ASU W. P. Carey School of Business
Arizona State University

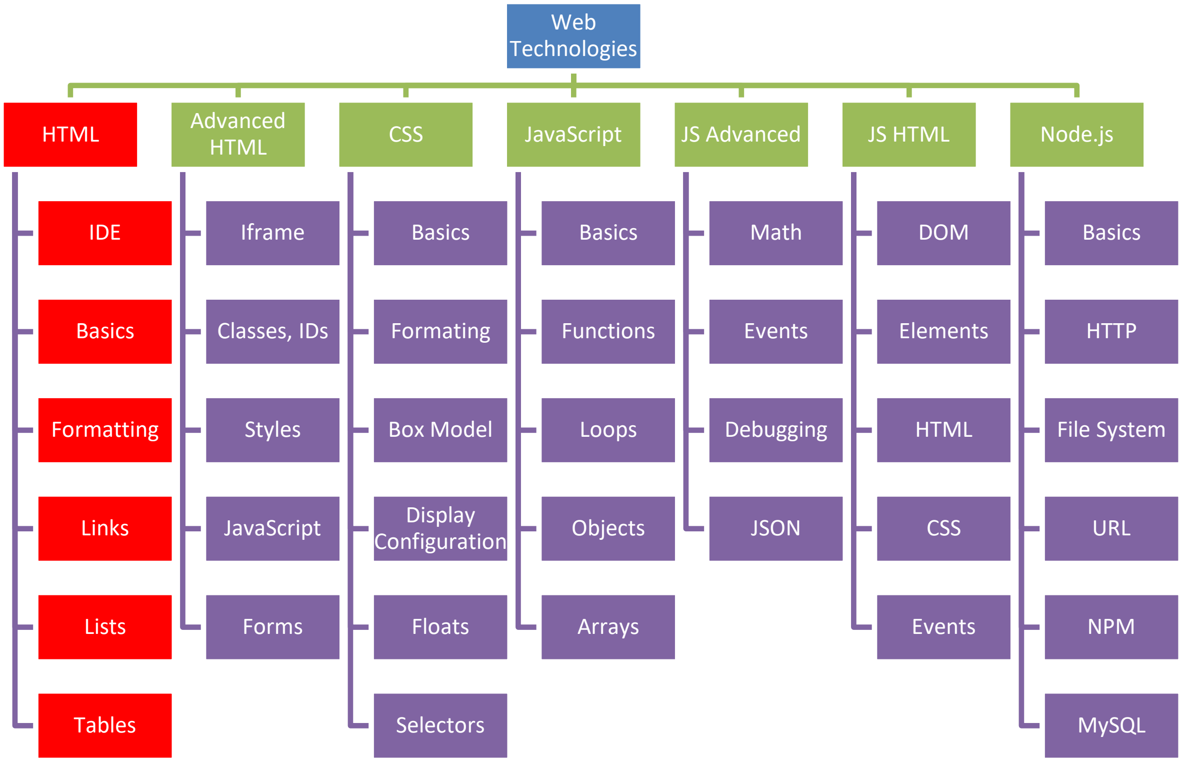
CIS 425 – Enterprise Web Technologies

Assignment: Basic Webpage

Learning Outcomes

1.1. Integrate basic HTML tags to form a complete page
1.2. Implement basic formatting and HTML symbols
1.3. Implement data in a tabular structure
1.4. Utilize HTML5 semantic tags to add meaning to document sections

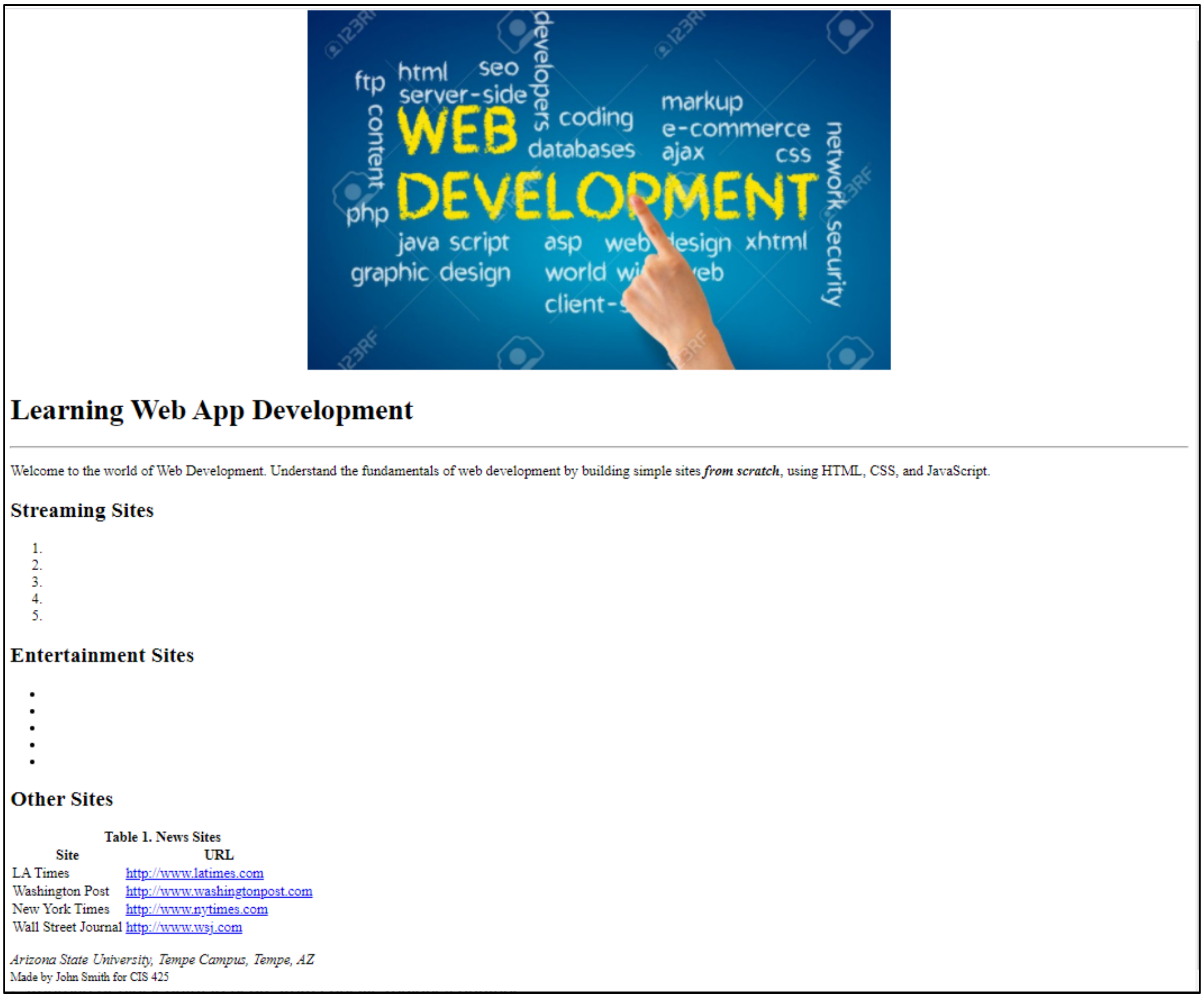
Topic Chart



Overview

Create a webpage using basic HTML elements. Markup text using formatting and image tags.

Sample Output



Learning Web App Development

Welcome to the world of Web Development. Understand the fundamentals of web development by building simple sites ***from scratch***, using HTML, CSS, and JavaScript.

Streaming Sites

1.
2.
3.
4.
5.

Entertainment Sites

•
•
•
•
•

Other Sites

Table 1. News Sites

Site	URL
LA Times	http://www.latimes.com
Washington Post	http://www.washingtonpost.com
New York Times	http://www.nytimes.com
Wall Street Journal	http://www.wsj.com

Arizona State University, Tempe Campus, Tempe, AZ
Made by John Smith for CIS 425

Instructions

1) Create a **new** folder for the assignment. Name the folder asuriteA2, replacing asurite with your own asurite ID, eg. jsmithA2
2) Create a new file and save it as **index.html**
3) Create a skeletal HTML structure. Account for DOCTYPE, HTML, HEAD, BODY, TITLE and charset.
4) Add your name, assignment #, class, and class time as an **HTML comment** at the top of the file.

   e.g., ***John Smith, A2, CIS 235, 9 am***

5) Look at the sample output. Write HTML to create the page. Bring it as close to the sample output as you can.
   - For Streaming Site and Entertainment Sites sections, **add** 5 of the top websites in each category as items in the lists.

6) Keep the following in mind in relation to individual elements.

HTML

- Use **only** HTML for formatting. *We'll format using CSS later!*
  - Do *not* use style attributes.
- Use HTML symbols as appropriate to ensure that symbols are properly displayed across all browsers.

HTML5

- Use semantic HTML5 tags as appropriate
- Your webpage must use `header, figure, main, section, caption, footer` `and address` at minimum. This is not an exhaustive list – you may need more tags!

Images/Tables

- Use relative paths
- Provide an alternate text, title, and captions for elements as applicable.

Links

- Open external links in new windows

7) Go to the W3C validation service at https://validator.w3.org/ and upload your file to test if it follows W3C guidelines. Fix any errors!

Since we are not using CSS, ignore any errors which direct you to use CSS instead of HTML formatting attributes!

Assignment Submission Instructions

- Create a Word document named `asuriteOutput.docx`, replacing asurite with your own ID.
    - On page 1, **insert a screenshot of your webpage** rendering in Chrome. *Hint: Zoom out to get a screenshot of the full page. Make sure the image occupies the full page of the Word document so that it is readable.*
    - On page 2, **insert a screenshot of your W3C validation report.** If you get an HTML formatting error, make sure it is visible.

  Place the Word docx file in your project folder.

- Your project folder should have:

  1) the HTML file
  2) all related image files
  3) the Word docx file showing validation and rendering results.

- **Zip the entire project folder**. Name the folder **asuriteA2**, replacing asurite with your own ASURITE ID, e.g. jsmith7E2

- **Submit the ZIP file to Canvas**.

- **Important Note:** If you use folder paths in any filename with your website, your file will not load properly on any other computer. Points will be deducted if your file does not render properly. *Think about it – should websites work only on the developers' computers or all computers?*