

SIT120 Practical

Week 1 – Introduction to HTML, CSS



GOALS:

- Gain better understanding of the web applications.
- Introduction to basics of web designing.
- Note 1-2 sentences reflections for each tasks

Please do not copy examples from the web or unit site for your tasks and try to be as creative as possible.

Please check with unitstaffs for helphub and individual support to discuss your ideas, queries and project design details.

COFFEE CATCH-UPS

Open to all commencing students in the
Faculty of Science, Engineering and Built Environment

Burwood

13 July 1.30 – 2.30pm, Corner Café

Brought to you by the Peer Support Network Program

deakin.edu.au/sebe/peer-support-network

Unit Information- 10 mins



Read the unit guide thoroughly to follow up unit structure and assignment deadlines.



Start using the Microsoft Teams channel to share ideas, interesting things you've found related to this unit.



To be successful in this unit, you must:

- Read all materials in preparation for your classes or seminars, and follow up each with further study and research on the topic;
- Start your assessment tasks well ahead of the due date;
- Read or listen to all feedback carefully, and use it in your future work;

Unit Information Contd...

- In brief, these are the assessment tasks for this Unit
- Project analysis and design 10%, project implementation and presentation 40%, practical portfolio (Upskilling) 50%.

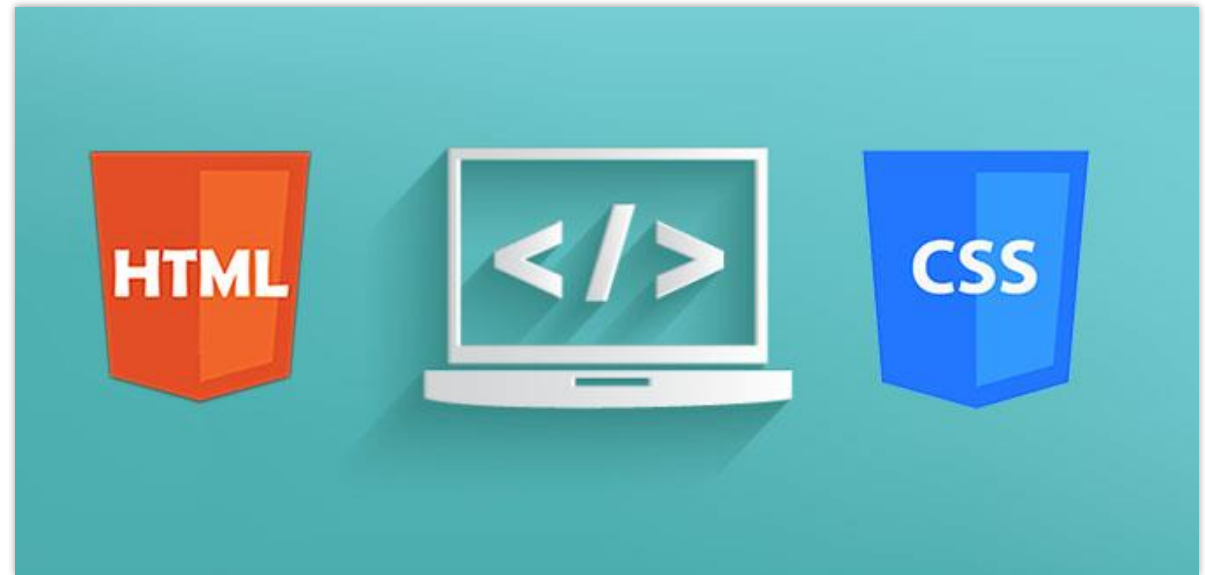
Preliminaries 10 mins

- In this Trimester we use **Codepen and jsFiddle** to play and plug with responsive web page applications. **jsFiddle** is a custom environment (based on user selections) to test (or fiddle with) your **JavaScript**, **HTML**, and **CSS** code right inside your browser.
- It is mandated to signup for a free **Codepan and jsFiddle** account to progress in this unit
- Signup link
<https://codepen.io/accounts/signup/user/free/>
<https://jsfiddle.net/user/signup/>
- Create public Github repository: All the code you write in this unit has to be saved separately in their respective format(i.e. html, css, js) in your local drives for assessment. Upload in github regularly and share with unit staffs.
- Visual Studio Code: a text editor with many powerful extensions. <https://code.visualstudio.com/>
- We are going to use **Vue.js** Framework in this unit for **responsive JavaScript programming**.
- Research: Why Vue.js???



HTML, CSS & JavaScript:

- An overview:
- **HTML** provides the *basic structure* of sites, which is enhanced and modified by other technologies like CSS and JavaScript.
- **CSS** is used to control *presentation, formatting, and layout*.
- **JavaScript** is used to control the *behaviour* of different elements.



HTML

- HTML stands for HyperText Markup Language. "Markup language" means that, rather than using a programming language to perform functions, HTML uses tags to identify different types of content and the purposes they each serve to the webpage.

```
<html>
```

```
<head>
```

```
<title>This Is Your Title </title>
```

```
</head>
```

```
<body>
```

```
<h1>This Is Your Header </h1>
```

```
<p>This is your paragraph. </p>
```

```
</body>
```

```
</html>
```

CSS

- CSS stands for Cascading Style Sheets. This programming language dictates how the HTML elements of a website should actually appear on the frontend of the page.
- HTML provides the raw tools needed to structure content on a website. CSS, on the other hand, helps to style this content so it appears to the user the way it was intended to be seen. These languages are kept separate to ensure websites are built correctly before they're reformatted.

Inline CSS

```
<p style="color: blue;">This is a paragraph.</p>
```

Internal CSS

```
<head>  
  <style type = text/css>  
    body {background-color: blue;}  
    p { color: yellow;}  
  </style>  
</head>
```

External CSS

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


JavaScript

- JavaScript is a logic-based programming language that can be used to modify website content and make it behave in different ways in response to a user's actions. Common uses for JavaScript include confirmation boxes, calls-to-action, and adding new identities to existing information.
- JavaScript can change HTML content, Attribute Values and change CSS styles. JS can also show or hide HTML Elements.

```
<script type="text/javascript">  
  function getInspiration() {  
    if (morningDay == "depressed")  
      start.Coding & get.Awesome;  
  }  
</script>
```

Task 1 – Create an HTML Page (15 minutes)

- Create an HTML page that includes at least five main HTML tags (like image, references, tables, forms, etc).
- Visit **unit resources and <https://www.w3schools.com/html/>** for more information on HTML details
- Study examples and use ideas from the web

Task 2 – Add CSS to your HTML page (15 minutes)

- Add inline, internal and external styling rules to the HTML page you have created in previous task.
- Give a column layout to the web page.
- Be as creative as you can.
- Visit our [Deakinsync unit resources](https://d2l.deakin.edu.au/d2l/le/content/1031929/Home)
(<https://d2l.deakin.edu.au/d2l/le/content/1031929/Home>)
- <https://www.w3schools.com/css/default.asp> for more information on CSS details
- Study examples and use ideas from the web

Task 3 – Adding JavaScript

- Try adding responsive script elements some where in the page you have created.
- For example a responsive button that shows date and time.
- You can find some **good start to JavaScript elements** in our **unitsite** and <https://www.w3schools.com/js/default.asp>
- Try to create a JavaScript page that lists scores of a student, calculate total score and display average.

```
<!DOCTYPE html>
<html>
<body>

<button type="button"
onclick="document.getElementById('demo').innerHTML =
Date()">
Click to display Date and Time.</button>
<p id="demo"></p>

</body>
</html>
```

Task 4 – How to use Git and GitHub

- Creating a GitHub account. <https://github.com/>
- Install Git on your machine. <https://git-scm.com/>
- Create your first repository then push one commit to the main branch.

Task 5 – Vue.js Framework (homework)

- Get started to learn Vue.js
- Create a basic todo component using Vue
- Visit this page for detailed information
<https://v2.vuejs.org/v2/guide/>

Learn two other examples from
<https://vuejs.org/v2/examples>

Todos:

- ☐ Assignment 1: Project
- ☐ Assignment 2: Project and PPT
- ☒ Assignment 3: Portfolio

Assessment Discussion (15 minutes)



OPEN THE SIT120
DEAKINSYNC SITE AND
LOOK THROUGH THE
VARIOUS SECTIONS
INCLUDING RESOURCES
AND DISCUSSIONS.



RESOURCES (MODULES,
ASSESSMENT,
SOFTWARE, LECTURE
RECORDINGS)



MICROSOFT TEAMS



READ THE
INFORMATION SHEET
FOR ASSIGNMENT 1,
AND FIND THE
SUBMISSION DROPBOX
SO YOU CAN BE
PREPARED.

Before next class

- Read the Unit guide
- Come and talk with us about your proposal
(Monday evening helphub 6-7 pm and Wednesday 6-7 pm) with Nisha and Ray
- Start using the dedicated Microsoft teams channel to share ideas with peers, interesting things you have found related to web app development.
- Read assessment task 1 information sheet and get prepared to start working on it.
- Save all the work you have done in this practical to your local drive and commit to Github.