Week 1 - Introductions

Semester 1
2024

ISCG6420
Internet & Website Development

Table of Contents

1 Course Information

Graduate Profile

The Graduate Profile provides a description of required competencies for graduates of the Bachelor program

ISCG6420 aligns its learning outcomes with the Graduate Profile

5.3 Graduate Profile

Unitec Academic Board policy requires that degree graduates have:

- a competence to apply technical and conceptual thinking;
- self-reliance and motivation for continued learning and professional development;
- the capabilities to be effective communicators and team members;
- a broad educational base to enable them to relate to the wider context of society;
- enquiring, flexible, creative and critical attitudes towards contemporary issues and underlying theoretical concepts;
- an appreciation of management and information technology within the context of their qualification;
- awareness of bicultural issues and an ability to function positively in a multicultural society;
- an awareness of social, ethical and environmental issues.

Graduates from this degree will be technically competent and conceptually sound through the study of specific subject area courses. They will have developed self-reliance and motivation for continued learning through being immersed in a student centred learning environment.

Learning Outcomes

The Graduate Profile provides a description of required competencies for graduates of the Bachelor program

ISCG6420 aligns its learning outcomes with the Graduate Profile

ISCG6420:	INTERNET AND WEBSITE DEVELOPMENT				
Course number:	ISCG6420	**Level:**	6	**Credits:**	15
Main programme:	BCS			Elective	
Pathway(s):					
Requisites / Restrictions:	Pre-requisites: (ISCG5422 or ISCG5236) and (ISCG5420 or ISCG5235) Co-requisites: ISCG5430 or ISCG5237 Restrictions: ISCG6240				
Other programmes:	DipITS, GDCMP				
NZSCED field of Study:		**Delivery mode:**			

Hours directed:	Hours in the Workplace:	Hours Self-directed:	Total Learning Hours:
32.5		117.5	150

Outcome Statement:

To provide the students with knowledge of advanced concepts of the Internet and website design and development

Learning Outcomes:

Learning Outcomes
1. Explain and apply the fundamentals of CSS (Cascading Style Sheets).
2. Explain and apply the fundamentals of JavaScript.
3. Use current client-side website development languages/technologies create a complex commercial or educational website.
4. Use the features of Web/Multimedia authoring packages to create a complex commercial or educational website with effective navigational interface.

Learning Support Resources

Unitec provides support for all students to help with your studies.

These services are included in your tuition fees.
You pay for them. Make use of them.

More information can be found at:
- Student Central
- Unitec.ac.nz

- **Access 4 Success**
- **Learning Advisors**
- **Library Services**
- **MAIA – Maori Support**
- **The Pacific Centre**
- **International Advisors**
- **Under 25**
- **Peer Assisted Study**
- **Scholarships**
- **Many more**

Course Schedule

The course schedule is up on Moodle

Subject to change. Course progression and assessment scheduling may cause the course to differ from this schedule.

Week	Topic	Assessment
1 - 28/02	- Introduction to web technologies and development - IDE configuration - HTML and CSS fundamentals	
2 - 06/03	- HTML and CSS - Website analysis and prototyping - Case study	
3 - 13/03	- Designing websites - CSS animations - Storyboarding	Project 1 (20%) Introduction
4 - 20/03	- Web interactive applications - User interaction - Animations continued	Project 1 Checkpoint 1
5 - 27/03	- External storage - XML	Project 1 Checkpoint 2
6 - 03/04	- HTML5 Canvas - Drawing stored and runtime assets	Project 1 Submission
7 - 10/04	- Testing with web tools - Drag and drop	Project 2 (50%) Introduction
8 - 17/04	Mid-Semester Break	
9 - 24/04		
10 - 01/05	- Exam	Exam (30%)
11 - 08/05	- Javascript - Debugging	Project 2 Checkpoint 1
12 - 15/05	- Javascript libraries - JSON - Web APIs	
13 - 22/05	- Web frameworks	Project 2 Checkpoint 2
14 - 29/05	Study Break	
15 - 05/06	Project 2 presentation	Project 2 Submission

Assessments

Assignment 1 is an individual, take-home assessment. Students will use their HTML, CSS and JS skills to deliver a booking website with marketing.

Test will take place during a regularly scheduled class. Students will demonstrate their skill with JS-controlled canvas drawing and animation.

Assignment 2 is a two-section, group, take-home assessment. Students will collaborate to create an interactive resort website with 2D game, and present their work to the class.

Assessment types for ISCG6420

Name	Weight	Type	L.O.
Assignment 1	20%	Project (independent)	1, 2
Test	30%	Open-book on-site	1, 2
Assignment 2	50%	Project (group)	3, 4
(no exam)			

Photo by Windows on Unsplash

Course Prescription

Total course learning hours: 150	
Directed: 32.5	Self-directed: 117.5

2 Topic Information

Web dev history
HTML

HTML – HyperText Markup Language

1989: Tim Berners-Lee at CERN proposes a system using hypertext documents to share data over network fabric.
1990: Tim creates the specification for HTML, web server, and web browser.
1995: HTML working group at IETF create the HTML 2.0 specification.
1997: HTML 4.0 spec released, with favour for CSS over visual markup.
2014: HTML 5 spec released.
2019: WHATWG & W3C collaborate on a single version of HTML as a continually updated Living Standard.

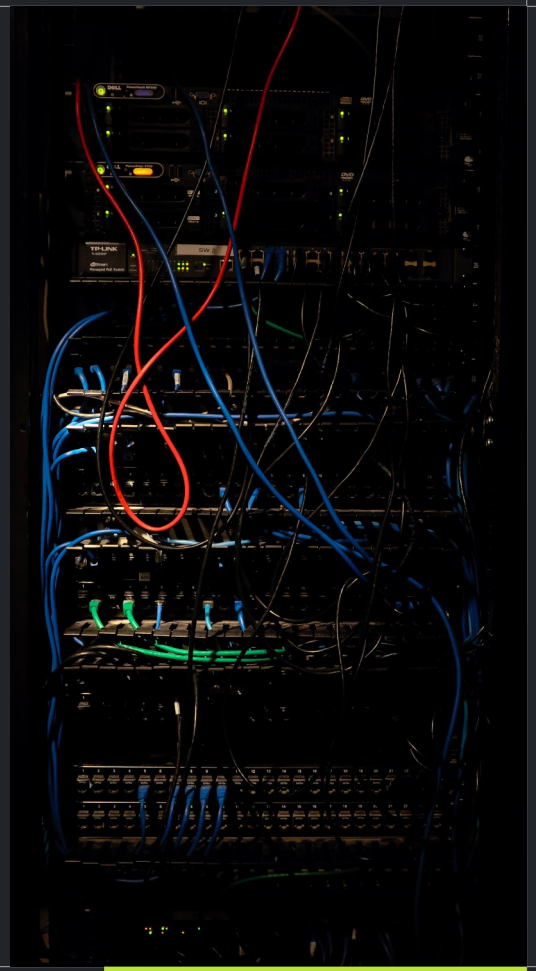


Photo by Neonbrand on Unsplash

Web dev history CSS

CSS – Cascading Style Sheet

1994: Håkon Wium Lie proposes CHSS while working with Tim Berners-Lee. Bert Bos proposes changes and becomes a co-author of CSS.
1996: CSS1 released with recommendation from W3C.
1997: HTML Editorial Review Board (ERB) is split into 3 working groups: HTML, DOM, CSS.
1998: CSS2 released with recommendation from W3C. Work begins on CSS3.
1999: First draft of CSS3 published. CSS moves away from single release model. Shifts to Module model with candidacy statuses. Candidates can be seen at https://www.w3.org/Style/CSS/specs

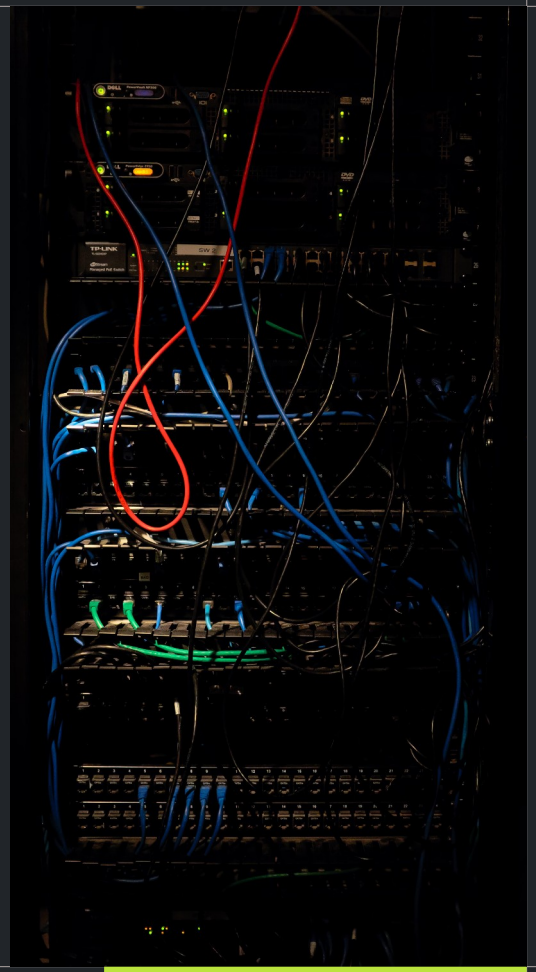


Photo by Neonbrand on Unsplash

Web dev history
JS

JS – JavaScript

1994: Netscape pursues creating a scripting language to make websites dynamic with two teams: one with Sun(Java), one with Brendan Eich.
1995: Netscape abandons Sun version and publishes Brendan Eich's with Netscape Navigator update. Netscape calls the language Javascript to leverage Sun's success with Java.
1996: Netscape submits JS to ECMA to become new international standard. Microsoft reverse engineers JS to create Jscript for Internet Explorer. IE success leads to Jscript replacing JS as de facto client-side language. Client-side scripting stagnates.
1997: Sun trademarks JavaScript to protect Java intellectual property.
2004: Netscape successor, Mozilla creates Firefox and competes with IE.

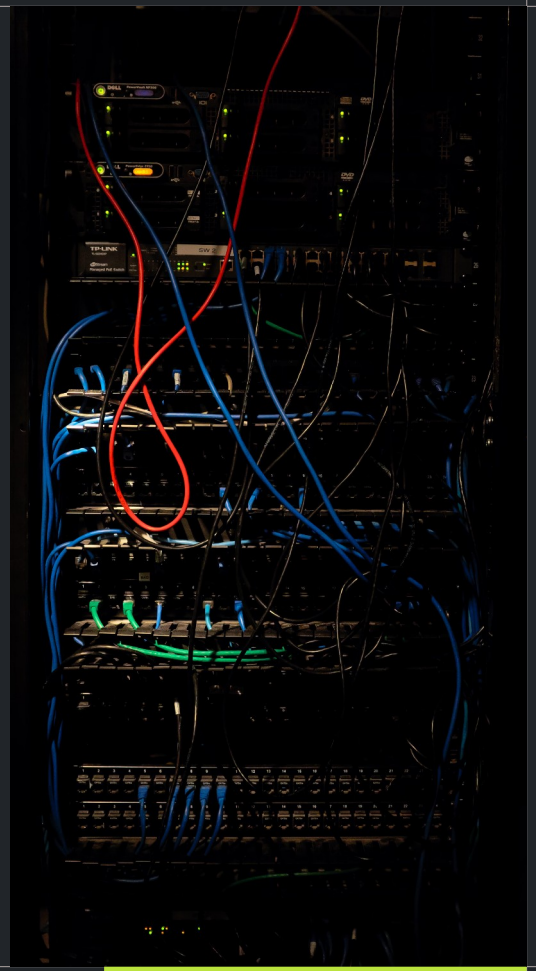


Photo by Neonbrand on Unsplash

Web dev history
JS continued

2005: James Garrett releases AJAX whitepaper. A new era in open source JS-based library development begins. jQuery released as a result.
2008: Google releases Chrome browser with V8 engine. Its Just In Time (JIT) compiler outperforms the competition.
2009: ECMA, Google, Microsoft, Mozilla agree to combine their work to create ECMAScript 5.
Node.js created using V8 engine. Use of JS outside of web browsers surges.
Sun microsystems acquired by Oracle Corporation.
2015: ECMAScript 6 released.

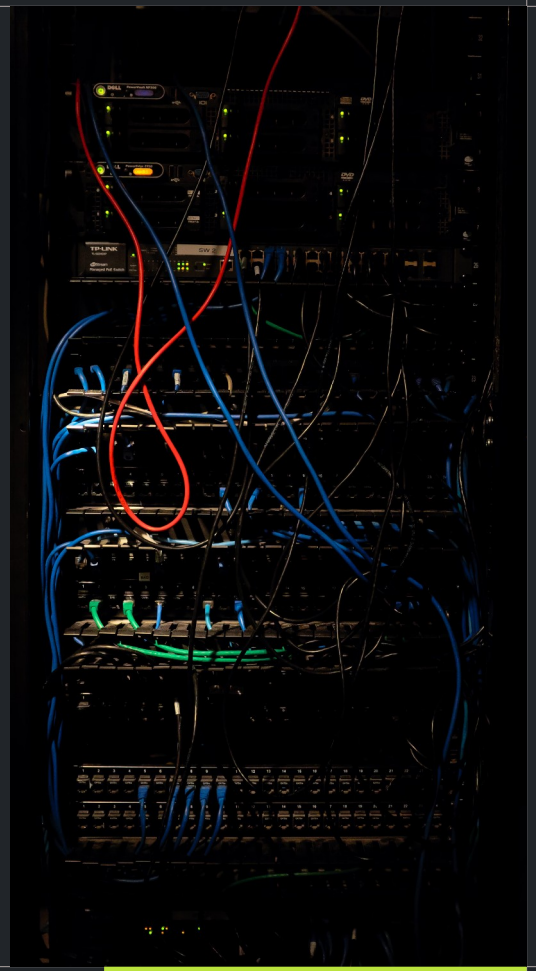


Photo by Neonbrand on Unsplash

Industry Members

Industry members consists of entities responsible for the standards and implementations of the languages.

HTML & CSS standards are maintained by WHATWG and W3C.

ECMAScript standards are maintained by ECMA International.

Implementation of these standards are maintained by software-compliant product owners such as the members of WHATWG.

01 HTML & CSS

World Wide Web Consortium (W3C)

Web Hypertext Application Technology Working Group (WHATWG)
Members: Apple, Google, Mozilla, Microsoft

02 JS / ES6

European Computer Manufacturers Association International (ECMA International)

WHATWG

Doc Sources

Working with W³ languages requires knowledge of language semantics and features. Many entities provide documentation with varying . Here are some well-established de facto sources of knowledge

01 HTML

WHATWG Living Standard spec:
https://html.spec.whatwg.org/multipage/

Mozilla Developer Network:
https://developer.mozilla.org/en-US/docs/Web/HTML/Reference

W3Schools (no relation to W3C):
https://www.w3schools.com/html/default.asp

02 CSS

W3C
https://www.w3.org/Style/CSS/specs.en.html

Mozilla Developer Network:
https://developer.mozilla.org/en-US/docs/Web/CSS

W3Schools (no relation to W3C):
https://www.w3schools.com/css/default.asp

03 JS / ES6

ECMA International 2022 Language Spec:
https://262.ecma-international.org/13.0/

Mozilla Developer Network:
https://developer.mozilla.org/en-US/docs/Web/JavaScript

W3Schools (no relation to W3C):
https://www.w3schools.com/js/default.asp

Front End Development

Front end development is a term that refers to developing and maintaining software that runs client-side

Web technologies have evolved significantly over the last 3 decades. Front end development has evolved with it. As frameworks like ReactJS and Angular have expanded their governance to more stack layers, the role of front end development has increased to include server-side or back end technologies.

Development roles that include front-end and back-end expertise are often refered to as “Full Stack”.
As app complexity increases, the need for full-stack developers decreases.

01 Front End

- Coding in client-side languages (HTML, CSS, JS)
- Working with front-end frameworks (React, Angular, Bootstrap)
- Building tools that improve site interaction
- Optimising for performance across supported browsers
- Troubleshooting and debugging
- Creating and implementing UI/UX design
- Prototyping application interfaces
- Integrating APIs

01 Back End

- Coding in server-side languages (PHP, C#, Java, Rust, Go, Python, Ruby)
- Working with back-end frameworks (Express.js, ASP.NET, Django, Flask)
- Working with web server technologies
- Supporting full application lifecycle
- Troubleshooting and debugging
- Building automation tools (build / compile, CI, CD, Testing)
- Developing APIs

Content referenced to Hackerrank

Front End Technologies

https://2022.stateofjs.com/en-US/libraries/front-end-frameworks/

What will I learn in ISCG6420

We will utilise HTML, CSS, JS, jQuery, AJAX, XML, JSON.

Progression depending, we may cover more technologies.

No optional topics will be included in any assessments.

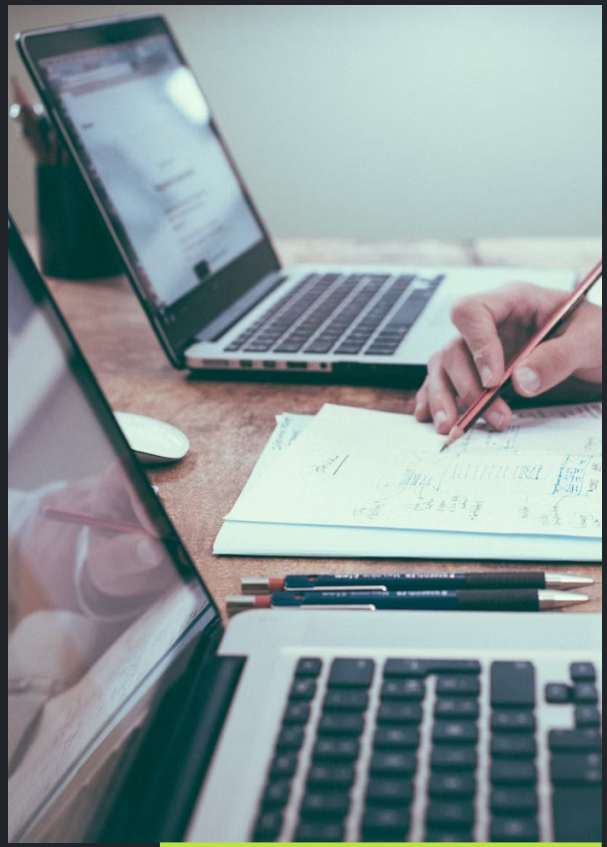


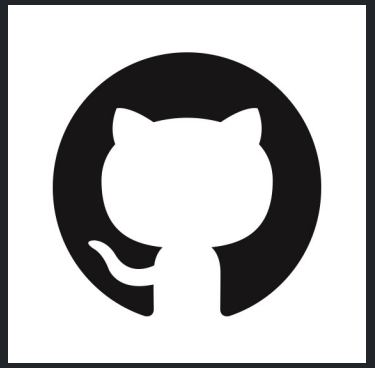
Photo by Scott Graham on Unsplash

Development Setup

Software Tools

Software required for this course:
Text editor or IDE
Git
Web browser with dev tools

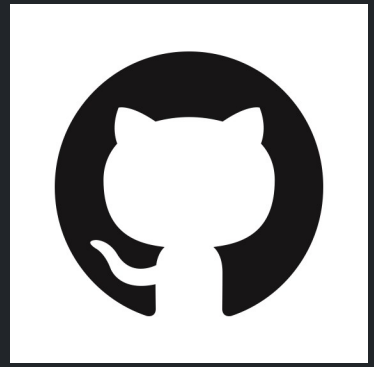
While you can get away with a text editor you are strongly encouraged to use an Integrated Development Environment (IDE).



Development Environment

Recommended IDE: VSCode or VSCodium

Github is recommended for your git repository as Github Pages will be used in an upcoming assignment.



Git CLI and Github Desktop are good options for interacting with Github.

Download Links

Use the links provided to download and install the software applications if you haven't done so already.

01 VSCode / VSCodium

https://code.visualstudio.com/Download

OR

https://vscodium.com/#install

02 Git / Github Desktop

https://git-scm.com/downloads

AND (optional)

https://desktop.github.com/

03 Good Reads

https://www.roboleary.net/tools/2022/05/31/vscode-vscodium-which-should-i-use.html

https://www.freecodecamp.org/news/a-beginners-guide-to-git-how-to-create-your-first-github-project-c3ff53f56861/

https://developer.mozilla.org/en-US/docs/Learn

Class Tasks

1

Setup your development environment. Ensure you have a code editor and git solution ready for development.

2

Create a new repository and clone it to your computer. Use your cloned repo directory as the root of your project (open the same folder in your code editor).

3

Work on exercise 1-4. These can be completed in your new cloned repo.

4

Push your repo changes to Github. This can be done by following step #3.C of the FreeCodeCamp git instructions linked on the previous slide.