

Web Development

Course Description

Web technology is a critical application area for computer science, and an understanding of the internet and web technologies will provide you with the ability to reason about many of the computer systems you interact with on a daily basis. This module provides you with a foundation skill set in web technology and web languages that you can build upon in later, more advanced modules in the BSc Computer Science programme.

You will learn the critical languages of the web: HTML, CSS and Javascript. Using HTML and CSS, you will learn how to markup, layout and style web content. You will learn about the document object model and how you can dynamically manipulate it with JavaScript to create interactive web pages. You will consider accessibility and usability issues, and how you can overcome them. You will learn about website deployment and how you can use it to make your websites accessible to other people. In the first part of the module you will require to work in teams on a small project and submit it as a team project in the middle of the study session. In the second part of the study session you will be working on an individual project.

Course Goals and Objectives

Upon successful completion of this course, a learner will be able to:

1. Explain the benefits and limitations of client-side interactivity for web applications
2. Understand web page structure and the Document Object Model
3. Use a text editing tool to create a web application
4. Manipulate the DOM and respond to events using Javascript
5. Apply knowledge of usability and accessibility issues to the design of websites
6. Create an interactive, usable and accessible website using well formed HTML, CSS and Javascript.

Textbook and Readings

There is no required textbook for this module. The module will draw on a number of different, largely web-based, public resources as well as the resources produced as bespoke material for this module. Some key online references are:

- W3 Schools Online Web Tutorials, <http://www.w3schools.com>
- Mozilla Developer Network Tutorials, <https://developer.mozilla.org/>
- Marijn Haverbeke, 2012 "Eloquent JavaScript: A Modern Introduction to Programming" - free online text <http://eloquentjavascript.net>.

Specific readings for each topic are listed with direct links to free online recourses that provide additional material on the topics of this course.

Course Outline

The course consists of ten topics divided in 20 weeks that focus on key concepts.

Topic 1. Introduction to the web	<p>Key concepts: You will continue learning about the technologies that are the foundation upon which all web sites are built. We will consider the roles of technologies such as the web server, the client browser, HTML and CSS technologies.</p> <p>Learning outcomes:</p> <ul style="list-style-type: none"> • Explain the role of the DOM of web pages and be able to interrogate it from within a browser. • Explain the purpose and construction of core web HTTP protocols and associated error messages. • Assemble a simple web site by uploading HTML and CSS files to a web server.
Topic 2. Designing web sites	<p>Key concepts: you will learn the importance of design when preparing to construct web pages and whole web sites. You will also learn the practical techniques of sketching and using a design to prepare annotated wireframes.</p>

	<p>Learning outcomes:</p> <ul style="list-style-type: none"> • Explain the need for design when developing web sites. • Describe the role of different parts of typical web pages. • Construct a wireframe showing the essential parts of a web page. • Construct alternative wireframes for use with displays such as mobile devices and printers. • Annotate the wireframes to show the type of content to be presented in the various parts of the pages. • Annotate the wireframes to show how they will be presented.
Topic 3. Introduction to HTML	<p>Key concepts: You will learn about how HTML is used to construct the framework within which content can be placed and presented, particularly taking into account the significance or meaning of the content being presented.</p> <p>Learning outcomes:</p> <ul style="list-style-type: none"> • Describe the role of HTML in describing and providing a framework for placing content within a web page. • Access a variety of online and print resources that provide the latest reference for the HTML languages. • Explain the function of a “head” section in an HTML document and the essential elements placed there. • Choose the appropriate formatting and media tags to support the content of an HTML page. • Attribute the semantics of content to the appropriately constructed HTML elements.

	<ul style="list-style-type: none"> Construct an HTML document using a code-editing application and upload the files to a web server.
Topic 4. Introduction to CSS	<p>Key concepts: You will learn how styling instructions can be applied to HTML structures in web pages, focusing on the appearance of the text in the pages.</p> <p>Learning outcomes:</p> <ul style="list-style-type: none"> Describe the role of style and presentation in reinforcing the message of content for target users. Use a wireframe to produce a detailed responsive style design. Style web pages using in-page style instructions and external style sheets. Compose appropriate CSS for setting the appearance of web page elements, including bounded spaces and fonts.
Topic 5. Positioning in CSS	<p>Key concepts: You will learn about the various techniques available within CSS to position HTML elements and to lay out web pages.</p> <p>Learning outcomes:</p> <ul style="list-style-type: none"> Manipulate HTML elements with block or inline behaviours are displayed. Understand the purpose of and use the various font and element size definitions. Use the layer functions within the CSS language to position elements in front of or behind other elements. Use the various CSS absolute and relative positioning functions to position HTML within a page.

<p>Topic 6. Responsive web sites</p>	<p>Key concepts:</p> <p>Learning outcomes:</p> <ul style="list-style-type: none"> • Describe the principles to be applied when designing and implementing responsive web sites. • Use the @Media CSS function to interrogate the client platform. • Create responsive web sites, taking advantage of the responsive design assistance provided by frameworks. • Design using columns and grids within responsive implementations. • Apply the constraints of the client browser viewport to dynamically position HTML elements.
<p>Topic 7. Introduction to JavaScript for the web</p>	<p>Key concepts: You will learn about the scope and syntax of the JavaScript language and will practise writing functional JavaScript code.</p> <p>Learning outcomes:</p> <ul style="list-style-type: none"> • Describe the role of JavaScript in providing dynamic user-controlled client-side web site behaviour. • Select the appropriate JavaScript commands and construct syntactically correct code. • Invoke embedded or external JavaScript code. • Construct JavaScript functions and objects.
<p>Topic 8. React to the user</p>	<p>Key concepts: You will learn how to produce dynamic web pages that respond to user inputs by manipulating HTML elements and CSS styles.</p>

	<p>Learning outcomes:</p> <ul style="list-style-type: none"> • Manipulate the HTML DOM using JavaScript. • Produce dynamic web pages that respond to user inputs. • Manipulate CSS styles using JavaScript.
Topic 9. JavaScript template libraries	<p>Key concepts: You will learn how to use JavaScript libraries and plugins to simplify the creation of functional JavaScript code.</p> <p>Learning outcomes:</p> <ul style="list-style-type: none"> • Explain the use of client side and server side templating engines. • Simplify the use of JSON data using JavaScript plugins/templates such as Handlebars.js • Handle data in JavaScript code.
Topic 10. Web hosting and professional practices	<p>Key concepts: You will learn how to deploy web sites that are inclusive and do not exploit users, working within recognised professional practices.</p> <p>Learning outcomes:</p> <ul style="list-style-type: none"> • Consider the services required from a web hosting provider. • Construct web sites that do not exploit users, their identities or data.

Learning Activities of This Course

The course is comprised of the following elements:

- Team project. You will be expected to work throughout the first half of the session with a team of other students. Your team will design and build a web site for a fictitious theme park. The team project comprises of four parts. In the first part, you will design the web

site using the wireframe method. In the next parts, you will build the HTML pages, add the content and then style a large-screen and small-screen mobile version of the site. Your team will have its own Slack channel which you can use to interact with your team.

- Lecture videos. In each topic the concepts you need to know will be presented through a collection of short video lectures. You may stream these videos for playback within the browser by clicking on their titles or download the videos.
- Practice Quizzes. Each topic will include one practice quizzes, intended for you to assess your understanding of the topics. You will be allowed unlimited attempts at each practice quiz. Each attempt may present a different selection of questions to you. There is no time limit on how long you take to complete each attempt at the quiz. These quizzes do not contribute toward your final score in the class.
- Peer Reviewed Assignments. There will be one peer reviewed assignment for every part of the team project. You will read a short prompt, then answer several questions about the information provided and perform a task. You will then be required to review three of your peers' submissions. Peer reviewed assignments will not contribute to your final grade, but will be valuable practice for your coursework and exam.
- Discussion Prompt. Each topic may include one discussion prompt. You will see the discussion prompt alongside other items in the lesson. Each prompt provides a space for you to respond. After responding, you can see and comment on your peers' responses. All prompts and responses are also accessible from the general discussion forum and the module discussion forum.
- Readings. Each topic may include several suggested readings. They are good supplementary materials for you to further understand the course topics.

How to Pass This Course

The course has two major assessments:

- Coursework 1: this consists of two activities, i.e. a portfolio of exercises completed by a team and an individual reflective essay. Both elements will be assessed half way through course (week 12)
- Coursework 2: this consists of code review and website that you develop and submit on the Coursera platform and which will be assessed in the end the course in week 22.

There are also several activities that are graded but have 0% weight. That means that they will not count towards your final grade, but they are a key part of your learning and you need to do them.

This is a detailed breakdown of all of the marks.

Activity	Required?	Deadline week	Estimated time per week	% of final grade
A single bundled team assignment	Yes	12	3 hours	20%
Individual reflective essay	Yes	12	3 hour	10%
Peer reviews	Yes	12	3 hours	0%
Web site submission	Yes	22	3 hours	70%