Week 12 Tutorial

COMP10001 – Foundations of Computing

Semester 2, 2025

Clement Chau

- Digital Ethics
- Algorithms
- HTML

Agenda

1. Week 12 Discussion – **Tutorial sheet** (~ 40 mins)
2. W12 GenAI Worksheet – **Ed Lessons** (~ 30 mins)
3. One-on-one Q&A (~ 40 mins)

12 (20/10)	Advanced Lecture - Generative AI	Exam Revision	Subject wrap-up	Week 12 tutorial sheet Week 12 tutorial solutions	• Ed worksheets 16 and 17 due (20/10 at 6 pm)

Ed worksheets 16 & 17 due (20/Oct, Monday at 6 pm)

Revision: Digital Ethics

1. Digital Ethics: Here are some of the points in the ACM (Association for Computing Machinery) Code of Ethics and Professional Conduct from `https://www.acm.org/code-of-ethics`

   (a) Contribute to society and to human well-being, acknowledging that all people are stakeholders in computing

   (b) Avoid harm

   (c) Be honest and trustworthy

   (d) Be fair and take action not to discriminate

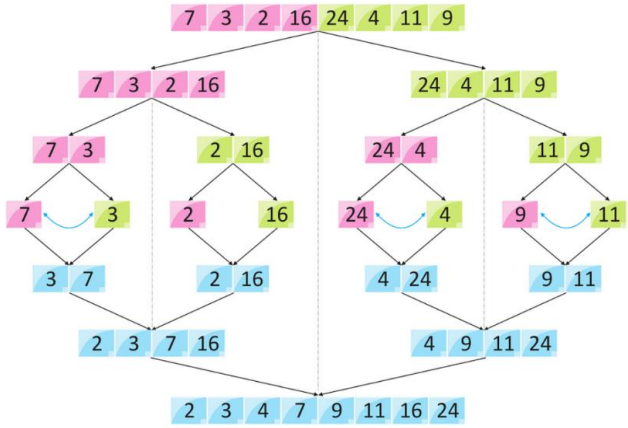
   (e) Respect the work required to produce new ideas, inventions, creative works, and computing artefacts

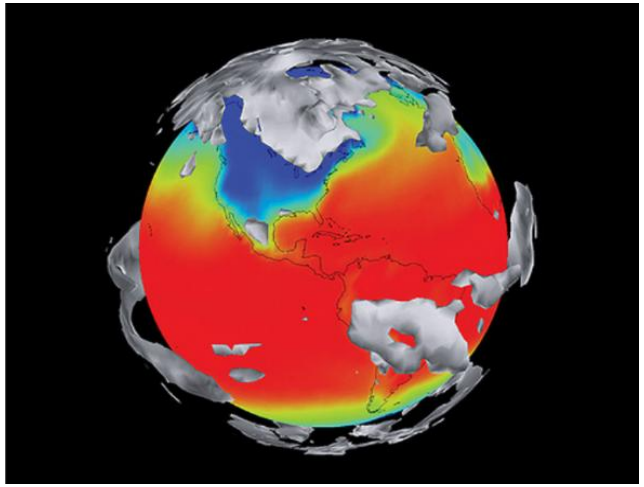
   (f) Respect privacy

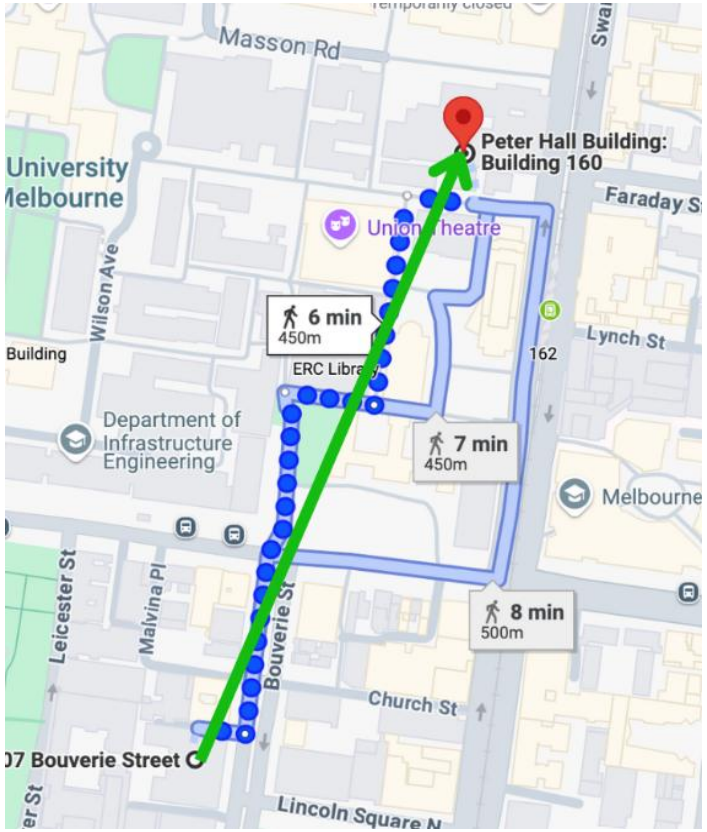
   (g) Honour confidentiality

Revision: Algorithms

- Criteria to judge algorithms?
  - Correctness
  - Efficiency
    - Time Complexity
    - Space Complexity
- Two types of algorithms?
  - Exact
    - Brute Force
    - Divide and Conquer
  - Approximate
    - Simulation
    - Heuristic Search







Revision: Linear Search and Binary Search

- Linear Search
  - Does **not** require the elements to be sorted first.
  - Goes through all elements in the list, **iteratively**.
  - **Brute-Force Approach.**
  - Worst case scenario (Time Complexity): **O(n)**
- Binary Search
  - Requires the elements to be **sorted** first.
  - For every iteration,
    - Divides the list into 2 equal length.
    - Continues **searching on only one-half** of the list and **discards the other**.
    - Repeats until an element is found, or it's exhausted of elements to search on (Element is not found in the list).
  - **Divide and Conquer Approach.**
  - Worst case scenario (Time Complexity): **O(log n)**

Revision: HTML

4. Recap the below concepts on HTML (covered in Week 12 lecture 1):

   - *HTML* (Hypertext Markup Language) is a markup language (not a programming language like Python) used to take a document composed of text and other media and communicate how it is to be rendered for display. An example HTML file is shown in the next page.
   - HTML *Tags* are <angle bracket delimited>commands which give instruction about how a document is to be formatted. Often there will be an "opening" <tag>and "closing" </tag>pair of tags where the second includes a slash to show it is closing the first.
   - Tags which we've covered include
     - *text formatting*: <b>(bold text), <i>(italic text) and <u>(underline)
     - *structural tags*: <html>(covers an entire html document), <head>(header section) and <body>(the body content of a document)
     - *lists and tables*: (unordered bullet-point list), (ordered list), (denoting a single list item), <table>(holding a table) which contains <tr>(table rows) and <td>(table cells)
     - *media tags*: <a>(hyperlink to a URL), <img>(an image), <audio>(some audio) and <video>(a video)
   - HTML *entities* are the "special characters", such as < (<),   (space) and & (&).

TuteSheet W12 – Ethics (1)

1. Digital Ethics: Here are some of the points in the ACM (Association for Computing Machinery) Code of Ethics and Professional Conduct from `https://www.acm.org/code-of-ethics`

   (a) **Contribute to society and to human well-being, acknowledging** that all people are stakeholders in computing
   (b) **Avoid harm**
   (c) **Be honest and trustworthy**
   (d) **Be fair and take action not to discriminate**
   (e) **Respect the work required** to produce new ideas, inventions, creative works, and computing artefacts
   (f) **Respect privacy**
   (g) **Honour confidentiality**

   Considering these points, discuss with your peers and answer the following open-ended questions.

   - When handling job applications, company A uses software to automatically screen applicants' resumes to decide whether they can proceed to the interview process. Some of the criteria that company A uses are based on existing employee's traits. Do you like this idea? Argue your point from the viewpoint of both company A, the applicant, and the wider society.

     Think about: What are the "traits"? What traits, or under what situation can this approach be used? Is it fair? Does it discriminate? What negative effect might it have on the wider community?

TuteSheet W12 – Ethics (2, 3)

(a) **Contribute to society and to human well-being, acknowledging** that all people are stakeholders in computing
(b) **Avoid harm**
(c) **Be honest and trustworthy**
(d) **Be fair and take action not to discriminate**
(e) **Respect the work required** to produce new ideas, inventions, creative works, and computing artefacts
(f) **Respect privacy**
(g) **Honour confidentiality**

- Supermarket B uses surveillance cameras in their self-checkout system. Do you think this is a good idea? Why?

  Think about: Are customers informed of the existence of cameras? Does the supermarket respect their privacy? Can you think of an ethical and cost-effective way to avoid stealing?

- An online gambling website C is looking for an employee with a computing background. They want the employees to find patterns in customer online behaviours and provide personalised solutions for each customer. Doing so can generate more income for the website. One of your friends wants to apply for the job. What do you think of their decision?

  Think about: What type of gambling is involved (legal or illegal)? Does this job contribute to human well-being and society? Why or why not?

TuteSheet W12 – Ethics (4)

(a) **Contribute to society and to human well-being, acknowledging** that all people are stakeholders in computing
(b) **Avoid harm**
(c) **Be honest and trustworthy**
(d) **Be fair and take action not to discriminate**
(e) **Respect the work required** to produce new ideas, inventions, creative works, and computing artefacts
(f) **Respect privacy**
(g) **Honour confidentiality**

- Student D is a part-time data analyst. Student D's employer gave them some confidential data to process to inform the business decisions. However, student D was right in the middle of the exam period. Due to the time pressure, they uploaded the data online and asked GenAI to do the task (such as ChatGPT), got the high-level analysis result, handled the result to their employer and claimed this was their own work. What are the problems of student D's approach?

Think about: Does student D honour confidentiality? Is student D honest? Is the analysis result trustworthy?

TuteSheet W12 – Algorithms

What is an Algorithm?

: A set of steps for solving an instance of a particular problem type.

Criteria to judge algorithms?

- ***Correctness***
  - should terminate for every input with the correct output
  - incorrect algorithms can either: (a) terminate with the wrong output; or (b) not terminate
- ***Efficiency***
  - ***runtime***: run as fast as possible
  - ***storage***: require as little storage as possible

Two types of algorithms?

- ***Exact***
  - Calculate the solution(s), **with a guarantee of correctness**.
  - (e.g.) ***brute force, divide and conquer***
- ***Approximate***
  - Estimate the solution(s), ideally with an **estimate of how "close"** this is to the exact solution(s).
  - (e.g.) ***simulation, heuristic search***

TuteSheet W12 – Algorithms

Some algorithm families:

- ***Brute-Force (Exact)***

  Assumptions
  - A candidate answer is **easy to test**
  - The set of candidate answers can be generated **exhaustively**

  Strategy
  - Generate candidate answers and test them **one by one until a solution is found**

  Examples:
  - ***Linear search*** | *Testing whether a number is prime* | *Coins and Denominations*

- ***Divide and Conquer (Exact)***

  Strategy
  - Solve a **smaller sub-problem** akin to the original problem
  - Extend the sub-solution to create the solution of the original problem

  Examples:
  - ***Binary search*** | *Many sorting algorithms*

TuteSheet W12 – Algorithms

Some algorithm families:

- ***Simulation (Approximate)***

  Strategy
  - Play out a complex scenario, often with random inputs
  - Multiple runs help verify the stability of an answer
  - Useful if it is possible to **describe the "rules" of a system**

  Applications
  - *Weather forecasting | Movement of planets | Prediction of share markets*

- ***Heuristic Search (Approximate)***

  Strategy
  - Search for a solution by exploring more promising avenues first
  - Often **sacrifice optimality for efficiency**, though some heuristic methods can be optimal.
  - Hope that the solution is close to optimal, but usually w/o proof.
  - Useful for solving inherently "hard" problems that have intractably large search spaces

TuteSheet W12 – Linear vs. Binary

3. Search the following sorted lists for the number 8, using **(a)** Linear search (Brute-Force approach) and **(b)** Binary search (Divide and Conquer approach)

   Think about the best, worst and average case scenarios of these algorithms. For example, can the best case scenario of a Brute-Force algorithm be faster than running the same task with a more clever algorithm?

(a)	1	2	4	5	8	9	10	12	15	19	21	23	25
(b)	8	9	11	15	16	17	22	24	27	28	29	32	33
(c)	2	4	5	6	7	9	11	12	13	15	19	22	25

Linear search

- It iterates through the list **from start to end**, testing whether each item of the list is the one being searched for.
- ***Brute-Force*** *approach because there is no logic to the order of the search*

Binary search

- It **starts at the middle of the list** and proceeds to search its upper half or lower half depending on whether the middle item is smaller or larger respectively than the value being searched for.
- ***Divide and Conquer*** *approach as the area to be searched is divided in half each iteration*

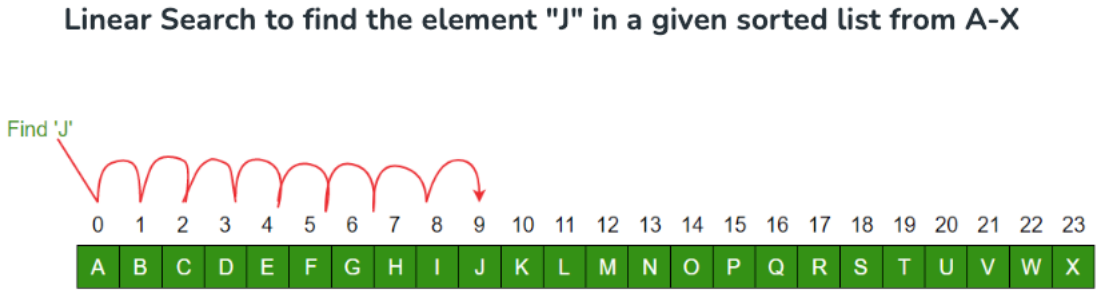
TuteSheet W12 – Linear vs. Binary

(a)	1	2	4	5	8	9	10	12	15	19	21	23	25
(b)	8	9	11	15	16	17	22	24	27	28	29	32	33
(c)	2	4	5	6	7	9	11	12	13	15	19	22	25

Linear search

It iterates through the list **from start to end**, testing whether each item of the list is the one being searched for.

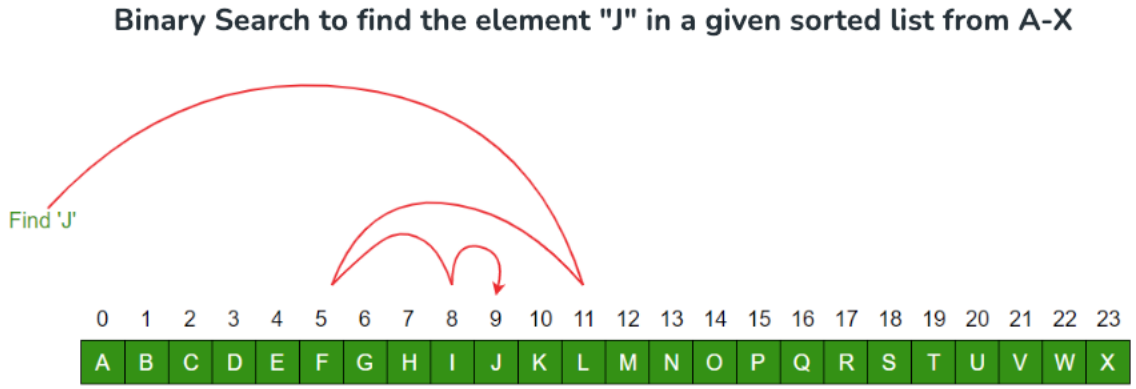
- *Linear search may be faster than binary search if you get lucky, but you can never depend on a best case scenario!*



Binary search

It **starts at the middle of the list** and proceeds to search its upper half or lower.

- *much more **reliable** than linear search*
- *take a **fairly short amount of time***



TuteSheet W12 – Linear vs. Binary

Linear Search	Binary Search
In linear search input data need not to be in sorted.	In binary search input data need to be in sorted order.
It is also called sequential search.	It is also called half-interval search.
The time complexity of linear search **O(n)**.	The time complexity of binary search **O(log n)**.
Multidimensional array can be used.	Only single dimensional array is used.
Linear search performs equality comparisons	Binary search performs ordering comparisons
It is less complex.	It is more complex.
It is very slow process.	It is very fast process.

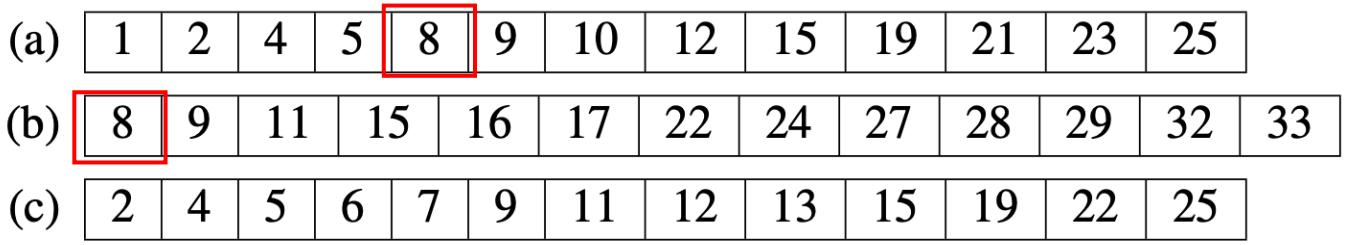
Binary search Time Complexity:

O(log n) :
*as the **input size n grows**, the number of operations (or steps) needed **grows much more slowly** , not linearly.*

So, for a list of:
*n = 8 → **$\log_2(8) = 3$***
*n = 16 → **$\log_2(16) = 4$***

https://www.geeksforgeeks.org/linear-search-vs-binary-search/

TuteSheet W12 – Algorithms




```
Testing List (a): [1, 2, 4, 5, 8, 9, 10, 12, 15, 19, 21, 23, 25]
Linear Search:  Found at index 4 | Steps: 5 | Time: 3.58 µs
Binary Search:  Found at index 4 | Steps: 3 | Time: 2.86 µs

Testing List (b): [8, 9, 11, 15, 16, 17, 22, 24, 27, 28, 29, 32, 33]
Linear Search:  Found at index 0 | Steps: 1 | Time: 1.67 µs
Binary Search:  Found at index 0 | Steps: 3 | Time: 1.91 µs

Testing List (c): [2, 4, 5, 6, 7, 9, 11, 12, 13, 15, 19, 22, 25]
Linear Search:  Not found | Steps: 13 | Time: 1.91 µs
Binary Search:  Not found | Steps: 4 | Time: 1.67 µs
```

Linear Search:

Best Case (b):

8 is the first element

→ FOUND immediately → ***O(1)***

Worst Case (c): ***O(n)***

Binary Search:

Best Case **O(1), but not appeared**

- *Middle: index 6 = 22 →*
  8 < 22 → go left
- *New middle: index 2 = 11 →*
  8 < 11 → go left
- *New middle: index 0 = 8 → FOUND*

Worst Case ***O(log n),*** could say (c)

TuteSheet W12 – HTML

4. Recap the below concepts on HTML (covered in Week 12 lecture 1):

   - *HTML* (Hypertext Markup Language) is a markup language (not a programming language like Python) used to take a document composed of text and other media and communicate how it is to be rendered for display. An example HTML file is shown in the next page.
   - HTML *Tags* are <angle bracket delimited>commands which give instruction about how a document is to be formatted. Often there will be an "opening" <tag>and "closing" [redacted]pair of tags where the second includes a slash to show it is closing the first.
   - Tags which we've covered include
     - *text formatting*: [redacted](bold text),[redacted](italic text) and [redacted](underline)
     - *structural tags*: [redacted](covers an entire html document), [redacted](header section) and [redacted](the body content of a document)
     - *lists and tables*: [redacted](unordered bullet-point list),[redacted](ordered list), [redacted]denoting a single list item), <table>(holding a table) which contains[redacted](table rows) and[redacted](table cells)
     - *media tags*: [redacted](hyperlink to a URL), <img>(an image), <audio>(some audio) and <video>(a video)
   - HTML *entities* are the "special characters", such as < (<),   ([redacted]) and & ([redacted].

TuteSheet W12 – HTML

5. Below is an incomplete HTML script. After completion, the result will look like Fig 1. Answer the following questions:

- Complete the HTML by filling the “XXXXXXXX" part. You can run your HTML code and check the result here: `https://www.w3schools.com/html/tryit.asp?filename=tryhtml_default`
- What is the “link to happiness"? What if the image is not shown properly?

→ *If the image is not shown properly, the alternative text “smiley" will be shown*

```
<!DOCTYPE html>
<html>
  <body>
    XXXXXXXX
      <li>
        <ul>
          XXXXXXXX
          <li><u>underline</u></li>
          <li><i>italic</i></li>
        </ul>
      </li>
      <li>
        <table border="1">
          XXXXXXXX
          <tr><td>Lawnmower full of rizz</td><td>COMP10001</td><td>76</td></tr>
          <tr><td>Slayed the CompRPG game</td><td>COMP10002</td><td>81</td></tr>
        </table>
      </li>
      <li><a href='https://canvas.lms.unimelb.edu.au'>XXXXXXXX</a></li>
      <li><img src='smiley.gif' alt='smiley'/></li>
      <li>&lt;entities&gt;</li>
    </ol>
  </body>
</html>
```

<ol>

*
This is bold yeah!
*

<tr><th>Name</th><th>Subject</th><th>Score</th></tr>

link to happiness

1.
   - This is **bold** yeah!
   - underline
   - *italic*
2.

Name	Subject	Score
Lawnmower full of rizz	COMP10001	76
Slayed the CompRPG game	COMP10002	81

3. link to happiness
4. 
5. <entities>

https://www.w3schools.com/html/tryit.asp?filename=tryhtml_default

W12 – GenAI Worksheet

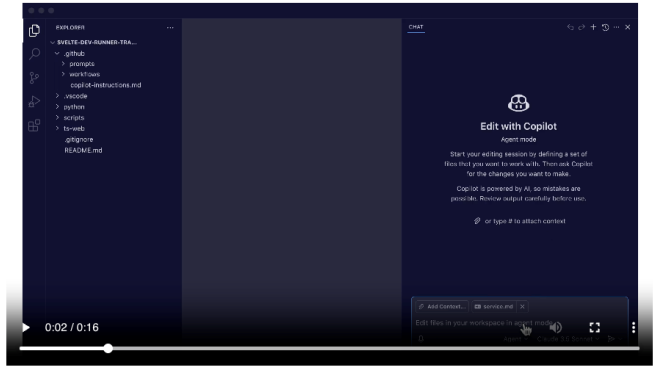
W12 - GenAI Worksheet (OPTIONAL; NOT FOR MARKS)

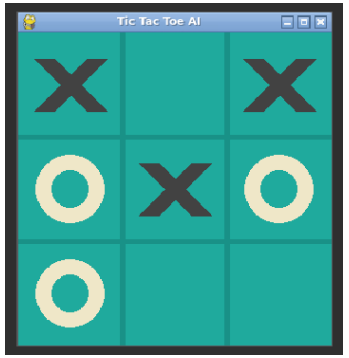
- GitHub Copilot Exercises
  Available: Mon October 20th, 12:00am
- Setup & Function Design Cycle
  Available: Mon October 20th, 12:00am
- Problem Decomposition
  Available: Mon October 20th, 12:00am
- Advanced - Game & Webpage
  Available: Mon October 20th, 12:00am
- Survey
  Available: Mon October 20th, 12:00am

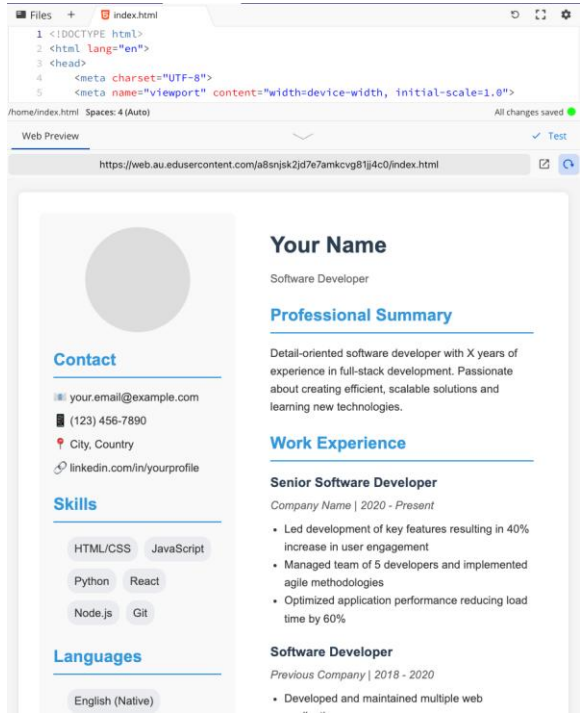
Setup - GitHub Copilot in VS Code

🚀 Getting Started with GitHub Copilot in VS Code

Now let's move from the Ed environment to your local setup by installing **Visual Studio Code (VS Code)**. Here's how to get started, step by step:



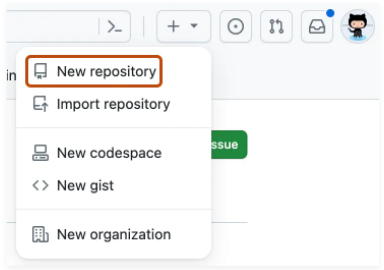




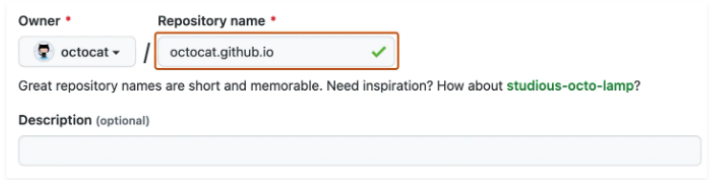
W12 – GenAI Worksheet

Creating your website

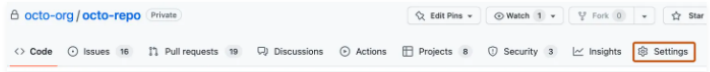
1. In the upper-right corner of any page, select +, then click **New repository**.



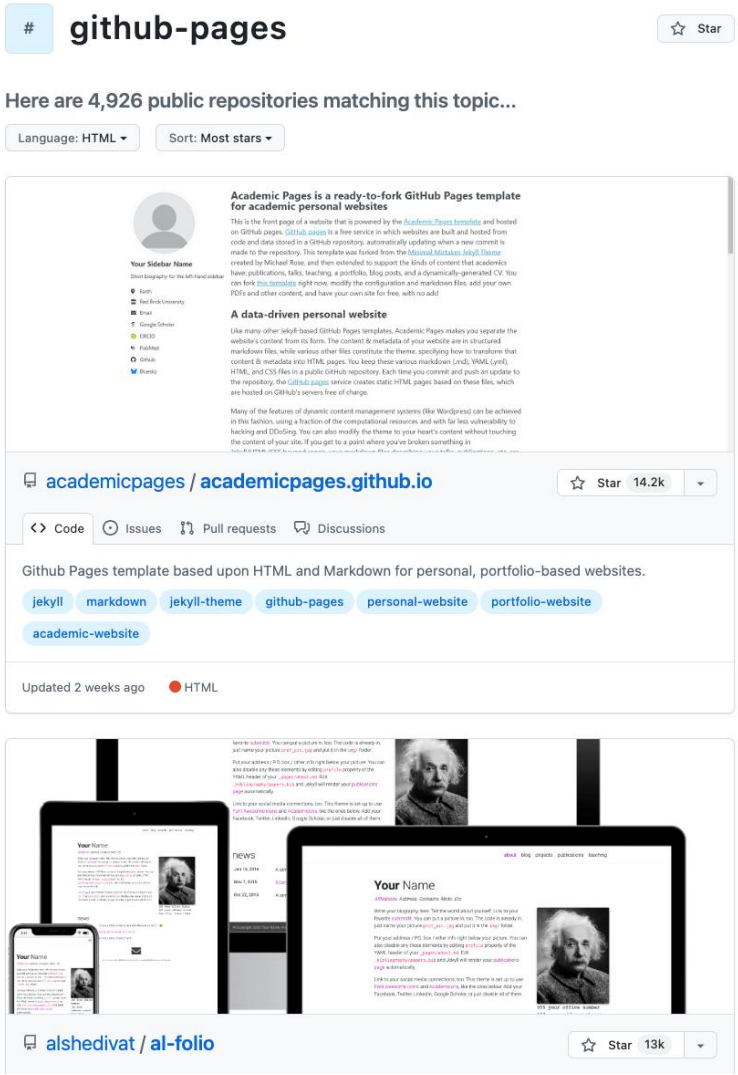
2. Enter `username.github.io` as the repository name. Replace `username` with your GitHub username. For example, if your username is `octocat`, the repository name should be `octocat.github.io`.



3. Choose a repository visibility. For more information, see About repositories.
4. Select **Initialize this repository with a README**.
5. Click **Create repository**.
6. Under your repository name, click ⚙ **Settings**. If you cannot see the "Settings" tab, select the ··· dropdown menu, then click **Settings**.



7. In the "Code and automation" section of the sidebar, click **Pages**.
8. Under "Build and deployment", under "Source", select **Deploy from a branch**.
9. Under "Build and deployment", under "Branch", use the branch dropdown menu and select a publishing source.



https://docs.github.com/en/pages/quickstart
https://github.com/topics/github-pages

That's it, thank you for the Semester! ^_^

- **Final due date:**
  - Final exam worth **50%** of your grade on **24 June, 12.30pm**.
    - Please see **my.unimelb.edu.au** for more details.
  - **If you don't sit the exam**, you will **FAIL** the subject due to the hurdle requirement **(30/60 for the MST + Final Exam combined)**.
  - All the best with your exam(s) and future endeavors!

Scan here for annotated slides