SWEN1005

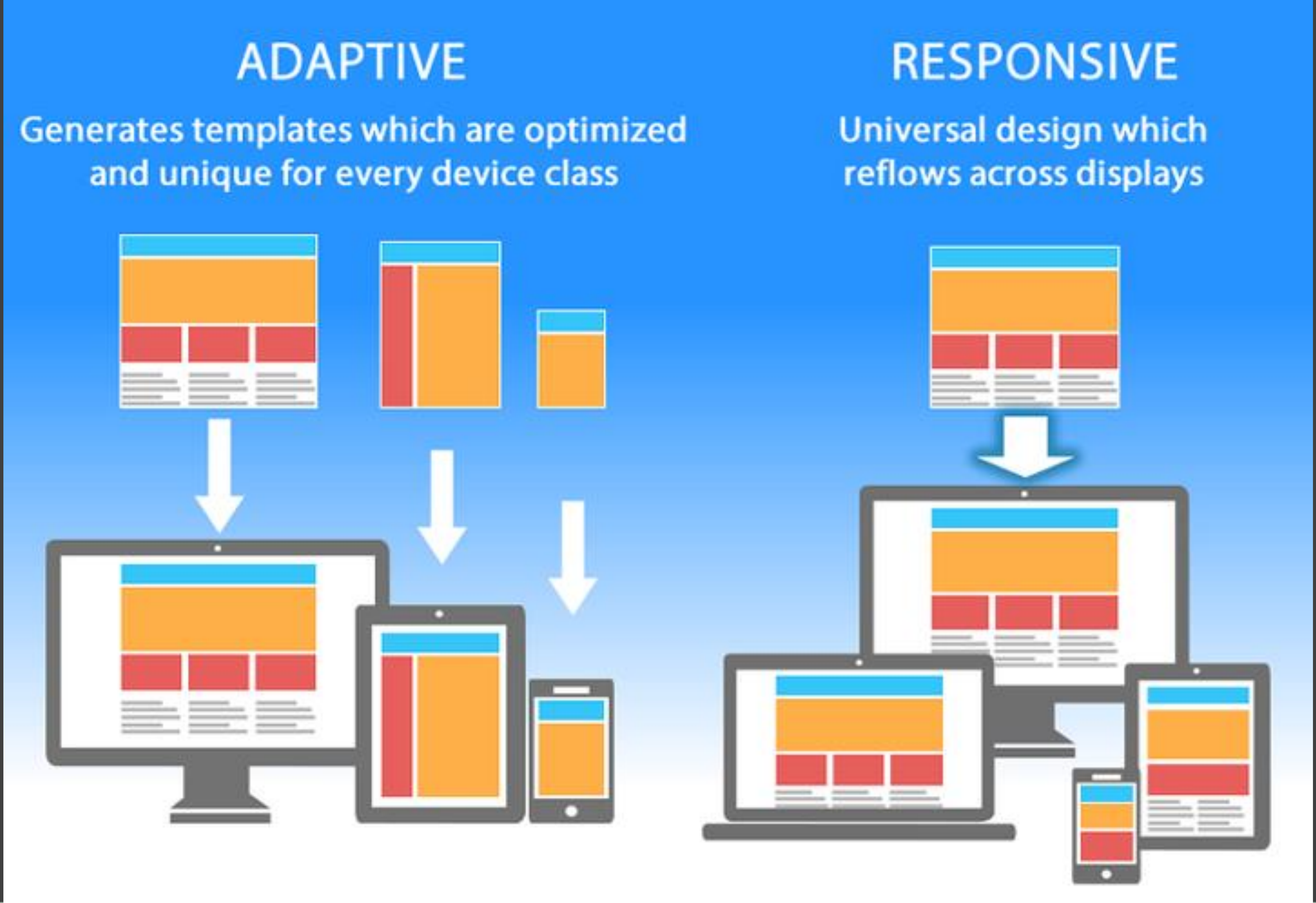
MOBILE WEB PROGRAMMING

Session One Review

MORE ON HTML5 RESPONSIVE SITE

Responsive vs Adaptive

- Similarity between responsive and adaptive sites
  - They both change appearance based on the browser environment they are being viewed on (the most common thing: the browser's width).
- The difference between responsive and adaptive sites
  - Responsive sites have smooth transition
  - Adaptive sites have a snap transition.



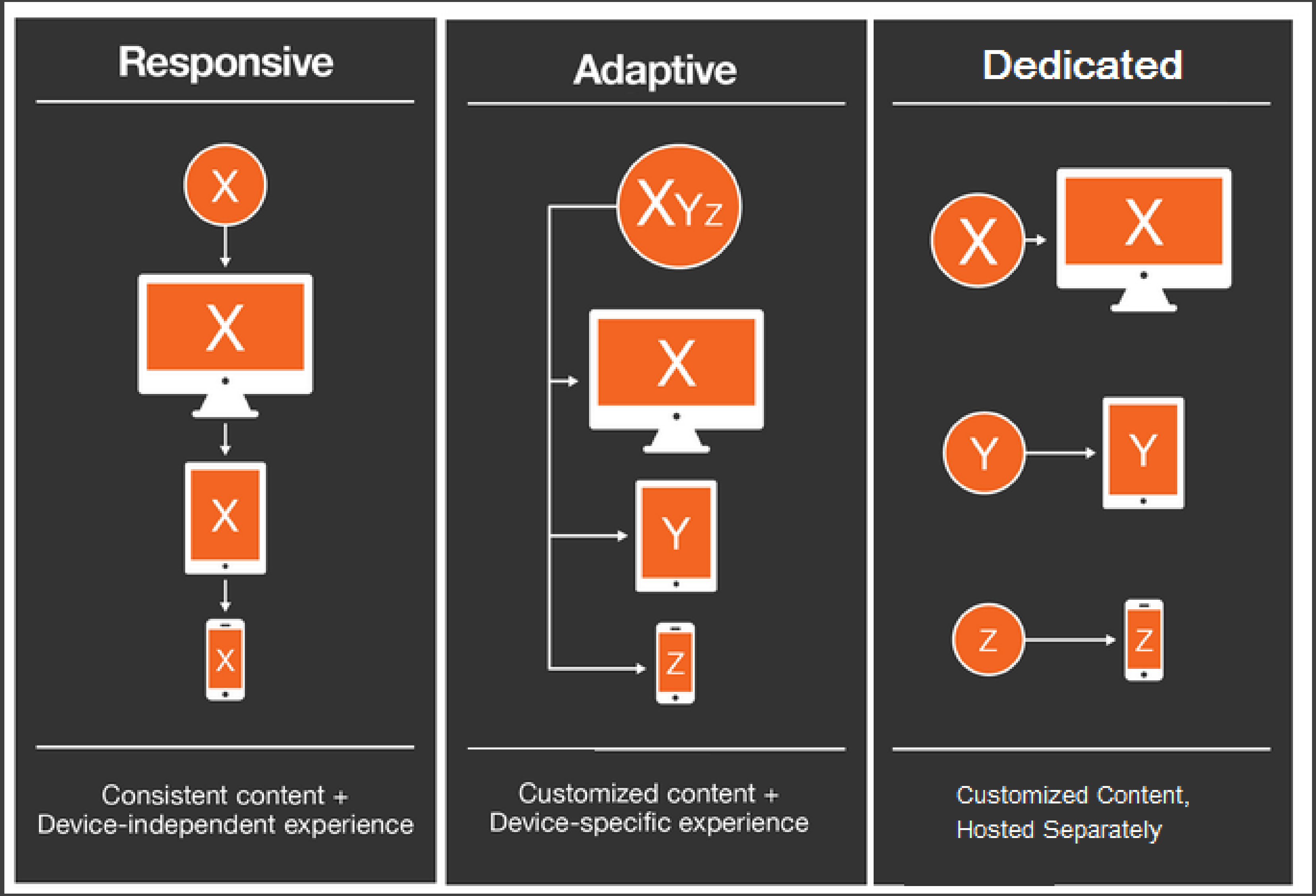
ADAPTIVE

Generates templates which are optimized and unique for every device class

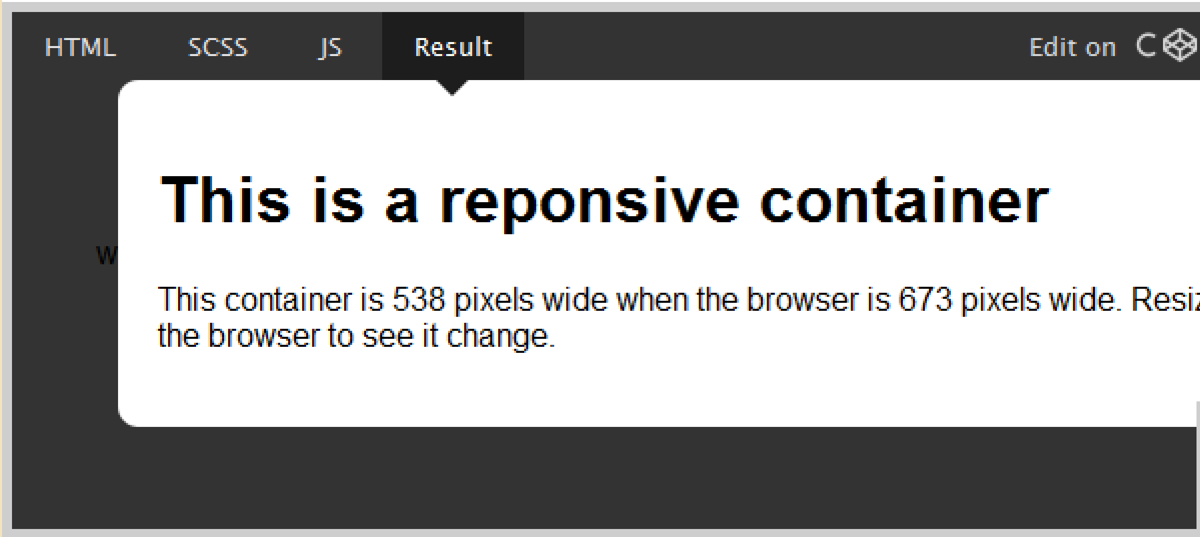
RESPONSIVE

Universal design which reflows across displays

Responsive vs Adaptive (or Dedicated)

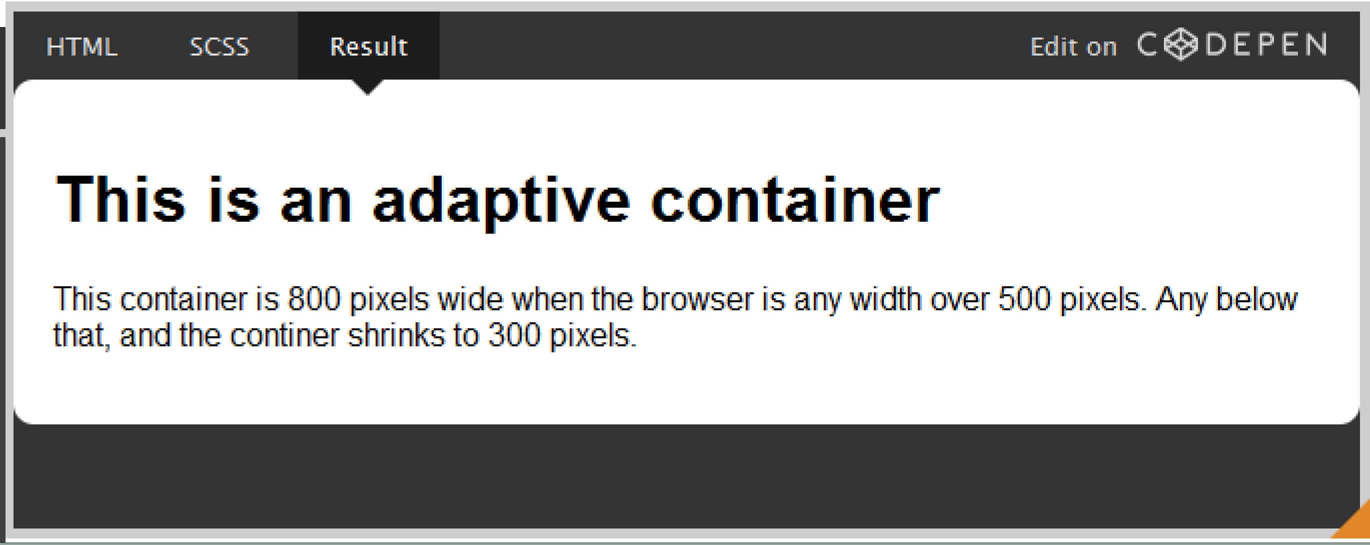


Responsive vs Adaptive



This is a reponsive container

This container is 538 pixels wide when the browser is 673 pixels wide. Resize the browser to see it change.



This is an adaptive container

This container is 800 pixels wide when the browser is any width over 500 pixels. Any below that, and the continer shrinks to 300 pixels.

Responsive Websites

- Responsive websites respond to the size of the browser at any given point.
- Is the browser 300px wide or 30000px wide? It doesn't matter because the layout will respond accordingly. No matter what the browser width may be, the site adjusts its layout (and perhaps functionality) in a way that is optimized to the screen.
- Responsive design is smooth because the layout fluidly adjusts regardless of what device it is viewed on

Adaptive websites

- Adaptive websites adapt to the width of the browser at a specific points.
- In other words, the website is only concerned about the browser being a specific width, at which point it adapts the layout.
- Adaptive design, snaps into place because the page is serving something different because of the browser or device it is viewed on.

Creating a Responsive Website

VIEWPORT

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.
- … when you use CSS and HTML to resize, hide, shrink, enlarge, or move the content to make it look good on any screen.

The Aim

- 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.

How do we start?

```
<!DOCTYPE html>
<html lang="en-us">
   <head>
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
      <style>

         *{

            box-sizing: border-box;

         }

      </style>
   </head>
</html>
```

What is the <meta … > element

- Snippets of text that describe a page's content
- They don't appear on the page itself, but only in the page's code.
- Little content descriptors that help tell search engines what a web page is about.
- Provides metadata about the HTML document
  - page description, keywords, author of the document, last modified…
  - other metadata.

Attributes of the <meta> element

Attribute	Value	Description
charset	character_set	Specifies the character encoding for the HTML document
content	text	Gives the value associated with the http-equiv or name attribute
http-equiv	content-type default-style refresh	Provides an HTTP header for the information/value of the content attribute
name	application-name, author description, generator keywords, viewport	Specifies a name for the metadata

Example and tips

```
<head>
  <meta charset="UTF-8">
  <meta name="description" content="Free Web tutorials">
  <meta name="keywords" content="HTML,CSS,XML,JavaScript">
  <meta name="author" content="John Doe">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
```

- <meta> tags always go inside the <head> element.
- Metadata is always passed as name/value pairs.
- The content attribute MUST be defined if the name or the http-equiv attribute is defined.
  - If none of these are defined, the content attribute CANNOT be defined.

Setting The Viewport

- HTML5 introduced a method to let web designers take control over the viewport, through the <meta> tag.
- The viewport is the user's visible area of a web page. It varies with the device, and will be smaller on a mobile phone than on a computer screen.

Viewport Details

- <meta name="viewport“ content="width=device-width, initial-scale=1.0">
  - The viewport value 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.

Charset

- <meta charset="UTF-8">
- For HTML5, the default character encoding is UTF-8.
- The default character set for HTML5 is UTF-8.
- All HTML 4 processors support UTF-8.
- All HTML5 and XML processors support both UTF-8 and UTF-16.

UTF-8 and UTF-16

Character-set	Description
UTF-8	A character in UTF8 can be from 1 to 4 bytes long. UTF-8 can represent any character in the Unicode standard. UTF-8 is backwards compatible with ASCII. UTF-8 is the preferred encoding for e-mail and web pages
UTF-16	16-bit Unicode Transformation Format is a variable-length character encoding for Unicode, capable of encoding the entire Unicode repertoire. UTF-16 is used in major operating systems and environments, like Microsoft Windows, Java and .NET.

The HTML <head> Element

- The <head> element is a container for metadata (data about data) and is placed between the <html> tag and the <body> tag.
- HTML metadata is data about the HTML document. Metadata is not displayed.

The HTML <head> Element

- Metadata typically define the document title, character set, styles, links, scripts, and other meta information.
  - <title>
  - <style>
  - <meta>
  - <link>
  - <script>
  - <base>

The HTML <title> Element

- The <title> element defines the title of the document, and is required in all HTML documents
- The <title> element:
  - Defines a title in the browser tab
  - Provides a title for the page when it is added to favorites
  - Displays a title for the page in search engine results

The HTML <style> Element

- The <style> tag is used to define style information for an HTML document.
- Inside the <style> element you specify how HTML elements should render in a browser.
- Each HTML document can contain multiple <style> tags.

There are three ways of styling

- External style sheet
- Internal style sheet
- Inline style

The HTML <style> Element

- To link to an external style sheet, use the <link> tag.
- If the "scoped" attribute is not used, each <style> tag must be located in the head section.
- The <style> tag also supports the Global Attributes in HTML.
- The <style> tag also supports the Event Attributes in HTML.

The HTML <style> Attributes

Attribute	Value	Description
media	media_query	Specifies what media/device the media resource is optimized for
scoped	scoped	Specifies that the styles only apply to this element's parent element and that element's child elements
type	text/css	Specifies the media type of the <style> tag

The HTML Global Attributes

Attribute	Description
accesskey	Specifies a shortcut key to activate/focus an element
class	Specifies one or more class names for an element (a style sheet’s class)
contenteditable	Specifies whether the content of an element is editable or not
contextmenu	Specifies an element’s context menu. The context menu appears when a user right-clicks on the element
data-*	Used to store custom data private to the page or application
dir	Specifies the text direction for the content in an element
draggable	Specifies whether an element is draggable or not
dropzone	Specifies if the dragged data is copied, moved, or linked, when dropped

The HTML Global Attributes

Attribute	Description
hidden	Specifies that an element is not yet, or is no longer, relevant
id	Specifies a unique id for an element
lang	Specifies the language of the element's content
spellcheck	Specifies if the element has its spelling and grammar checked or not
style	Specifies an inline CSS style for an element
tabindex	Specifies the tabbing order of an element
title	Specifies extra information about an element
translate	Specifies if the content of an element should be translated or not

The <link> tag

- Is a singleton tag
- The <link> tag defines a link between a document and an external resource.
- The <link> tag is used to link to external style sheets.

Example

-
  ```
  <head>
    <link rel="stylesheet" type="text/css" href="theme.css">
  </head>
  ```

The <link> tag Attributes

Attribute	Value	Description
crossorigin	anonymous use-credentials	How the element handles cross-origin requests
href	URL	Specifies the location of the linked document
hreflang	language_code	How the language of the text in the linked document
media	media_query	What device the linked document will display
sizes	Height x Width, any	The size of the linked resource. Only for rel="icon"
target	_blank, _self, _top, _parent, frame_name	Not supported in HTML5. Specifies where the linked document is to be loaded
type	media_type	Specifies the media type of the linked document

The <link> tag Attributes

Attribute	Value	Description
rel	Alternate, author, dns-prefetch help, icon, license, next, pingback, preconnect, prefetch, preload, prerender, prev, search, stylesheet	Required. Specifies the relationship between the current document and the linked document
rev	reversed relationship	Not supported in HTML5. Specifies the relationship between the linked document and the current document

HTML <script> Tag

- The <script> tag is used to define a client-side script (JavaScript).
- The <script> element either contains scripting statements, or it points to an external script file through the src attribute.
- Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content.

HTML <script> Tag

- If the "src" attribute is present, the <script> element must be empty.
- There is also the <noscript> element for users that have disabled scripts in their browser, or have a browser that doesn't support client-side scripting.
- There are several ways an external script can be executed:
  - If async="async": The script is executed asynchronously with the rest of the page (the script will be executed while the page continues the parsing)
  - If async is not present and defer="defer": The script is executed when the page has finished parsing
  - If neither async or defer is present: The script is fetched and executed immediately, before the browser continues parsing the page

HTML <script> Tag

Example

```
<script>
  document.getElementById("demo").innerHTML = "Hello, World!";
</script>
```

HTML <script> Tag –jQuery reference

Example

```
<script src="https://ajax.googleapis.com/.../jquery.min.js"></script>
<script>
  $(document).ready(function(){
     $("button").click(function(){
        $("p").hide();
     });
  });
</script>
```

HTML <script> Tag

Attribute	Value	Description
async	async	Specifies that the script is executed asynchronously (only for external scripts)
charset	charset	Specifies the character encoding used in an external script file
defer	defer	Specifies that the script is executed when the page has finished parsing (only for external scripts)
src	URL	Specifies the URL of an external script file
type	media_type	Specifies the media type of the script

The HTML <base> tag

- The <base> tag specifies the base URL/target for all relative URLs in a document.
- There can be at most one <base> element in a document, and it must be inside the <head> element.
- Put the <base> tag as the first element inside the <head> element, so that other elements in the head section uses the information from the <base> element.
- If the <base> tag is present, it must have either an href attribute or a target attribute, or both.

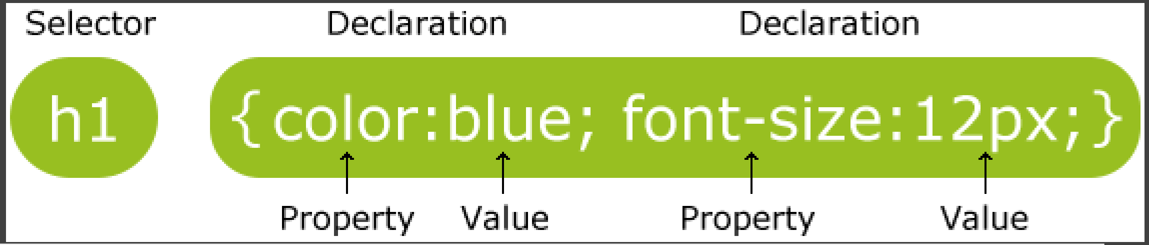
Web Standards

CSS – CASCADING STYLE SHEETS

What is CSS?

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files

CSS Syntax



- A CSS rule-set consists of a selector and a declaration block:
- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.
- A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

CSS Selectors

- CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and more.

Example (class)

```
.flex-item
   {
      background-color: cornflowerblue;
      width: 100px;
      height: 100px; margin: 10px;
   }
```

The element Selector

- The element selector selects elements based on the element name.
- Example

```
body { background-color: lightblue;}
h1 {
  color: navy;
  margin-left: 20px;
}
```

The id Selector

- The id selector uses the id attribute of an HTML element to select a specific element.
- The id of an element should be unique within a page, so the id selector is used to select one unique element!
- To select an element with a specific id, write a hash (#) character, followed by the id of the element.

The id Selector

- Example

```
#para1
 {
    text-align: center;
    color: red;
 }
```

The class Selector

- The class selector selects elements with a specific class attribute.
- To select elements with a specific class, write a period (.) character, followed by the name of the class.
- In the example below, all HTML elements with class="center" will be red and center-aligned:

Example

```
.center {
   text-align: center;
   color: red;
}
```

The class Selector – element specific

- You can also specify that only specific HTML elements should be affected by a class.
- Example

```
p.center
{
      text-align: center;
      color: red;
}
```

Example – Grouping Selectors

```
h1, h2, p

  {
    text-align: center;
    color: red;

  }
```

CSS Comments

- Comments are used to explain the code, and may help when you edit the source code at a later date.
- Comments are ignored by browsers.
- A CSS comment starts with /* and ends with */.

Example

```
p {
    color: red;
    /* This is a single-line comment */
    text-align: center;
}

/* This is
a multi-line
comment */
```

The End

SESSION THREE