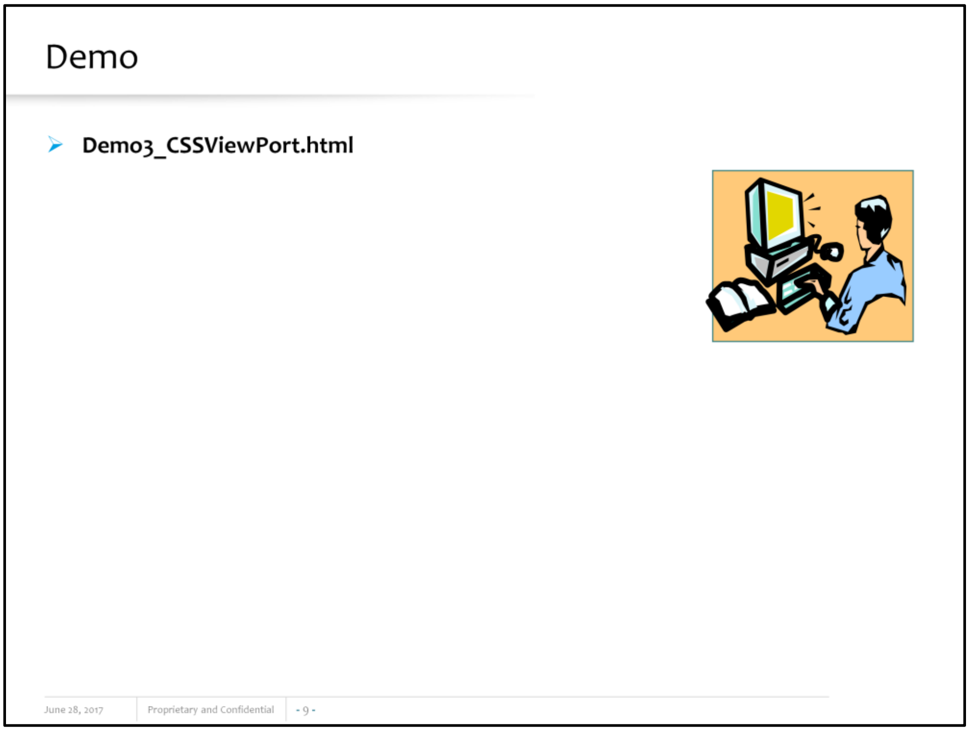
Rules for Viewport

- Viewport units is a CSS feature that allows you to automate some aspects of your responsive design.
- Using viewport units, the sizes of specific elements can become smaller if the browser does the same.
- Instead of pixel, em or percentage values, you can use these settings:
- **vw:** the percentage of the browser's width
- **vh:** the percentage of the browser's height
- **vmin:** the minimum % of browser's height or width, the smallest value of both
- **vmax:** the maximum % of browser's height or width, the smallest value of both

June 28, 2017 | Proprietary and Confidential | - 8 -

Instructor Notes:

Add instructor notes here.

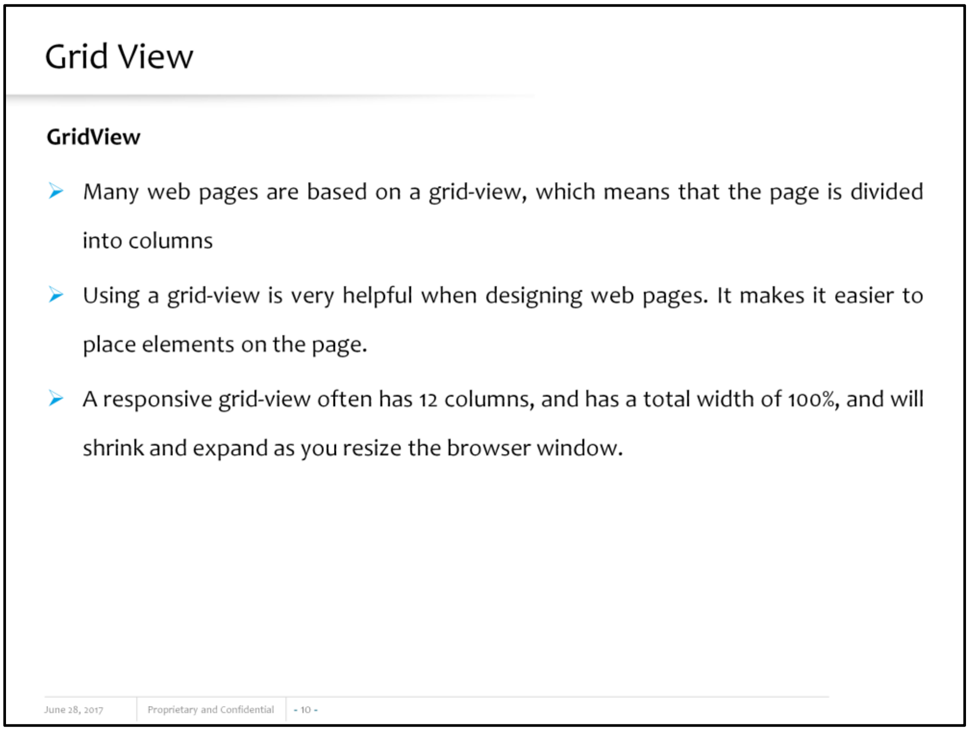


Demo

- **Demo3_CSSViewPort.html**

Instructor Notes:

Add instructor notes here.



Grid View

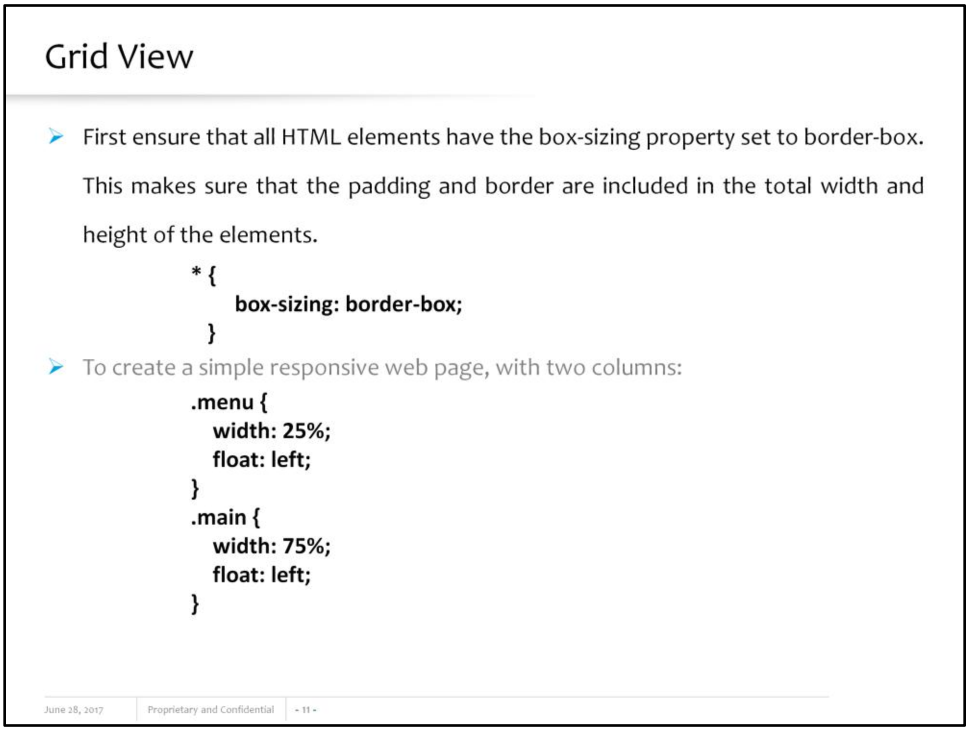
GridView

- 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.

June 28, 2017 | Proprietary and Confidential | - 10 -

Instructor Notes:

Add instructor notes here.



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.

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

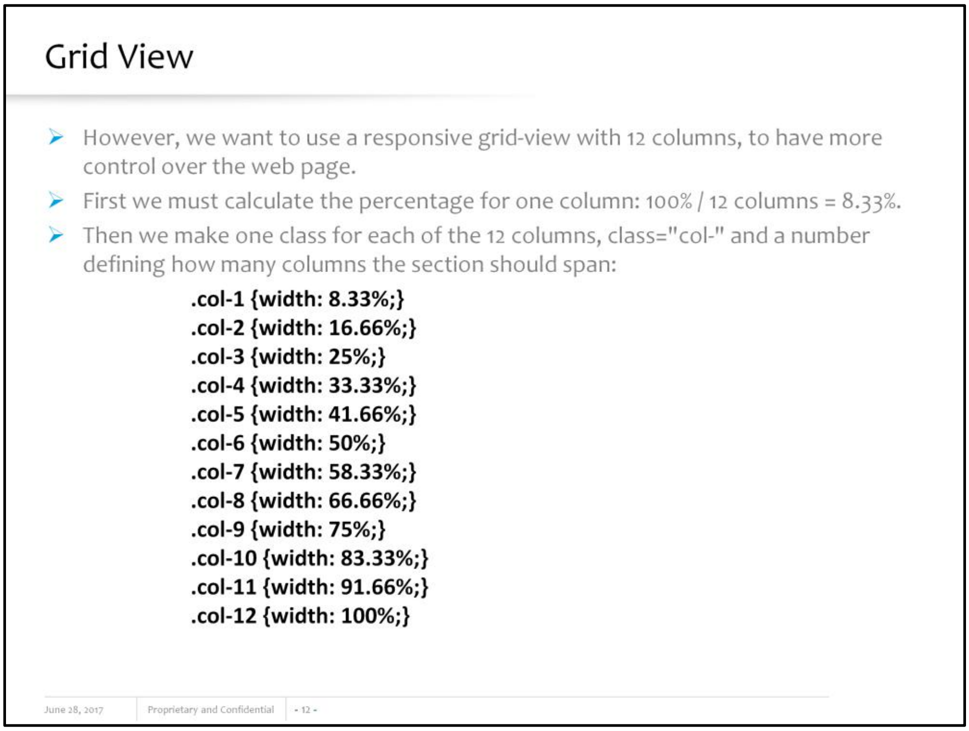
- To create a simple responsive web page, with two columns:

```
.menu {
   width: 25%;
   float: left;
}
.main {
   width: 75%;
   float: left;
}
```

June 28, 2017 | Proprietary and Confidential | - 11 -

Instructor Notes:

Add instructor notes here.



Grid View

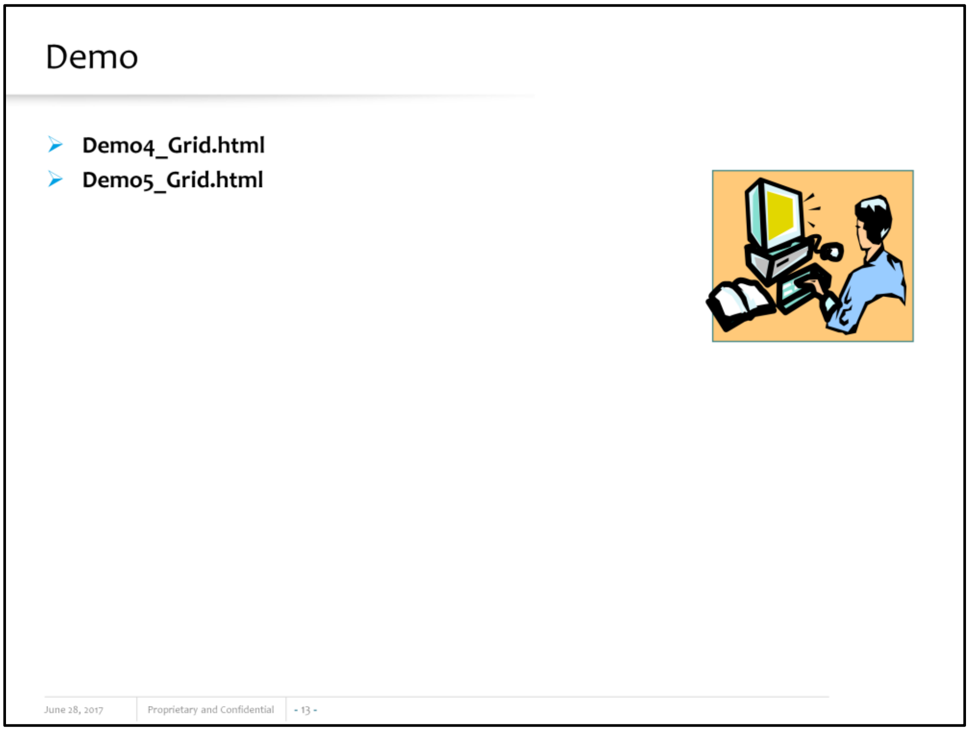
- However, we want to use a responsive grid-view with 12 columns, to have more control over the web page.
- First we must calculate the percentage for one column: 100% / 12 columns = 8.33%.
- Then we make one class for each of the 12 columns, class="col-" and a number defining how many columns the section should span:

```
.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%;}
```

June 28, 2017 | Proprietary and Confidential | - 12 -

Instructor Notes:

Add instructor notes here.



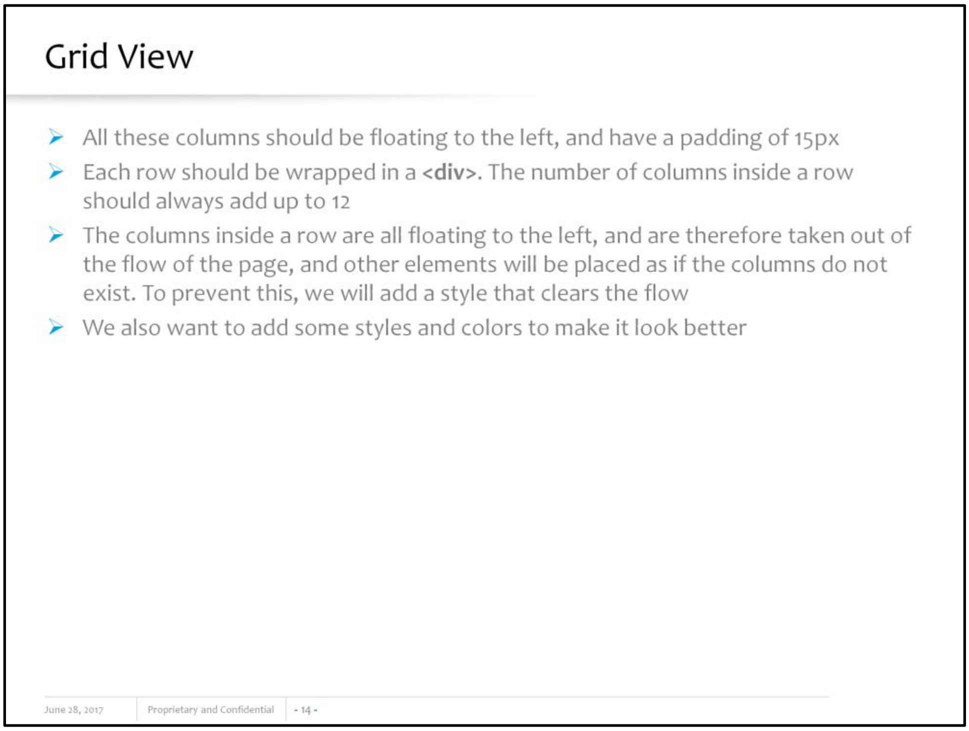
Demo

- Demo4_Grid.html
- Demo5_Grid.html

June 28, 2017 | Proprietary and Confidential | - 13 -

Instructor Notes:

Add instructor notes here.

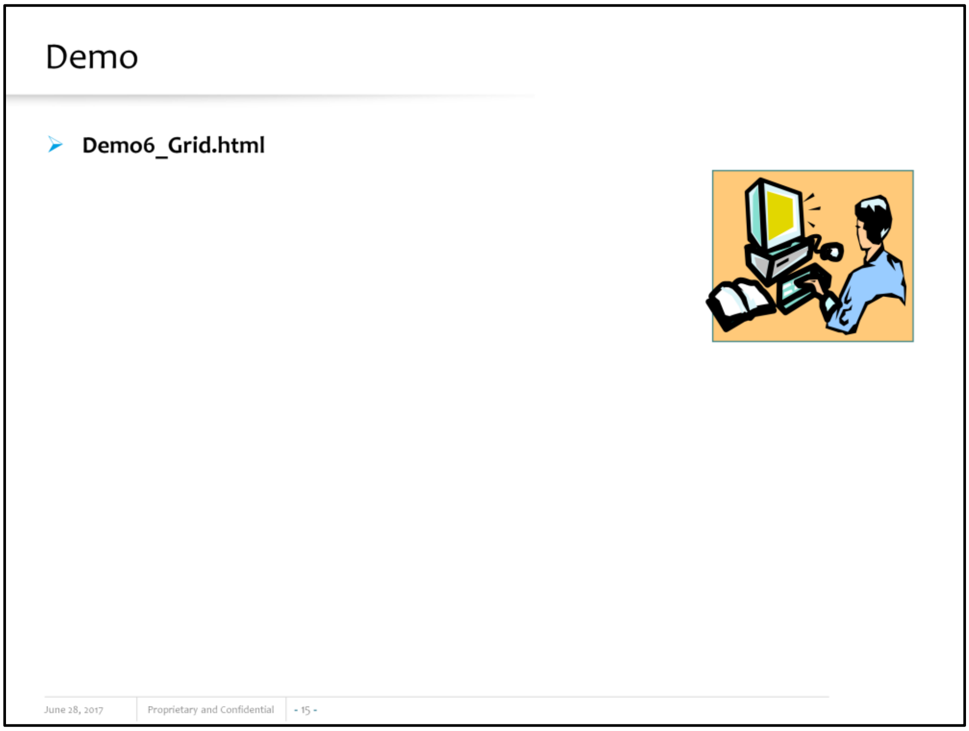


Grid View

- All these columns should be floating to the left, and have a padding of 15px
- Each row should be wrapped in a **<div>**. The number of columns inside a row should always add up to 12
- 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
- We also want to add some styles and colors to make it look better

Instructor Notes:

Add instructor notes here.



Demo

- Demo6_Grid.html

June 28, 2017 | Proprietary and Confidential | - 15 -