

While folks are joining

Get your laptops ready and keep the Sandbox opened. We will be coding away in the session!

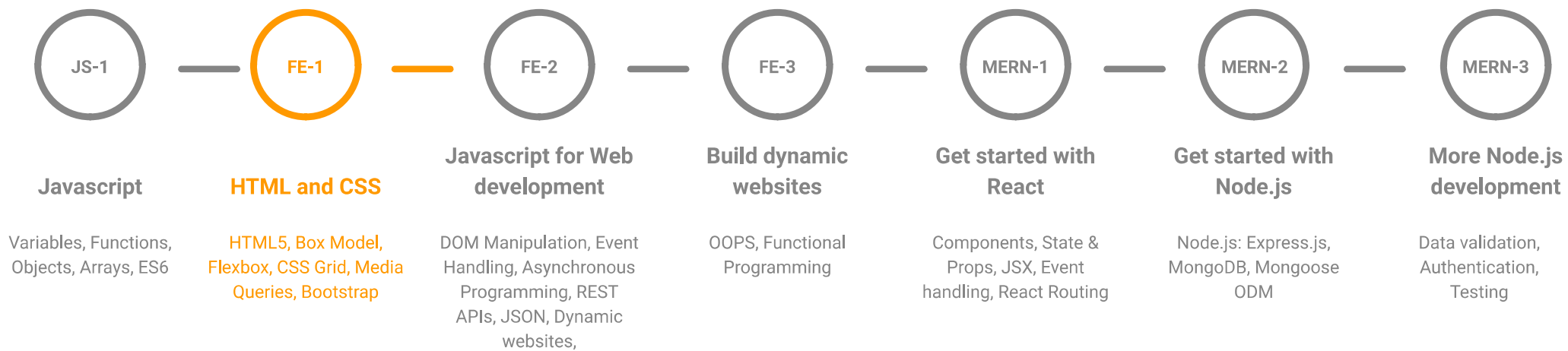


Crio Fullstack Sprint: FE-1

Session 1

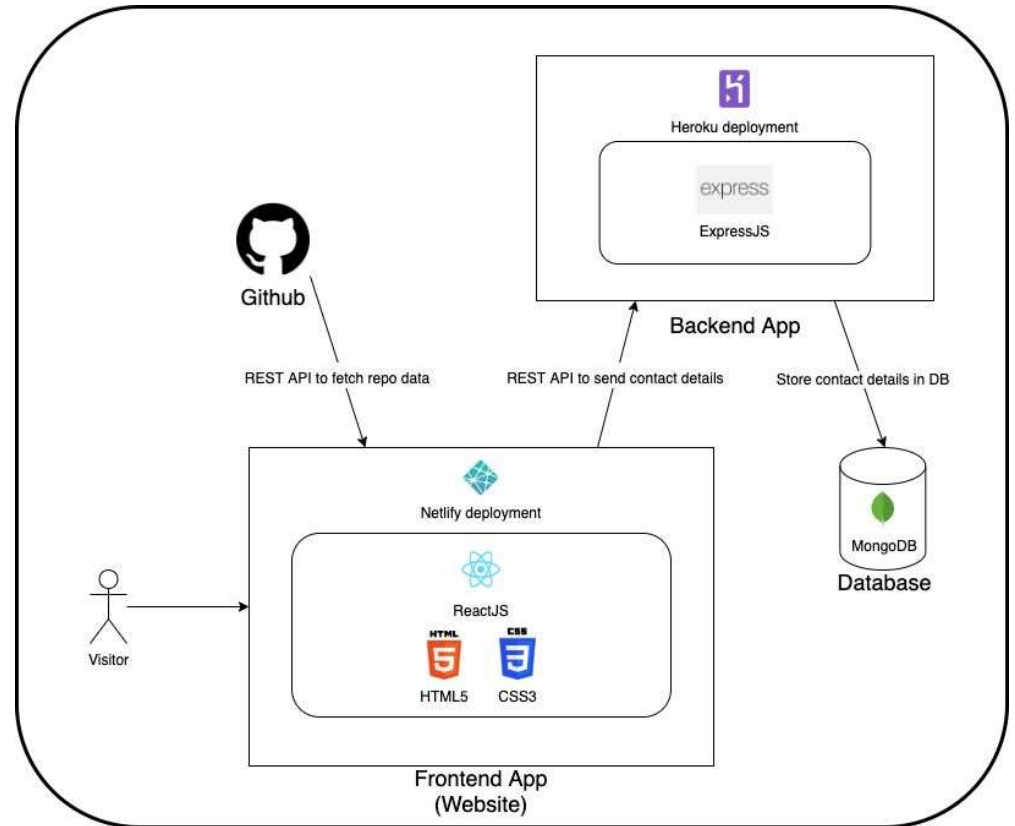


Crio Fullstack Development Track - Journey



Full-stack Dev overview

- Creating a front-end website for end-users.
- Creating a database to store the data for the frontend.
- Creating an application to connect front-end to database.
- Deploying the website/web-app to cloud servers.



What we'll learn in this sprint?

- Introduction to web development
- How the web works?
- Introduction to HTML
- Building Pages with plain HTML
- Adding styling to pages with CSS
- Box model, Flexbox, and Grid
- Responsive web pages
- Building a static website



In this session

- Intro to web development
- How the web works?
- Introduction to HTML
- HTML Elements & Tags
- Heading and Paragraphs
- Lists
- HTML Attributes



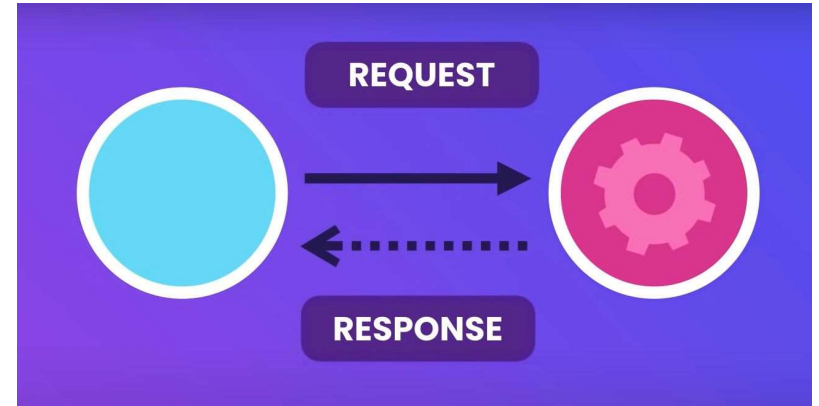
Intro to web development

- Two broad divisions - Client-side and Server-side
- Client = Your browser
- Server = Other computer where the resources live.
- Front-End = Whatever displayed on the screen!
 - HTML (Hypertext Markup Language) - The Structure
 - CSS (Cascading Style Sheets) - the styling
 - JS (JavaScript) - the functionality



How the web works? (Recap)

1. You type a URL in your browser.
2. Browser sends a message to a server.
3. Server understands the message.
4. Server sends a message back to your browser.
5. Browser reads the message
6. If there is any other information needs, it agains asks the server for it
7. This process is repeated several times till the browser has complete info.
8. Browser renders a web page on screen



Activity 1 - Let's talk to the server

Type in the url -

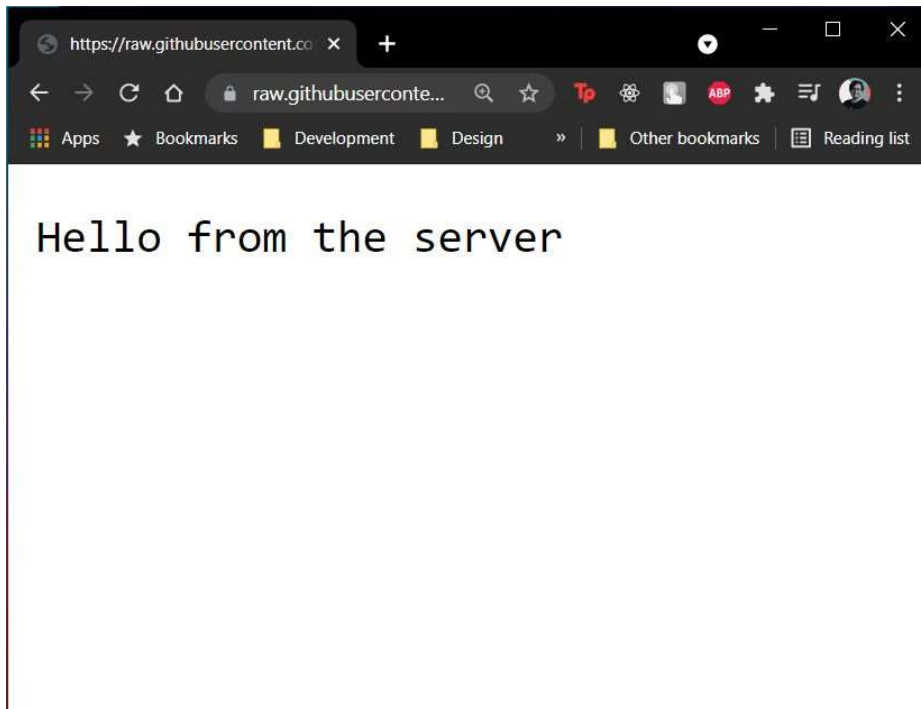
<https://raw.githubusercontent.com/viveknigam3003/crio-assets/main/file.txt> in your browser.

What do you see?

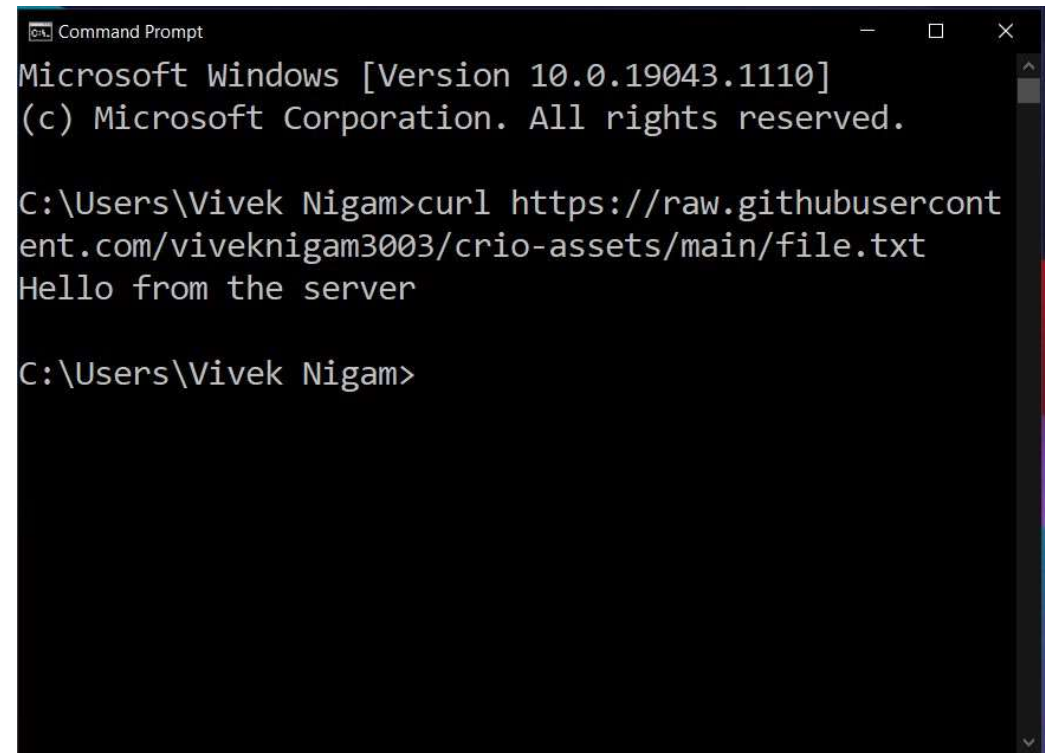
Next, open terminal or powershell or cmd and type `curl <above-url>` and see what happens?



Activity 1 - Let's talk to the server



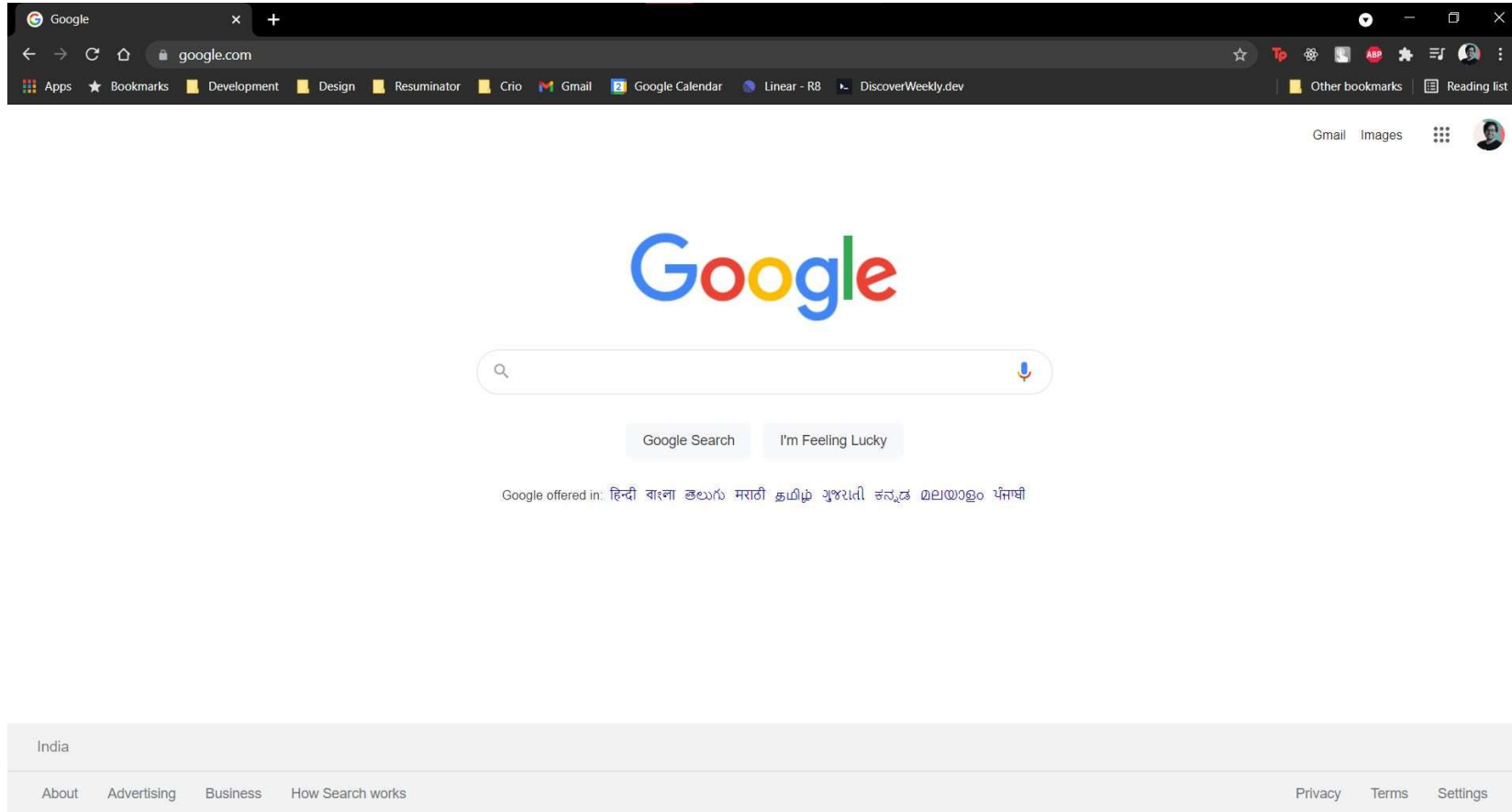
On the browser



On the cmd prompt



Activity 2 - What is a web page built of?



Activity 2 - What all can you see on this page?

The image shows a screenshot of the Google homepage in a web browser. A UI element inspector overlay is visible on the left side, listing the following elements:

- Elements
- Images
- Links
- Inputs
- Buttons

The Google homepage content includes:

- Google logo
- Search bar with a magnifying glass icon and a voice search icon
- Buttons: "Google Search" and "I'm Feeling Lucky"
- Text: "Google offered in:" followed by a list of languages: हिन्दी, বাংলা, తెలుగు, मराठी, தமிழ், ગુજરાતી, ಕನ್ನಡ, മലയാളം, ਪੰਜਾਬੀ
- Footer: "India" (with a dropdown arrow), "About", "Advertising", "Business", "How Search works", "Privacy", "Terms", "Settings"

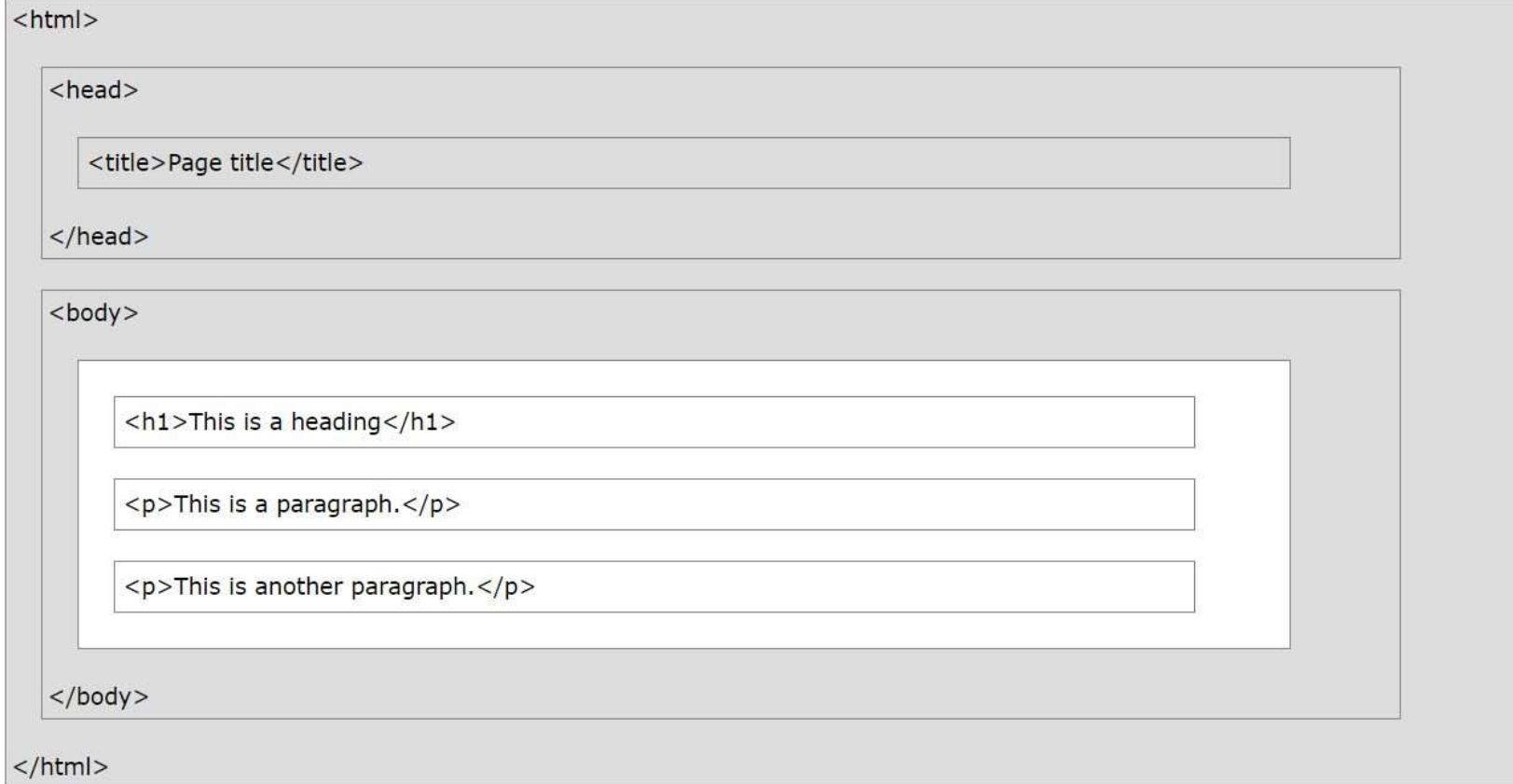


What is HTML?

- Stands for HyperText Markup Language
- Standard markup language for creating web pages
- Responsible for structure of website/webpage
- Made up of elements -> which tell the browser what and how to display
- Contains all information like
 - Structure (Elements)
 - Content
 - Styles
 - Fonts etc...



Structure of a page



HTML Elements & Tags

- Everything in HTML is made up of elements
- Element = Start tag (`<tag>`) + Content (text) + End Tag (`</tag>`)
- Tag: Tells browser the type of the content in the element - image, input, button, text?
- Start Tags: `<tagName>`, End Tags: `</tagName>`
- There are a bunch of tags in HTML - some mandatory, some optional.



Structure of a page

Code

```
<!DOCTYPE html>  
<html>  
  <head></head>  
  <body>  
  
    </body>  
</html>
```

The <html>, <head>, and <body> are known as Tags

Best practice - Start your HTML files with a DOCTYPE declaration to tell the browser which kind of document to expect.



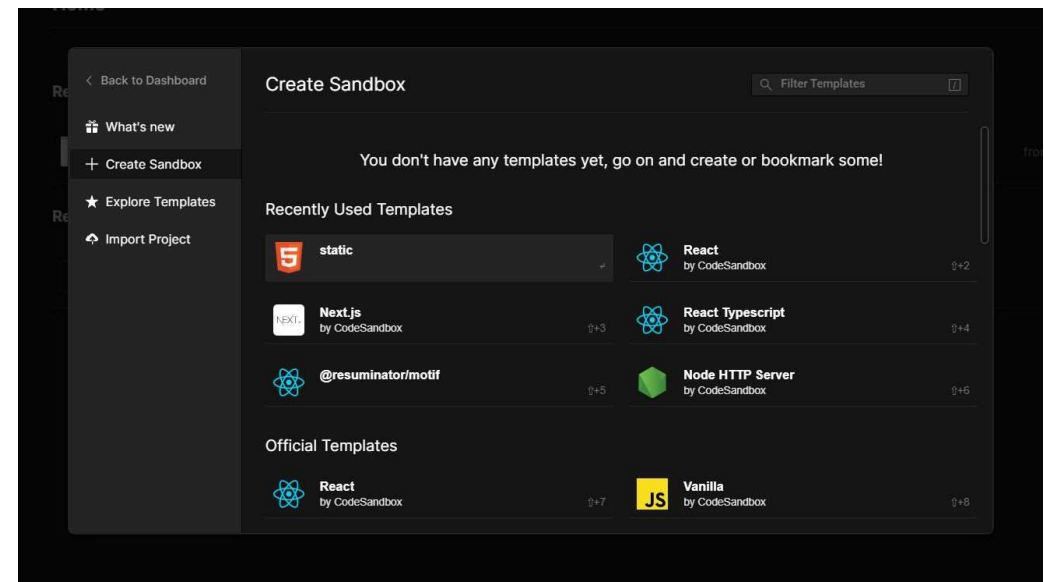
`<html>`, `<head>`, & `<body>`

- Every HTML page will (generally) have the html, head, and body tag.
- `<html>` tag encloses all the page content
 - Hence it is also called the root element.
- `<head>` tag defines the metadata (data about data) of the page
 - Like title (which is displayed on browser tab) using the `<title>` tag
 - Fonts for the page
 - Styles etc.
- `<body>` tag contains the content which is displayed inside the browser window as a page.



Activity 3 - Creating your first HTML Page

- Create a new “static” sandbox on Codesandbox.
- Create a file “activity3.html” inside a new folder, “session1”
- Add a html, head, and body tag and close them appropriately.
- Write some text inside the body tag like - “First HTML Page”
- Save the file to see the results
- 🍰 Cherry points - Give a title to your page - **Cute Webpage**.
Hint - Use the <title> tag



Headings in HTML `<h1>` to `<h6>`

Contribution Guide

Thanks a lot for your interest in contributing to Resuminator! It's because of developers and designers like you that Resuminator thrives to become a better product for our community. 🌟

Resuminator shall be ready soon to accept contributions freely from open-source contributors like you but till then you can follow below guidelines to keep Resuminator clutter free.

In the meantime, if you're really interested to help build this product along with us or wish to be a part of it early on in its development - email us at hello@resuminator.in and we can discuss how we can work together on this. 😊

Reporting Bugs

This section guides you through submitting a bug report for Resuminator. Following these guidelines helps maintainers and the community understand your report 📄, reproduce the behavior 🖥️ 🖥️, and find related reports 🔍

Before creating bug reports, please check the existing issues as you might find out that you don't need to create one. When you are creating a bug report, please include as many details as possible. Fill out [the required template](#), the information it asks for helps us resolve issues faster.

Reporting Bugs

[How Do I Submit A \(Good\) Bug Report?](#)

Suggesting Enhancements

[How Do I Submit A \(Good\) Enhancement Suggestion?](#)

Your First Code Contribution

Pull Requests

Suggestion And Discussions



Headings in HTML `<h1>` to `<h6>`

Contribution Guide H1

Thanks a lot for your interest in contributing to Resonator! The team of developers and designers like you that Resonator strives to become a better product for our community. 🐾

Resonator shall be really keen to accept contributions from those who contribute like you but all that you can follow some guidelines to keep Resonator better than.

In this meantime, if you're really interested to help build this product along with us or wish to be a part of it, apply as a developer. Email us at info@resonator.in and we will discuss how we can work together on this. 🐾

Reporting Bugs H2

This section guides you through submitting a bug report for Resonator. Following these guidelines helps maintainers and the community understand your report 🐾, reproduce the behavior 🐾, and find related reports 🐾.

Before creating bug reports, please check the existing issues as you might find out that you don't need to create one. When you are creating a bug report, please include as many details as possible. Fill out [this required template](#), the information it asks for helps us resolve issues faster.

- H2 → [Reporting Bugs](#)
- H3 → [How Do I Submit A \(Good\) Bug Report?](#)
- [Suggesting Enhancements](#)
- [How Do I Submit A \(Good\) Enhancement Suggestion?](#)
- [Your First Code Contribution](#)
- [Pull Requests](#)
- [Suggestion And Discussions](#)



<h1> to <h6>

- Headings help in emphasizing a section of text on a page
- Sections can be separated by emphasis using different heading sizes.
- Largest heading \Rightarrow most important section
- Smallest heading \Rightarrow least important section
- HTML Headings are created with h tags
 - H1 (<h1>) \rightarrow Largest most important heading
 - H6 (<h6>) \rightarrow Smallest, least important heading

H1 - Create the best documentation

H2 - Create the best documentation

H3 - Create the best documentation

H4 - Create the best documentation

H5 - Create the best documentation

H6 - Create the best documentation



5 mins break!

<p> &

- Most HTML pages have blocks of text a.k.a. paragraphs.
- To add paragraphs of text to our page, we use <p> tags
- Browsers add some (vertical) whitespace before and after the paragraph block

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

```
<p>Another lengthy paragraph can be here</p>
```

- A paragraph starts on a new line by default.
- If you want to start text from new line, BUT not change the paragraph, we use a
 or break tag.

```
<p>Another lengthy paragraph can be here.<br />
```

```
This will be in the same paragraph but new line</p>
```



HTML Lists

Look at the following screen snippet

Assignment

1. Read [Shay Howe's Introduction to Terminology & Syntax](#)
2. Read [Shay Howe on the HTML Basics](#)
3. Read [Shay Howe on Best Practices](#)



HTML Lists

- Often we need to provide a list on a page - for content or links
- Two types of list
 - Unordered or Bulleted Lists created by ``
 - Ordered or Numbered Lists created by ``
- Each item under the list must be enclosed in its own `` tag to denote it as a list item.

```
<ul>
  <li>HTML</li>
  <li>CSS</li>
  <li>JS</li>
</ul>
<ol>
  <li>Step 1</li>
  <li>Step 2</li>
  <li>Step 3</li>
</ol>
```

- 
- HTML
 - CSS
 - JS

1. Step 1
2. Step 2
3. Step 3



Activity 4 - Adding Lists

- Add a heading - **Some cool features of Cupcake Ipsum**
- Create a bulleted list of these features - It's fun to use, Easy to create, Makes you look cool
- Add another heading - **How to generate it?**
- Create a numbered list of the steps - Go to the website, Select 1 paragraph, Select "short" option, check both the boxes, click generate, copy to clipboard

Some cool features of Cupcake Ipsum

- It's fun to use
- Easy to create
- Make you look cool

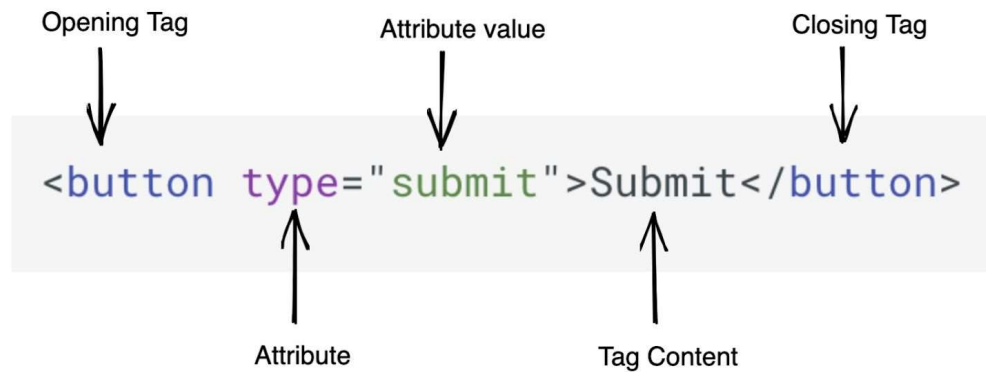
How to generate it?

1. Go to the website
2. Select the options
3. Copy to clipboard.
4. Profit!



HTML Attributes

- Attributes provides additional information about elements
- Always specified in the start tag
- Attributes usually come in name/value pairs like: name="value"



Summary

- Web development - includes client-side and server-side development
- Frontend development (client-side)
 - HTML - Structure
 - CSS - Style
 - JS - Functionality
- Browser (client) communicates with the server on your behalf
 - HTTP is the language of communication
 - Browser sends a request each for any extra information/resource it needs



Summary

- HTML defines the content of your webpage eg: images, buttons, links
- Basic HTML structure

```
<!DOCTYPE html>
<html>
  <head></head>
  <body>

  </body>
</html>
```

- The `<!DOCTYPE html>` tag lets the browser know it's HTML5 format
- `<h1>`, `<h2>`, ..., `<h6>` tags are used for headings with decreasing font-size
- `<p>` tag is for paragraph content and `
` for adding a line break



Takehome Activity - Adding content to web page

- Go to <http://www.cupcakeipsum.com> and generate a filler text of 1 paragraph from here.
- Use the heading tags to add a heading - **Cupcake Ipsum**
- Add a subheading - **The cutest lorem ipsum filler**
- Add the paragraphs generated from the website.
- Save the website to see the results
- Final result should look like this →

Cupcake Ipsum

The cutest lorem ipsum

Cupcake ipsum dolor sit amet cotton candy tootsie roll pudding. Sesame snaps caramels sweet roll marshmallow tiramisu. Donut

I love lemon drops donut cake ice cream pastry. Cookie I love gummi bears I love sugar plum tootsie roll.



To watch out for!

- Takehomes implementation exercises doesn't need to be submitted anywhere
- Each of the session Takehomes also include a Quiz - it's Mandatory!



Note for take homes

- Please finish the take home of a session before the next session
→ Because we use a lot of the code from it as a starter code for next session.
- Take home commitment ~2-3 hours.



Until next session 

Thank you for joining in today, we'd love to hear your thoughts and feedback



Thank you

