

Web Dev + Devops Cohort by 100xDevs

Goal

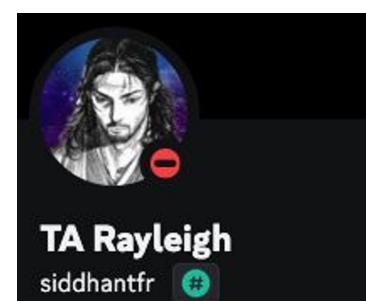
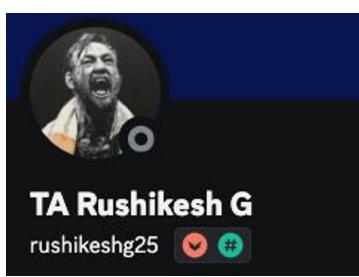
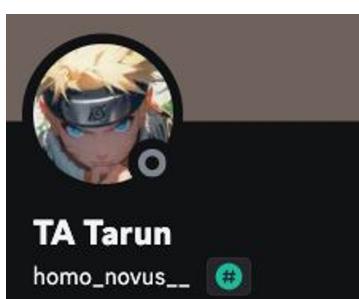
Make you incredibly solid at building and deploying Web apps.

Syllabus

Web Dev - <https://blog.100xdevs.com/Web-dev-6e1c515142e14347ae174c55e6c953c6>

Devops - <https://blog.100xdevs.com/Devops-39f1c2ef45e741e79b1f09f8793d6ad8?pvs=25>

TAs

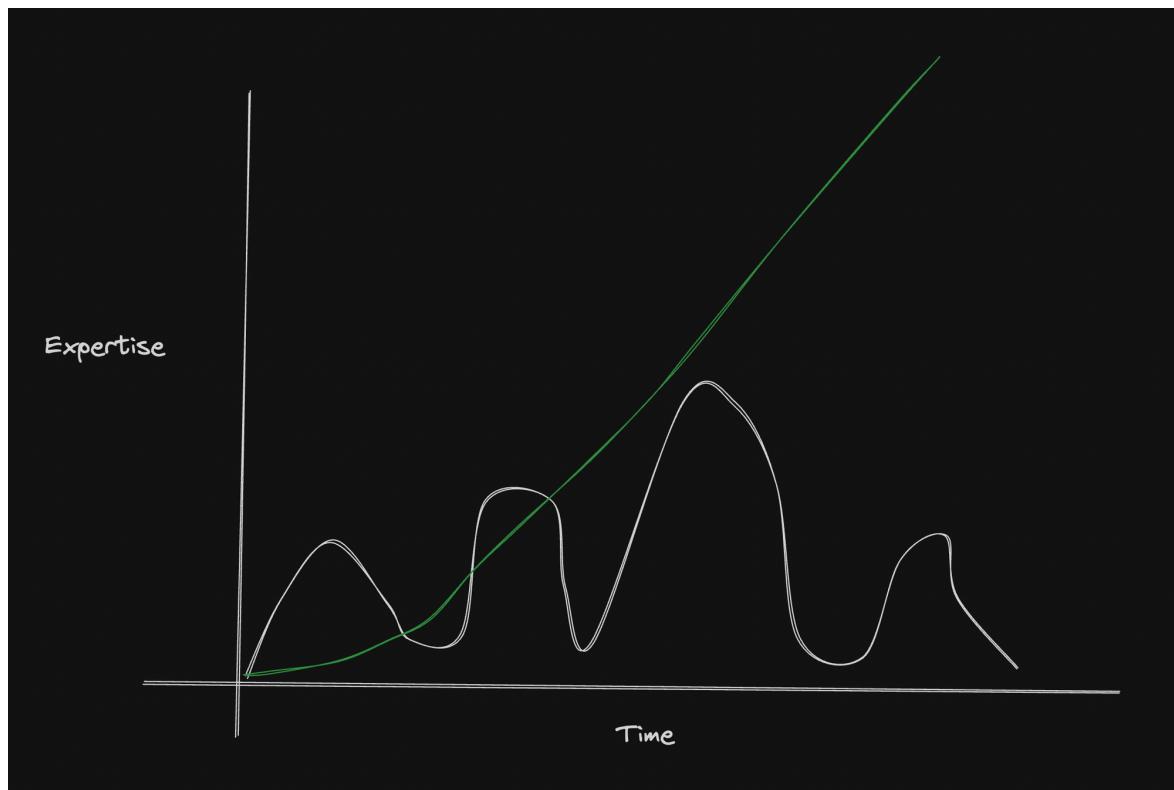


Cohort Projects

1. <https://github.com/code100x/cms/> - Led by hsingh
2. <https://github.com/code100x/daily-code/> - Led by Sargam
3. <https://github.com/code100x/job-board> - Led by Vineet
4. <https://github.com/code100x/algorithmic-arena> - Led by Nimit
5. Framer like app - Up for grabs

Avoid the spikes

1. We will go slow, today only focussing on HTML/CSS/Javascript
2. Any topic that you feel is missing / you want we will do polls from time to time we can add
3. Go slow but stay consistent, avoid the spikes



What would I do if I was you?

1. Get to know people in discord, try to find a group of 3-4 motivated people.

2. Look at all the success stories here and how they got a job - <https://www.youtube.com/@100xDevs-n1w>
3. Repeated advise - GSoC is the best internship you can do. Already start prepping for it competition is high. Tell me how can I help here we're hoping for ~50 GSoC selections from the cohort this year. Good video to watch - <https://www.youtube.com/watch?v=OmzCvb-QBak>



We'll be covering things slowly in Cohort 3, so if you think things are going too slow you can

1. Participate in projects/hackathons
2. Sprint through Cohort 2.0

Select your level

Complete beginner (very basic/no coding before)

Things will be hard. You should spend ~20 hours/week atleast on Cohort material, assignments, other content covering what we're doing in the Cohort. Don't slack off

Intermediate (Done basic JS, HTML and some DSA)

You should be for the first ~10 weeks or so. We see major drop off after the first major project.

Try chewing some glass after week 10, that is where you will actually learn things that differentiate you from a typical MERN developer

Advance (Know MERN decent/well)

Feel free to drop off for the first few weeks. I'll see you on Github. Participate in the Superteam hackathon , aim for GSoC this year. Already start looking at orgs

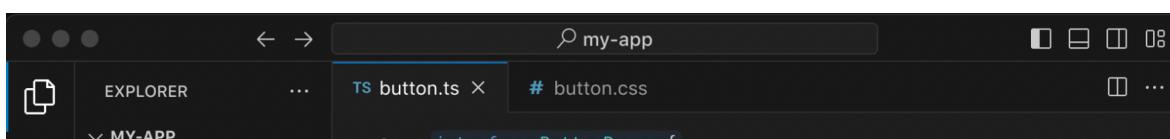
Good things to hash out

1. Installing VSCode locally
2. HTML/CSS
3. Cloning the landing page of a simple website

VSCode Installation

Download from here - <https://code.visualstudio.com/>

When you open it locally, you should see something like this



The screenshot shows a code editor interface with a sidebar containing project files like components, actionbar, breadcrumbs, button, button.css, button.ts, countBadge, dialog, dropdown, findinput, grid, hover, inputBox, .gitignore, .mailmap, .mention-bot, .yarnrc, and yarn.lock. The main area displays a TypeScript file named button.ts with the following code:

```
1  interface ButtonProps {
2    onClick: () => void;
3    text: string;
4  }
5
6  const Button: React.FC<Props> = ({ onClick, text }) => {
7    return <button onClick={onClick}>{text}</button>;
8  };
9
10 export default Button;
```

A modal window titled "Create a new button component" is open, containing two buttons: "Accept" and "Discard". Below the buttons, it says "Changed 9 lines". At the bottom of the editor, there are tabs for PROBLEMS, OUTPUT, TERMINAL, and other icons. The TERMINAL tab is active, showing logs from monaco.d.ts generation and compilation:

```
[09:44:50] [monaco.d.ts] Starting monaco.d.ts generation
[09:44:56] [monaco.d.ts] Finished monaco.d.ts generation
[09:44:56] Finished compilation with 557 errors after 80542 ms
```

The status bar at the bottom shows "my-app git:(main) []" and "TypeScript".

HTML

HTML stands for hyper text markup language. It is the foundation of building **unstyled** websites.

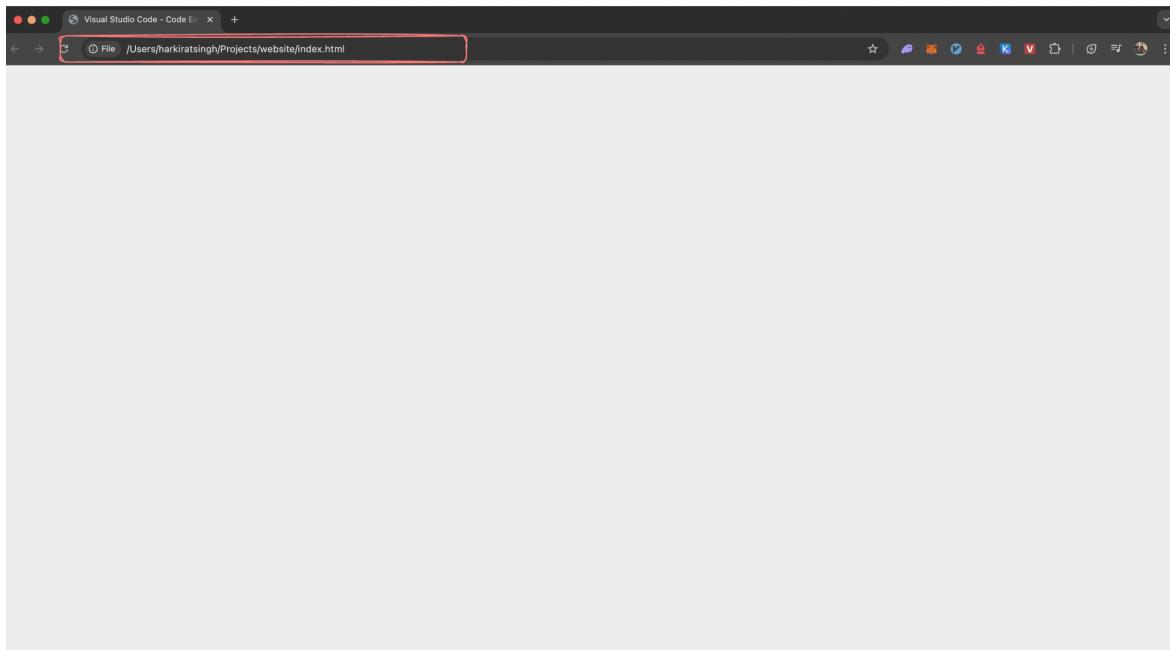
Step 1 - Creating a simple HTML file locally

1. Create a folder on your machine called **website**
2. Add a **index.html** file inside it
3. Write the following code in it -

```
<html>
    <title>
        Visual Studio Code - Code Editor
    </title>
</html>
```

Copy

Now open this in your browser



Step 2 - Tags

1. div, span
2. head
3. body
4. h1, h2, h3, h4, h5, h6
5. b, i, u
6. a
7. img
8. input
9. button
10. br

Step 3 - Building a basic HTML page

```
<html>
    <title>
        Visual Studio Code - Code Editor
    </title>
</html>
<body>
    <div>
        <span>Visual Studio Code</span>
        <a href="/">Docs</span>
        <a href="/">Updates</span>
        <a href="/">Blog</span>
        <a href="/">API</span>
        <a href="/">Extensions</span>
        <a href="/">FAQs</span>
        <a href="/">Learn</span>

        <input type="text" placeholder="Search Docs">
        <button>Download</button>
    </div>
    <br/>

    <div>
        <a href="/">Version 1.82</a> is now available! Read about the new features
    </div>

    <br/>
</body>
```

Visual Studio Code [Docs](#) [Updates](#) [Blog](#) [API](#) [Extensions](#) [FAQs](#) [Learn](#) [Download](#)

[Version 1.82](#) is now available! Read about the new features and fixes from July.

CSS

CSS stands for Cascading Style Sheets. It is used to style our applications

You can add CSS to your HTML app by using -

1. The `style` attribute (inline styles)
2. In an external css file

Approach #1 - Inline styles

Try updating the `body` tag in the last style as follows -

```
<body style="background-color: black;">  
... rest of the code  
</body>
```

[Copy](#)

Approach #2 - External styles

1. Add a new file called index.css
2. Add the following code in it

```
body {  
    background-color: black;  
}
```

[Copy](#)

- Update index.html

```
<html>  
    <title>  
        Visual Studio Code - Code Editor  
    </title>  
    <link rel="stylesheet" href="index.css">  
</html>  
<body>  
    ... rest of the code  
</body>
```

[Copy](#)



Common style attributes

- `color` : Sets the text color.
- `background-color` : Sets the background color.
- `font-size` : Sets the size of the text.
- `margin` : Sets the outer space around an element.
- `padding` : Sets the inner space within an element.
- `border` : Sets the border around an element.

Flexbox

Flexbox is a CSS layout model designed to help with the arrangement of items within a container.

Update the website to the following -

```
<html>
  <title>
    Visual Studio Code - Code Editor
  </title>
</html>
<body>
  <div style="display: flex;">
    <div>Visual Studio Code</div>
    <a href="/">Docs</span>
    <a href="/">Updates</span>
    <a href="/">Blog</span>
    <a href="/">API</span>
    <a href="/">Extensions</span>
    <a href="/">FAQs</span>
    <a href="/">Learn</span>
  </div>
  <div>
    <input type="text" placeholder="Search Docs">
    <button>Download</button>
  </div>
  <br/>

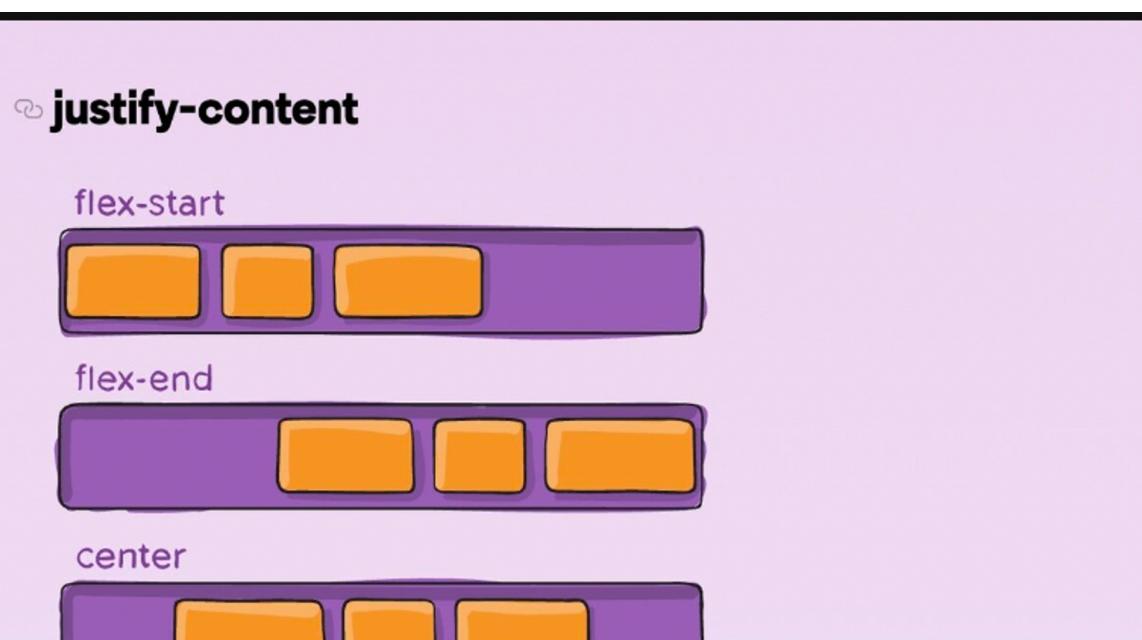
  <div>
    <a href="/">Version 1.82</a> is now available! Read about the new features
  </div>

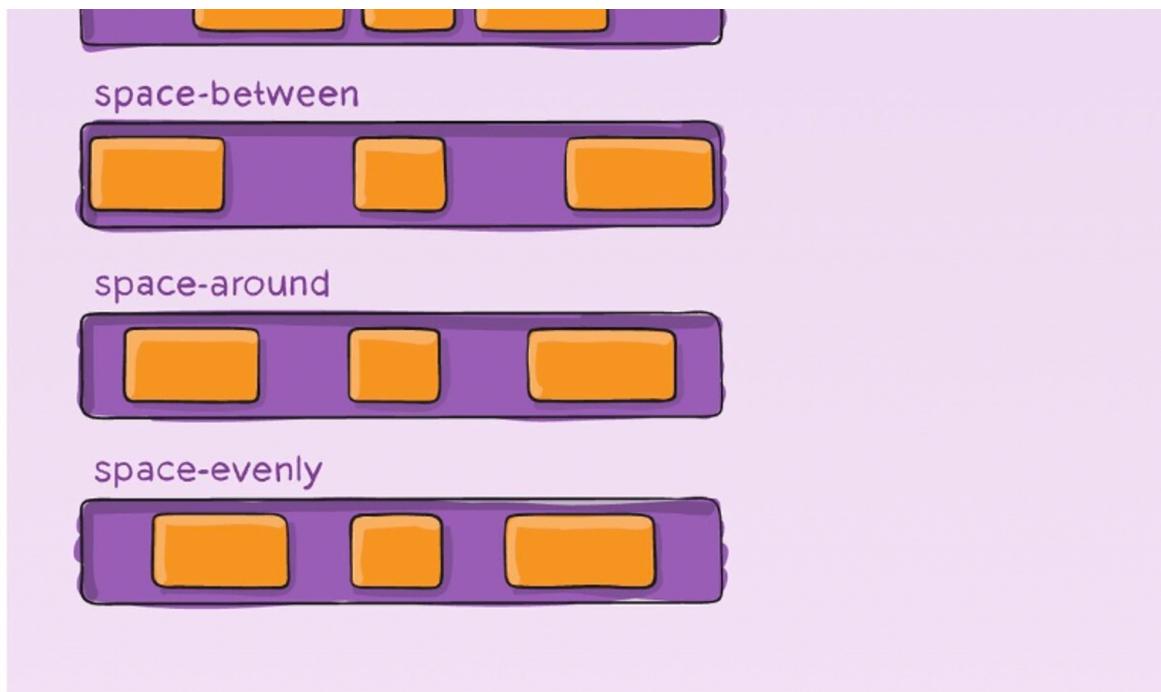
  <br/>
</body>
```

Notice that the elements are positioned right next to each other even though `Visual Studio code` is inside a `div`

Justify content

Try experimenting with the `justify-content` property





```
<html>
    <title>
        Visual Studio Code - Code Editor
    </title>
</html>
<body>
    <div style="display: flex; justify-content: space-between;">
        <div>Visual Studio Code</div>
        <a href="/">Docs</span>
        <a href="/">Updates</span>
        <a href="/">Blog</span>
        <a href="/">API</span>
        <a href="/">Extensions</span>
        <a href="/">FAQs</span>
        <a href="/">Learn</span>
    </div>
    <div>
        <input type="text" placeholder="Search Docs">
        <button>Download</button>
    </div>
    <br/>

    <div>
        <a href="/">Version 1.82</a> is now available! Read about the new features
    </div>

    <br/>
</body>
```

Another example

```
<html>  
  </html>  
  <body>  
    <header>  
  
      </header>  
      <section>  
        <div style="border-width: thick; border-style: solid; display: flex; justify-content: center; align-items: center; height: 100px; width: 100px; background-color: red; margin-bottom: 10px;">  
          <div style="background: red; padding: 10px; text-align: center; color: white; font-weight: bold; font-size: 1.2em;">  
            Code with GitHub Copilot  
          </div>  
        </div>  
        <h6>  
          Write code faster and smarter with GitHub Copilot, your AI pair program.  
        </h6>  
        <p>Try GitHub Copilot free for 30 days</p>  
        <p>Completions present suggestions automatically to help you code more efficiently.</p>  
  
        <p>Copilot Chat understands the context of your code, workspace, extensions, and environment to suggest completions and answers to your questions. It's available in the GitHub Copilot extension for VS Code, and as a web-based service at https://copilot.github.com/</p>  
        <div style="background: green; padding: 10px; margin-top: 10px;">  
            
        </div>  
      </section>  
      <footer>  
        </footer>  
    </body>
```

Classes and ids

In CSS, classes and IDs are used as selectors to apply styles to HTML elements. They help in targeting specific elements for styling and can be used to enhance the modularity and reusability of CSS code.