

A · P · U
ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

CT053-3-1

Fundamentals of Web Design and Development

Responsive Web Design

What is Responsive Web Design?

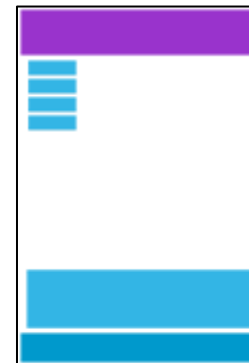
- Responsive web design makes your web page look good on all devices.
- Responsive web design uses only HTML and CSS.
- Responsive web design is not a program or a JavaScript.

Designing For The Best Experience For All Users

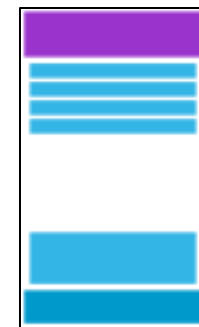
- Web pages can be viewed using many different devices: desktops, tablets, and phones. Your web page should look good, and be easy to use, regardless of the device.
- Web pages should not leave out information to fit smaller devices, but rather adapt its content to fit any device:



Desktop

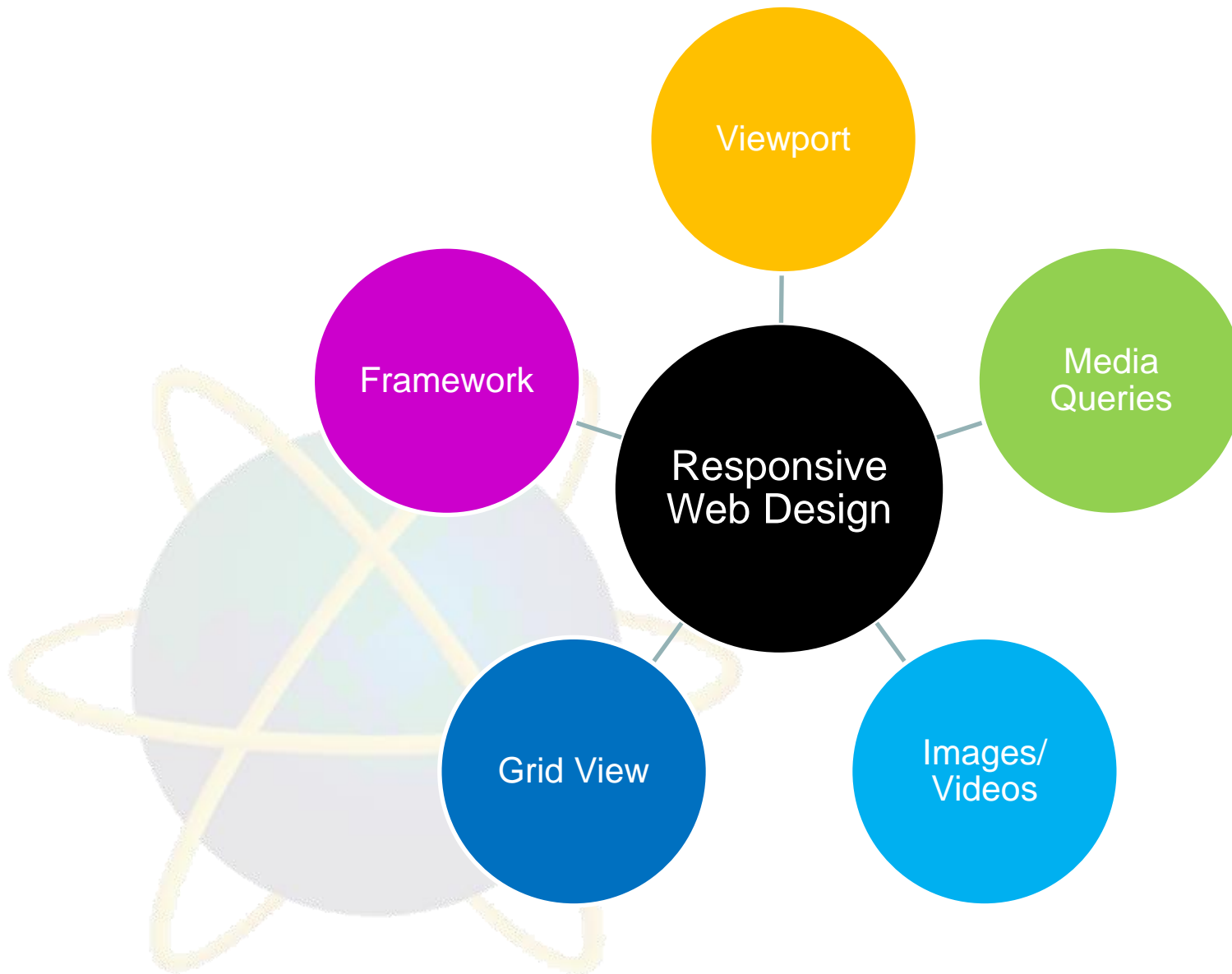


Tablet



Phone

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.



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.

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">
```

Setting The Viewport

```
<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.

With Viewport vs. No Viewport

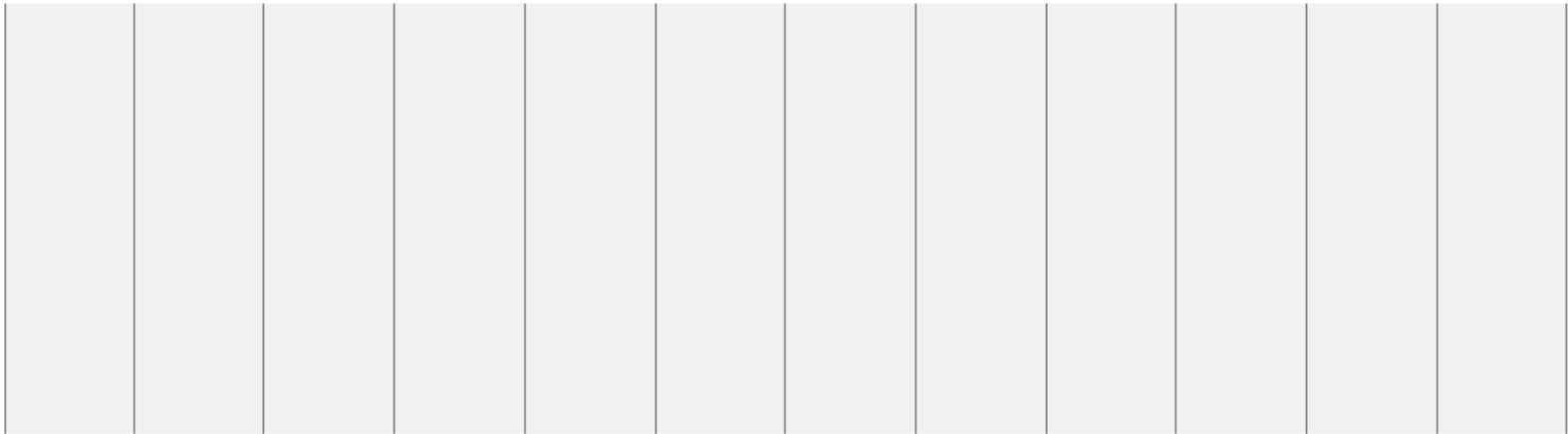


Size Content to The Viewport

- Users are used to scroll websites vertically on both desktop and mobile devices - but not horizontally - **results in a poor user experience.**
- Some additional rules to follow:
 - **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.

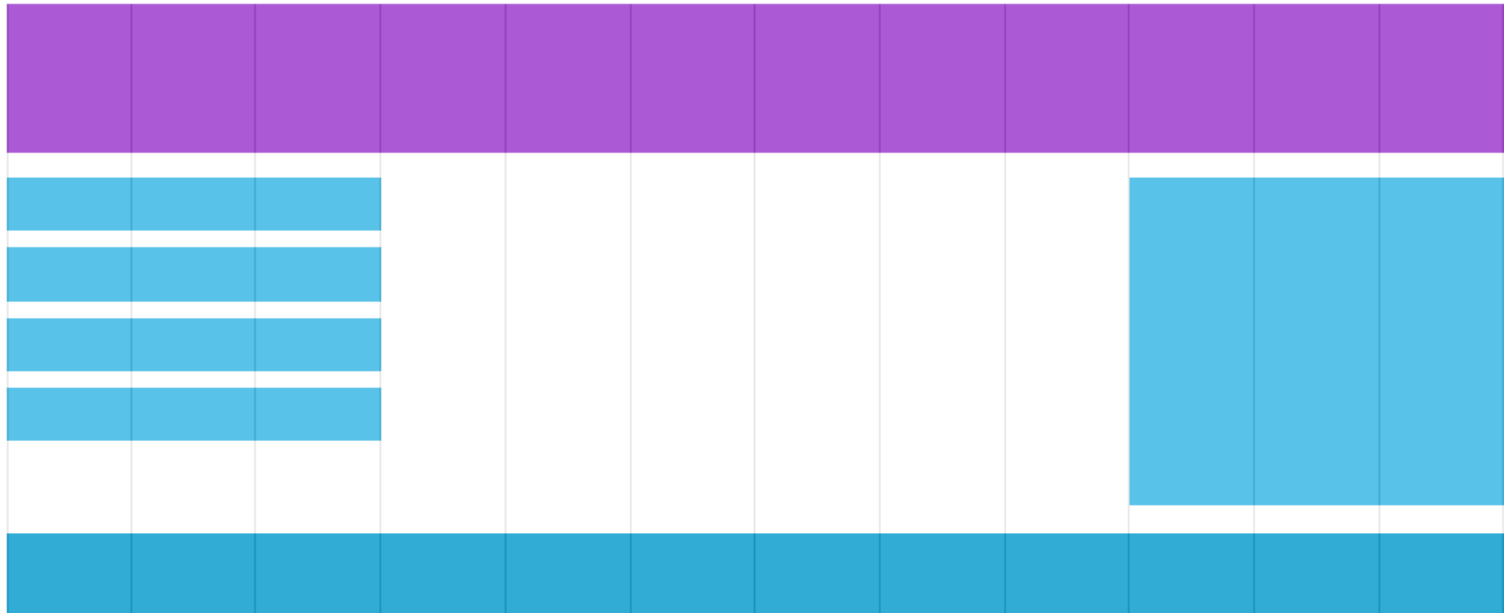
Grid View

- Grid-view means the page is divided into columns:



Grid View

- 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.

Grid View

First we must calculate the percentage for one column: $100\% / 12 \text{ columns} = 8.33\%$.

Then we make one class for each of the 12 columns, `class="col-"` and a number defining how many columns the section should span:

25%

75%

```
.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%;}
```

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

Media Query

- 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.
- E.g.

```
@media only screen and (max-width: 500px) {  
  body {  
    background-color: lightblue;  
  }  
}
```

If the browser window is smaller than 500px, the background color will change to lightblue:

Media Query

- Add a Breakpoint where certain parts of the design will behave differently on each side of the breakpoint.

```
/* For mobile phones: */  
selector-of-your-choice {  
  . . .  
}  
/* For tablets: */  
@media only screen and (min-width: 600px) {  
  . . .  
}  
/* For desktop: */  
@media only screen and (min-width: 768px) {  
  . . .  
}
```

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).

Instead of changing styles when the width gets *smaller* than 768px, we should change the design when the width gets *larger* than 768px.

Images/ videos

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

If the **width** property is set to 100%, the image will be responsive and scale up and down:

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

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.

Frameworks

- A framework is a standardized set of concepts, practices and criteria for dealing with a common type of problem, which can be used as a reference to help us approach and resolve new problems of a similar nature.
- There are many existing CSS Frameworks that offer Responsive Design.
- They are free, and easy to use.

Frameworks

PRESENTATION LAYER / FRONT-END

User interfaces

Languages: HTML, CSS, Javascript...

Frameworks: Bootstrap, Foundation 3, Grids Systems...

APPLICATION LAYER / BACK-END

Logic and operation of the website

Languages: PHP, PYTHON, RUBY, JAVA...

Frameworks: Symfony, Django, Ruby On Rails, Spring...

DATA LAYER

Data persistence on databases

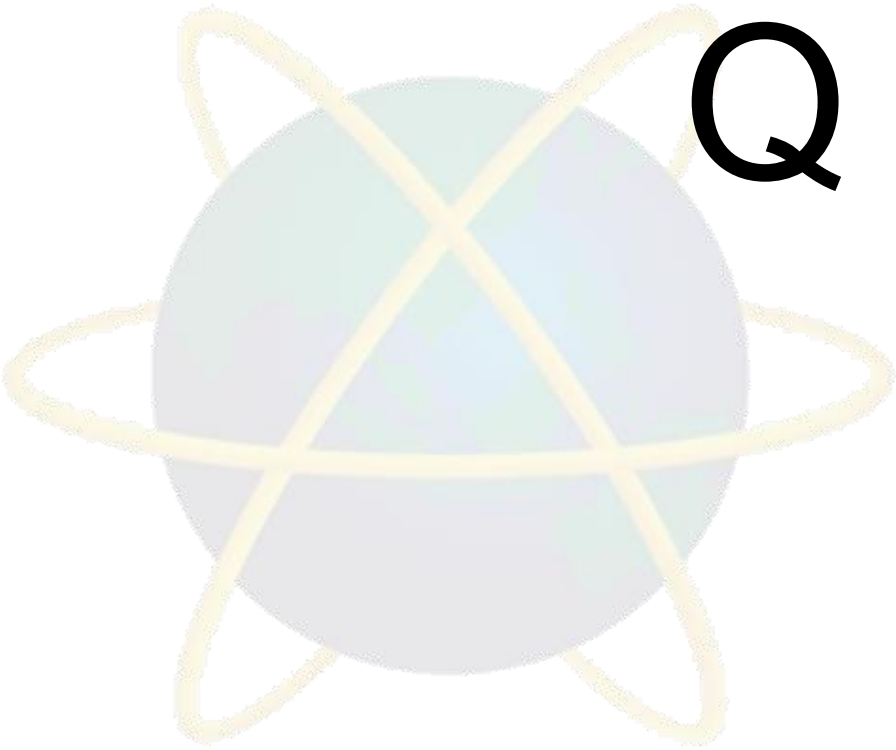
Technologies: MySQL, PostgreSQL, NoSQL...

Advantages and disadvantages of using frameworks

- Advantages
 - Speeds up the mock-up process
 - Clean and tidy code
 - Solutions to common CSS problems
 - Browser compatibility
 - Learn good practices
 - Having a single procedure to resolve common problems makes maintaining various projects more straightforward.
 - Helpful in collaborative work
- Disadvantages
 - Mixes content and presentation
 - Unused code leftover (especially when you want to customize the web design)
 - Slower learning curve – not good for beginner
 - You don't learn to do it yourself

To use or not to use RWD framework

- Is it advisable to use a framework?
 - Not necessarily. The developer must take the final decision on whether or not to use a framework.
 - Frameworks are a resource that can be extremely useful for many people, but that doesn't mean they are necessarily useful for you.



Q & A