Print-Friendly Page

CPNT 260 - Web Page Construction Fundamentals

Course Description:
This web design course provides the foundation of web site creation using the most recent version of Hypertext Markup Language (HTML) and cascading style sheets to develop consistent web pages that are easy to standardize and modify.

1.5 credits

Time Guidelines:
The standard instructional time for this course is 80 hours.

Effective Year
2018/2019

Course Assessment:

Assignments (x 5)	50%
Final Project	25%
Attendance & Participation	25%
Total	100%

SAIT Policies and Procedures:
For information on the SAIT Grading Scale, please visit policy AC 3.1.1 Grading Progression Procedure: http://www.sait.ca/Documents/About SAIT/Administration/Policies and Procedures/AC.3.1.1 Grading and Progression Procedure.pdf

For information on SAIT Academic Policies, please visit: www.sait.ca/about-sait/administration/policies-and-procedures/academic-student

Required Course Publication(s):
Duckett, J. (2011). *HTML and CSS: Design and Build Websites* (1st ed.). John Wiley and Sons. ISBN: 9781118008188.

Course Learning Outcome(s):
1. Create a simple HMTL document.

Objectives:

1.1 Explain the difference between HTML, CSS, and JavaScript.

1.2 Briefly summarize the history and development of HTML and CSS totheir current incarnation.

1.3 Describe the purpose of domain extensions and how they are used.

1.4 Define a tag, element and attribute.

1.5 Identify depreciated and obsolete elements from previous versions of HTML.

1.6 Define the rules for naming web pages.

1.7 Add a file extension and save a web page.

1.8 View the web page in a browser.

1.9 Validate a web page.

1.10 View the source of a web page.

1.11 Create a web page using appropriate document structure.

1.12 Explain how the viewport attribute makes a web page responsive.

1.13 Use structural elements correctly, and explain how to address browser support and accessibility concerns.

1.14 Add appropriately marked-up text to a web page, including paragraphs and headings.

1.15 Distinguish between block and inline elements.

1.16 Add HTML comments to a web page.

2. Add lists, tables and links to a web page.

Objectives:

2.1 Create a table using accessible HTML.

2.2 Define the lists available in HTML and their semantic purpose.

2.3 Create an ordered, unordered and definition list.

2.4 Distinguish between local and remote links.

2.5 Create two pages in the same location and link them using local links.

2.6 Add remote links to other sites on a web page.

2.7 Add an anchor to jump to another location in a web page.

2.8 Create a simple navigationfor a web site using a list.

3. Insert images into a web page with appropriate markup.

Objectives:

3.1 Identify image file format types that can be used on web pages, and the properties, advantages and disadvantages of each.

3.2 Explain what image resolution is and how it affects what you see on the screen.

3.3 Describe how vector images are created with SVG, and explain their benefits and drawbacks.

3.4 Downloadimages from the web that can be legally used on a website.

3.5 Insert an image into a web page.

3.6 Add accessibility features to the image.

3.7Add a caption to an image.

3.8 Describe how responsive image techniques can be used to improve the performance of a website.

3.9 Explain the appropriate usage of image sprites, icon fonts, SVG icons, favicons, and touch icons.

4. Create a form for a website.

Objectives:

4.1 Define the rules and guidelines for creating a usable and accessible form.

4.2Group related inputs usinglegend and fieldset.

4.3 Add a labelto aninput field on a form.

4.4 Describe when each input type should be used.

4.5Explain why using the correctinput type willhelp with accessibility and mobile usability.

4.6 Use attributes to determine what data will be submitted in the form.

4.7 Create a form that uses the various input types, a text area,select, and submit.

4.8 Explain how to make forms easier for users to complete on a mobile device.

5. Outline the origins and fundamental concepts of Cascading Style Sheets (CSS).

Objectives:

5.1Describe browser support for CSS and explain when vendor prefixes should be used.

5.2Define the purpose and importance of separating structure and data from presentation.

5.3Explain basic CSS syntax and write a simple CSS declaration.

5.4Write CSS in an external style sheet, embedded styles, and inline.

5.5Write a CSS comment.

6. Use CSS qualities of colour,units and typographyto modify the appearance of elements.

Objectives:

6.1 Define the various CSS methods for defining colour.

6.2Explain the difference betweenabsolute andrelative units of measurement in CSS.

6.3Add a border to an element using CSS.

6.4 Use web fonts to style text with various typefaces.

6.5 Use font properties to adjust the size, spacing, and appearance of text.

6.6 Adjust font properties with media queries so that text scales properly on responsive pages.

7. Use CSS selection techniques to gain access to different elements in an HTML document.

Objectives:

7.1 Definetechniques for selecting elements on a page.

7.2Use class and ID attributes.

7.3Describe the types of selectors and how their usage differs.

7.4 Use the Cascade, specificity, and inheritance to apply styles accurately to various elements in a layout.

8. Use media queries to create a responsive website.

Objectives:

8.1Create a media queryusing the proper syntax.

8.2 List the different media features that can be targeted in a query.

8.3 Explain browser support for different types of media queries.

8.4 Describe why small-screen first is the best approach for adding media queries to a web page.

8.5 Explain what a breakpoint is, and how to choose where a breakpoint should be added to a layout.

9. Use CSS to modify the appearance of inline images and modify the backgrounds of elements.

Objectives:

9.1Use float to position images.

9.2Create a simple CSS image gallery.

9.3Use CSS to add decorative images to a web page.

9.4 Apply background colours and gradients to elements.

10. Use CSS to modify the appearance of tables and lists.

Objectives:

10.1Customize the appearance of lists.

10.2Customize the appearance of tables.

10.3Create a responsive navigation menu for a web site using HTML and CSS.

10.4 Use a plugin to create a responsive navigation menu.

11. Use CSS to modify the position of elements on the page.

Objectives:

11.1 Identify the uses and potential issues with static,relative, fixed and absolute positioning.

11.2Modify the position of an element usingeach of the types ofpositioning.

11.3Describe the stacking order and z-index of overlapping elements.

11.4Create a sticky menu bar with fixed positioning.

12. Use box properties together with float and position to achieve advanced web page layout.

Objectives:

12.1 Create a multiple-column layout using at least three different methods.

12.2 Use a framework to create a multiple-column layout.

12.3 Explain how Flexbox and Grid layout work differently than older methods of layout.

13. Outline issues with browser and device compatibility.

Objectives:

13.1 Define performance and explain why it needs to be considered when building a website.

13.2 Describe the various factors that can affect the performance of a website.

13.3. Use browser tools to measure performance.

13.4 Explain the necessity of testing a website on different device types, device brands, browsers, and operating systems.

13.5 Test a website using a device simulator.

13.6 Use a tool like Modernizer to target specific code based on device features.