

# Responsive Web Design



## Web Technology

**Asst. Prof. Manop Phankokkruad, Ph.D.**

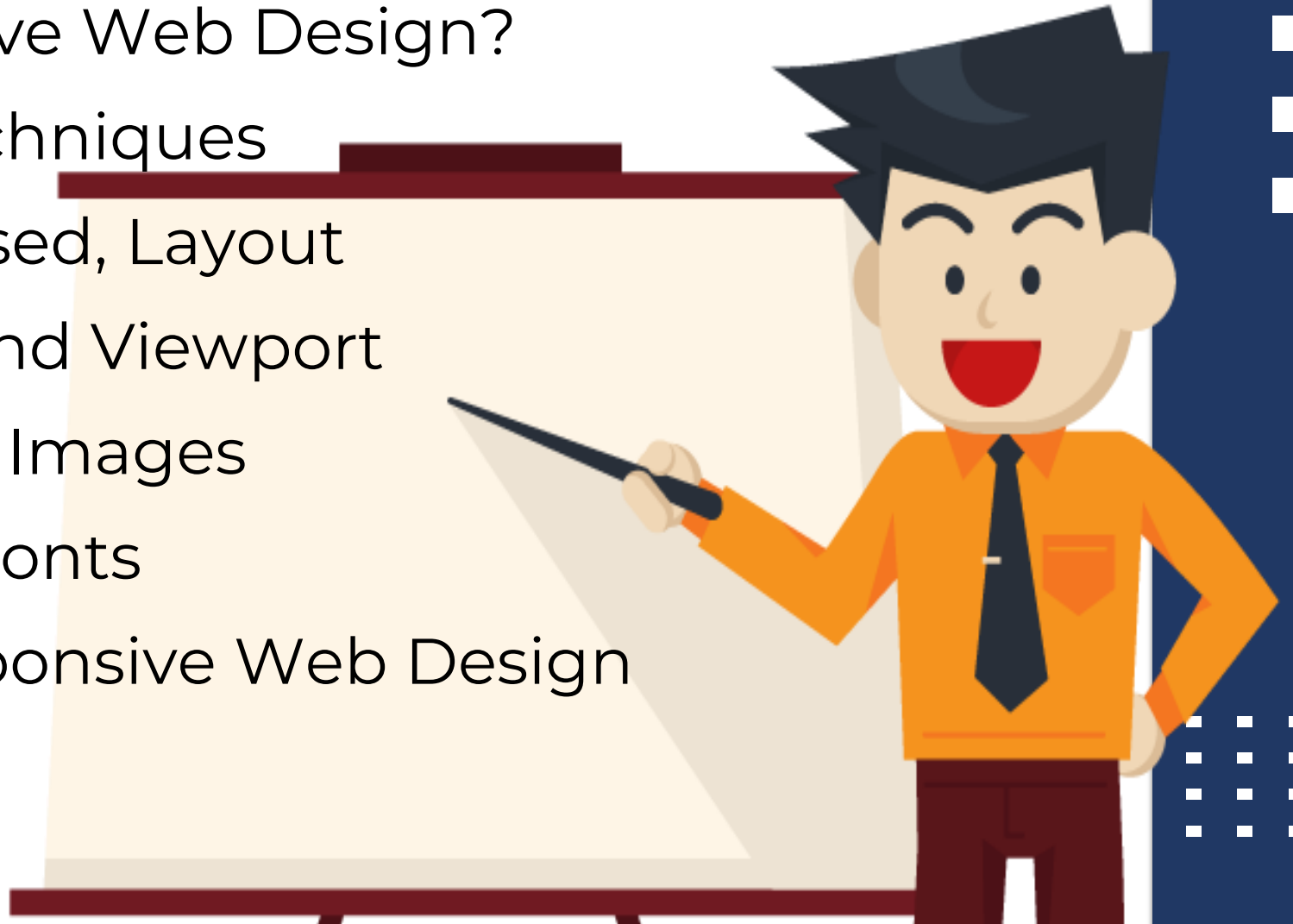
Faculty of Information Technology

King Mongkut's Institute of Technology Ladkrabang



# Outline

- ❑ What is Responsive Web Design?
- ❑ Fundamental Techniques
  - Flexible, grid-based, Layout
  - Media Queries and Viewport
  - Flexible Media & Images
- ❑ Scaling & Sizing Fonts
- ❑ Bootstrap in Responsive Web Design



# What is Responsive Web Design?



Responsive Web Design is a process of designing and building websites to provide better accessibility and optimal viewing experience to the user by optimizing it for different devices.



With the growing trend of smart phones and tablets, it has become almost unavoidable to ignore the optimization of sites for mobile devices. Responsive web design is a preferable alternative and an efficient way to target a wide range of devices with much less efforts.



# What is Responsive Web Design?

- ❑ Website design approach that aims to optimize the viewing experience across a wide range of devices.
- ❑ One website for all devices: desktop, tablets, smartphones.
- ❑ Reformats the page layout per screen resolution.
- ❑ Ideal for sites with less complex functionality and high volume of content.



# What is Responsive Web Design?



# Advantages of RWD

- ❑ One single HTML document to be maintained
- ❑ One single CSS file to be maintained
- ❑ The site is easily accessible on any type of device.
- ❑ Users will have a similar experience using the site when they access the site from different devices.
- ❑ Responsive Web is flexible and adaptable
- ❑ Easier than maintaining several website for different devices.

# What is needed for RWD ?

- ❑ An understanding of the various devices and pixel density.
- ❑ How user experience and user needs are different for mobile users.
- ❑ How Web browsers vary in rendering Web pages.
- ❑ HTML expertise.
- ❑ Extensive use of CSS required.

# Screen size is a challenge

- ❑ Many of these devices use touch interfaces and come in a wide variety of screen resolutions and screen sizes.
- ❑ The size of a pixel varies according to the device.
- ❑ Mobile networks can be slower, and most users have fixed data plans.



# Screen size is a challenge

- ❑ Limited memory, processor power and battery.
- ❑ A wide variety of browsers are used on mobile devices.

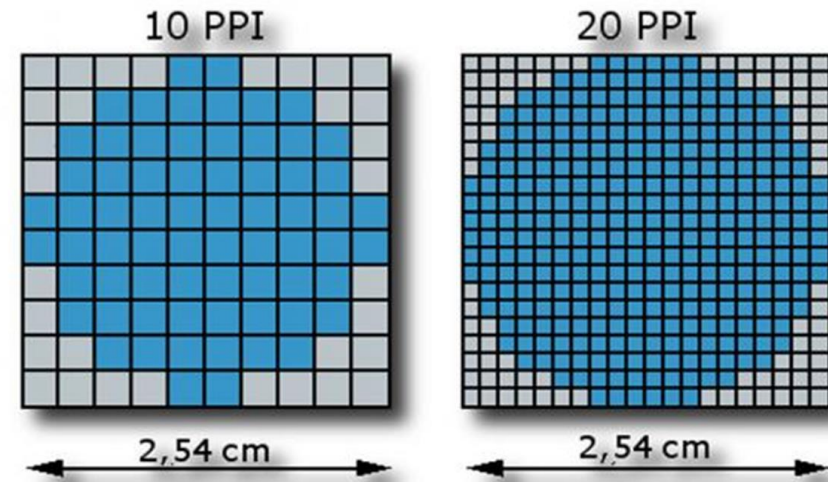


# Variation in number & size of pixels

## Physical size

- A device pixel is the tiniest physical unit in a display.
- **Pixel density** refers to the number of device pixels on a physical surface. It is often measured in pixels per inch (PPI).

Pixel density is a good indication of how clear the device display will be.



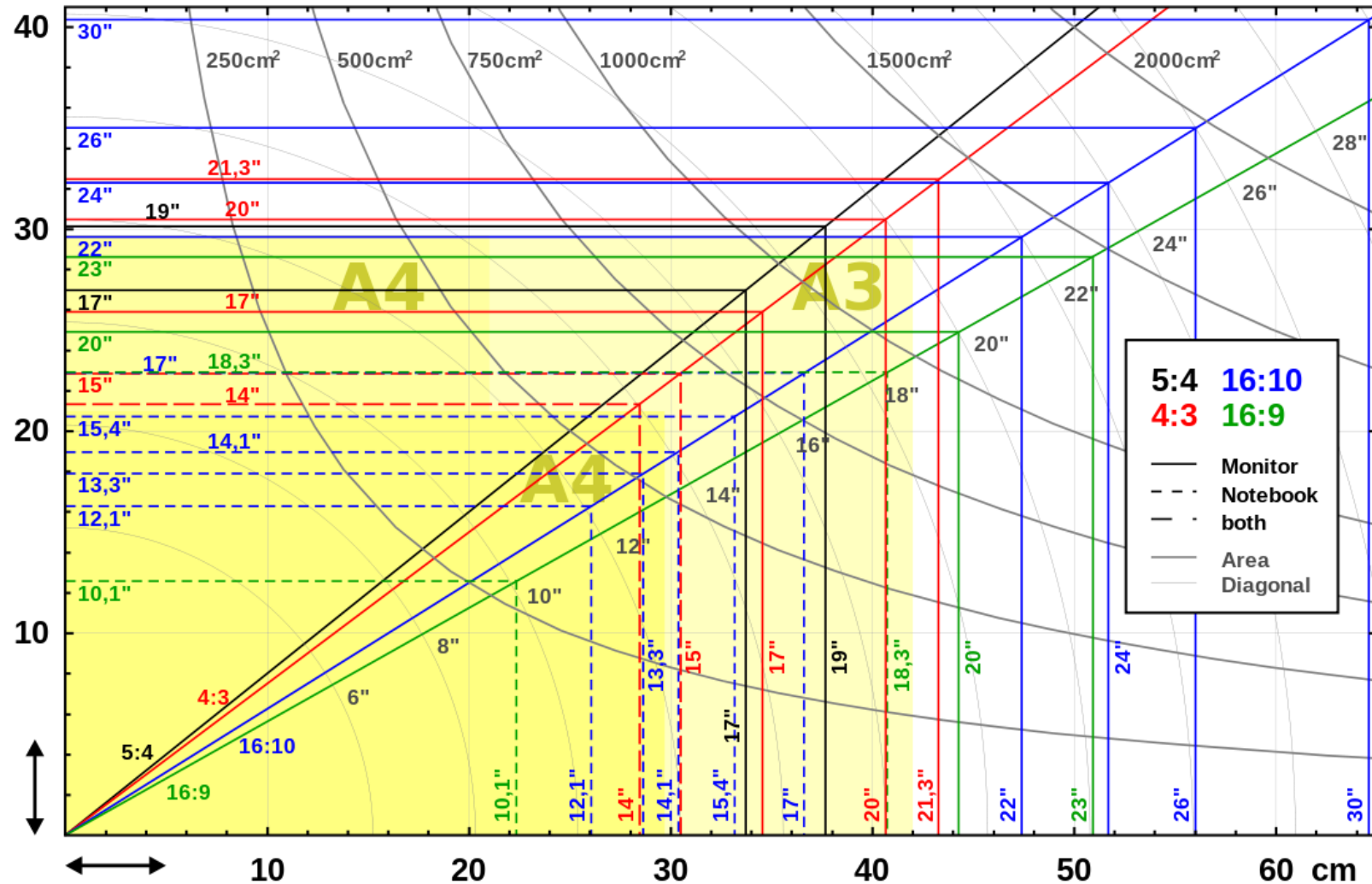
# Variation in number & size of pixels

## Resolution

- Resolution is a simple count of the number of pixels across the entire width and height of a device.
- The Samsung Galaxy Note 20 resolution is 2400x1080. Compare that to the Samsung Galaxy Note II which is 720x1280.

# Variation in number & size of pixels

## LCD screen sizes



# Fundamental Techniques

There are three parts in RWD:

## 1. Flexible, grid-based layouts

A grid that is flexible, which automatically resizes itself according to the size of the window.

## 2. Media query and Viewport

Media queries allow the browser to use different styling. This functionality is used to modify the layout at specific widths in the browser.

## 3. Flexible media & images

Flexible images allow the browser to scale images depending on the size of the browser.

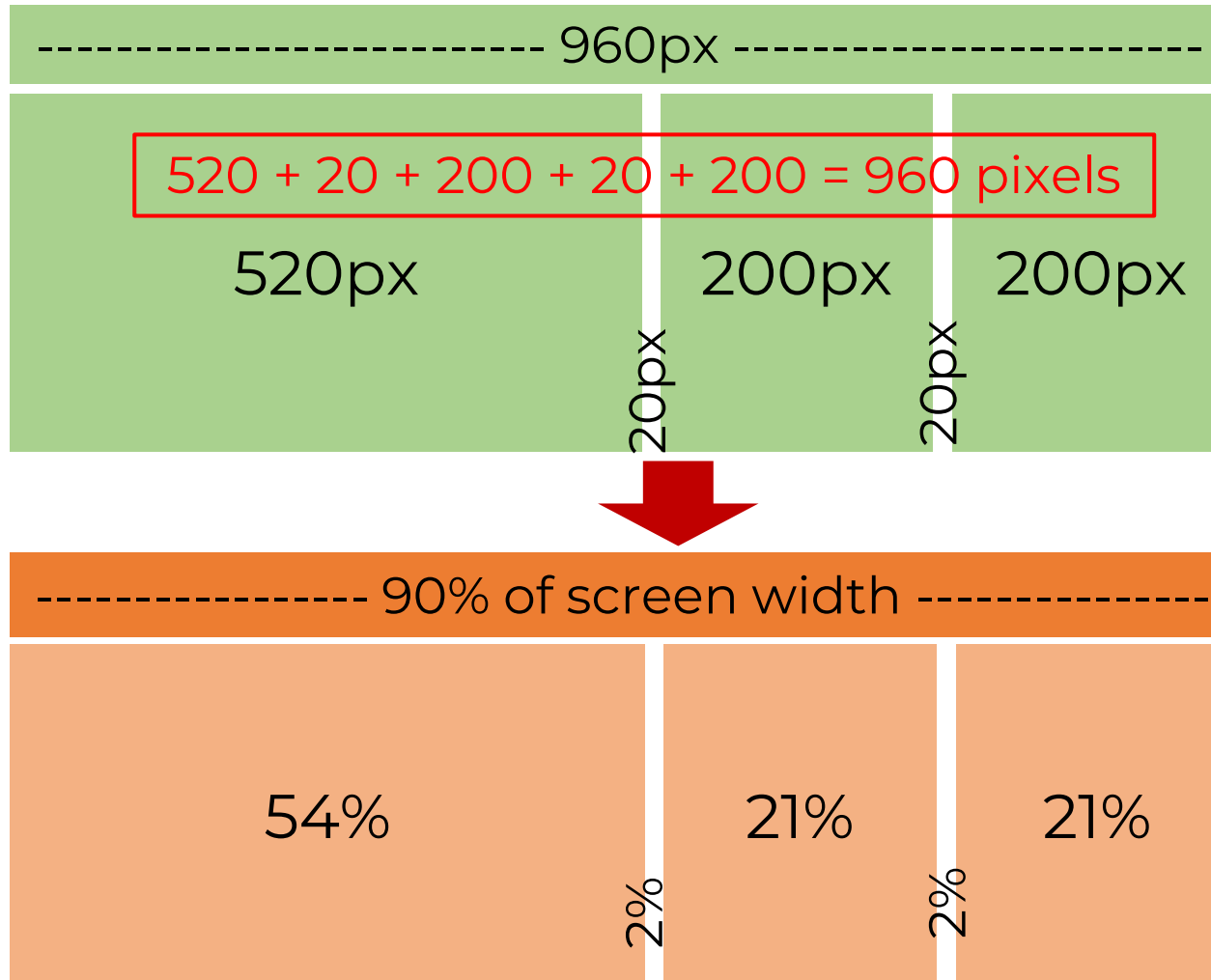
# 1. Flexible, grid-based, Layout

A liquid layout (*sometimes called fluid layout*) uses relative units instead of fixed units. Typically a liquid layout will use percentages instead of pixels, but any relative unit of measurement will work. Idea behind liquid layout, it's more carefully designed in terms of proportion to use percentage.

Proportion of each page element is the target element divided by the context.

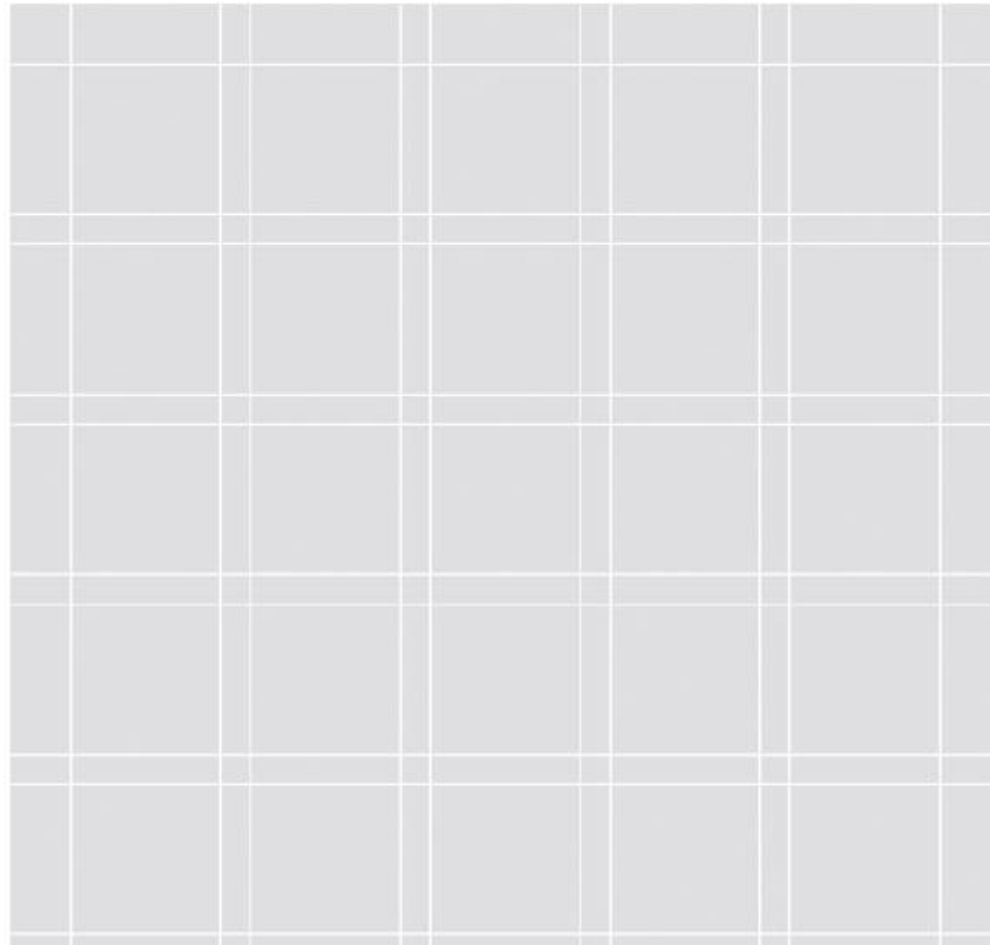
# 1. Flexible, grid-based, Layout

**Figure** : use percentages instead of pixels



# Creating a Flexible Grid

**Figure** : Topographic grid





# Creating a Flexible Grid

## Example

```
<div class="page">
<div class="blog section">
<h1 class="lede">Recently in The Bot Blog</h1>
<div class="main">
...
</div><!-- /end .main -->
<div class="other">
...
</div><!-- /end .other -->
</div><!-- /end .blog.section -->
</div><!-- /end .page -->
```

# 1. Flexible, grid-based, Layout

## **Pros :**

- Can be more user-friendly, because it adjusts to the user's set up.
- The amount of extra white space is similar between all browsers and screen resolutions, which can be more visually appealing.
- If designed well, a fluid layout can eliminate horizontal scroll bars in smaller screen resolutions.

# 1. Flexible, grid-based, Layout

## **Cons :**

- Less control over what the user sees
- May overlook problems because the layout looks fine on their specific screen resolution.
- Images, video and other types of content with set widths may need to be set at multiple widths to accommodate different screen resolutions.
- With incredibly large screen resolutions, a lack of content may create excess white space that can diminish aesthetic appeal.

# 2: Media Query and Viewport

## What is a Media Query?

Media query is a CSS technique introduced in CSS3. The main purpose of a media query is to apply different CSS rules in order to obtain different layouts, depending on the width of the display window afforded to your content.

## What is 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.

## 2: Media Query and Viewport

Media query is the backbone of RWD.

- ❑ The values in the media queries are expressed in percentages and not pixels.
- ❑ Media queries provide the ability to
  - Specify different styles for individual browser device circumstances
  - Specify the width of the viewport or device orientation
- ❑ Using Media queries in the CSS file to change the styling of the HTML elements is based on **certain “breakpoints”**.

# Coding Media Query

- ❑ The following code will display the font-size at 100% if the width is at least 1024 px

```
@media screen and (min-width: 1024px) {  
  body { font-size: 100%; }  
}
```

- ❑ The following code tests the orientation and the device-width

```
@media screen and (min-device-width: 480px)  
and (orientation: landscape) {  
  body { font-size: 100%; }  
}
```

# Coding Media Query (cont.)

- ❑ The following code renders a page that the body background color will change to blue only between 500px and 700px.

```
@media screen (min-width:500px) and (Max-width:700px) {  
  body {background: blue;}  
}
```

## 2: Media Query and Viewport

### ❑ Viewport meta tag:

- Tells the browser how to behave when rendering the page – you tell the browser how big the viewport will be.
- Use the viewport meta tag in the **<head>** section.
- If we are using RWD, it's good to have the **meta tag viewport** as

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



# Coding Meta Viewport tags

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

## ❑ **width=device-width:**

- The page adapts to the device's width
- Syncs with the device's width

## ❑ **initial-scale=1:**

- Make the initial scale at 100%
- When the viewport is larger than the screen width, the scale factor will shrink down to fit the width within the viewport.

# 3: Flexible Media & Images

Using media queries, designers are able to:

- ❑ Extend the media declarations to include various media properties, based on device being used. Such as **screen size, orientation, and color**.
- ❑ write a rule that prevents images from exceeding the width of their container

# 3: Flexible Media & Images

## Example

```
<div class="figure">
<p>
<b src="robot.jpg" alt="" />
<b class="figcaption">the robot walks</b>
</p>
</div>
```

```
.figure {
float: right;
margin-bottom: 0.5em;
margin-left: 2.55319149%;
width: 49.14893617%;
}
```

# Scaling & Sizing Fonts

- ❑ The font-size property sets the size of the text.
- ❑ Being able to manage the text size is important in web design.
- ❑ Always use the proper HTML tags, like **<h1>** - **<h6>** for headings and **<p>** for paragraphs.
- ❑ The font-size value can be an **absolute or relative**.
- ❑ If you do not specify a font size, the default size for normal text, like paragraphs, is 16px.

# Fonts : Scale & Size

## ❑ CSS Font size

- Setting the text size with pixels gives you full control over the text size.
- The text can be resized in all browsers using the zoom tool.

## ❑ CSS Font size units

- Percent(%)
- px
- em
- rem
- vh and vw

# Fonts : Scale & Size

## **em**

- ❑ The **em** is a scalable unit that is used in web document media.
- ❑ An **em** is equal to the current font-size, for instance, if the font-size of the document is 16px, 1em is equal to 16px. Ems are scalable in nature, so 2em would equal 32px, .5em would equal 8px, etc.
- ❑ **em** is becoming increasingly popular in web documents due to scalability and their mobile-device-friendly nature.

# Fonts : Scale & Size

## **Pixels (px)**

- ☐ Pixels are fixed-size units that are used in screen media.
- ☐ One pixel is equal to one dot on the computer screen.
- ☐ Many web designers use pixel units in web documents in order to produce a pixel-perfect representation of their site as it is rendered in the browser.
- ☐ One problem with the pixel unit is that it does not scale upward for visually-impaired readers or downward to fit mobile devices.

# Fonts : Scale & Size

## Points (pt)

- ❑ Points are traditionally used in print media (anything that is to be printed on paper, etc.).
- ❑ One point is equal to  $1/72$  of an inch. Points are much like pixels, in that they are fixed-size units and cannot scale in size.



# Fonts : Scale & Size

## Percent (%)

- ❑ The percent unit is much like the **em** unit, save for a few fundamental differences.
- ❑ First and foremost, the current font-size is equal to 100% (i.e. 12pt = 100%). While using the percent unit, your text remains fully scalable for mobile devices and for accessibility.

# Font Size with ‘%’ and ‘em’

- ❑ Use a combination of **percent**, **px** and **em**
- ❑ The solution that works in all browsers, is to set a default font-size in percent for the <body> element:

	<code>body { font-size: 100%; }</code>	<code>body { font-size: 120%; }</code>
<code>font-size: 1em</code>	The quick brown fox	The quick brown
<code>font-size: 12pt</code>	The quick brown fox	The quick brown fox
<code>font-size: 16px</code>	The quick brown fox	The quick brown fox
<code>font-size: 100%</code>	The quick brown fox	The quick brown

# More Information

- ❑ HTML Responsive Web Design  
[https://www.w3schools.com/html/html\\_responsive.asp](https://www.w3schools.com/html/html_responsive.asp)
- ❑ Introduction to Bootstrap  
<https://getbootstrap.com/docs/4.0/getting-started/introduction/>
- ❑ Bootstrap 4 Examples  
<https://www.tutorialrepublic.com/twitter-bootstrap-examples.php>