Introduction:
Front-End & HTML

Introduction:

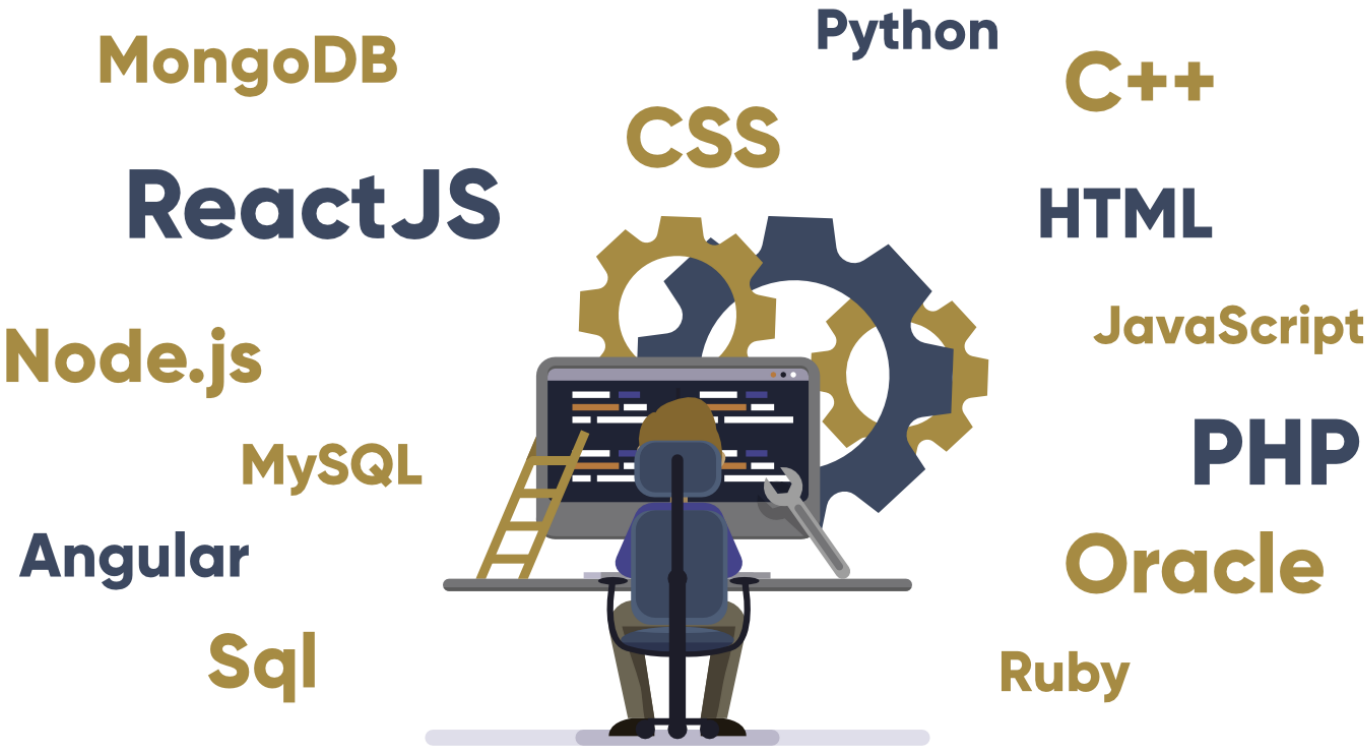
A Full-Stack developer is someone who can do both front end and back end. Front end interacts directly with the end users; this is the main purpose. Back end enables front end to work properly accordingly.

All resources required to render everything we see on a web page comes from the AttainU server. When we refresh, we can check in the inspect option, all the requests and servers that we need to render or view a particular web page.

We need to separate the two (front-end and back-end), so that we separate the business logic and protect it from others.

Full stack development: It refers to the development of both **front end** (client side) and **back end** (server side) portions of web application.

Full stack web Developers: Full stack web developers have the ability to design complete web application and websites. They work on the frontend, backend, database and debugging of web application or websites.



Technology related to full stack development:

Front End: It is the visible part of website or web application which is responsible for user experience. The user directly interacts with the front end portion of the web application or website.
Front End Languages: The front end portion is built by using some languages which are discussed below:

- **HTML:** HTML stands for Hyper Text Markup Language. It is used to design the front-end portion of web pages using markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. The markup language is used to define the text documentation within tag which defines the structure of web pages.
- **CSS:** Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.

- **JavaScript:** JavaScript is a famous scripting language used to create the magic on the sites to make the site interactive for the user. It is used to enhancing the functionality of a website to running cool games and web-based software.

Front End Frameworks and Libraries:

- AngularJS**:** AngularJs is a JavaScript open-source front-end framework that is mainly used to develop single page web applications (SPAs). It is a continuously growing and expanding framework which provides better ways for developing web applications. It changes the static HTML to dynamic HTML. It is an open-source project which can be freely used and changed by anyone. It extends HTML attributes with Directives, and data is bound with HTML.
- React.js**:** React is a declarative, efficient, and flexible JavaScript library for building user interfaces. ReactJS is an open-source, component-based front-end library responsible only for the view layer of the application. It is maintained by Facebook.
- Bootstrap**:** Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first web sites.
- jQuery**:** jQuery is an open-source JavaScript library that simplifies the interactions between an HTML/CSS document, or more precisely the Document Object Model (DOM), and JavaScript. Elaborating the terms, jQuery simplifies HTML document traversing and manipulation, browser event handling, DOM animations, Ajax interactions, and cross-browser JavaScript development.
- **SASS:** It is the most reliable, mature and robust CSS extension language. It is used to extend the functionality of an existing CSS of a site including everything from variables, inheritance, and nesting with ease.

- Some other libraries and frameworks are: Semantic-UI, Foundation, Materialize, Backbone.js, Express.js, Ember.js etc.

Other Important Points:

- Work with text editors to use shortcuts and its facilities i.e. Visual studio, Atom, Sublime etc.
- Make UI responsible using grid system.
- Git and git commands like init, add, commit etc for version control and to work with team.
- Other tools like npm & yarn package managers, sass css pre-processor, browser DevTools i.e. chrome devtools.
- Understand using HTTP, JSON, GraphQL APIs to fetch data using axios or other tools.
- It also requires some design skill to make layout and look better.

Back end: It refers to the server-side development of web application or website with a primary focus on how the website works. It is responsible for managing the database through queries and APIs by client-side commands. This type of website mainly consists of three parts front end, back end, and database.

The back end portion is built by using some libraries, frameworks, and languages which are discussed below:

- **PHP:** PHP is a server-side scripting language designed specifically for web development. Since, PHP code executed on server side so it is called server side scripting language.
- **C++** It is a general purpose programming language and widely used now a days for competitive programming. It is also used as backend language.
- **Java:** Java is one of the most popular and widely used programming language and platform. It is highly scalable. Java components are easily available.

- **Python:** Python is a programming language that lets you work quickly and integrate systems more efficiently.
- **JavaScript:** Javascript can be used as both (front end and back end) programming languages.
- Node.js**:** Node.js is an open source and cross-platform runtime environment for executing JavaScript code outside of a browser. You need to remember that NodeJS is not a framework and it's not a programming language. Most of the people are confused and understand it's a framework or a programming language. We often use Node.js for building back-end services like APIs like Web App or Mobile App. It's used in production by large companies such as Paypal, Uber, Netflix, Wallmart and so on.
- **Back End Frameworks:** The list of back end frameworks are: Express, Django, Rails, Laravel, Spring etc.
- The other back end program/scripting languages are: C#, Ruby, REST, GO etc.

Other Important Points:

- Structuring the data in efficient way.
- Handle request-response of APIs for storing and retrieve data.
- Security of data is important.

Note: JavaScript is essential for all stacks as it is dominant technology on Web.

Database**:** Database is the collection of inter-related data which helps in efficient retrieval, insertion and deletion of data from database and organizes the data in the form of tables, views, schemas, reports etc.

- **Oracle:** Oracle database is the collection of data which is treated as a unit. The purpose of this database is to store and retrieve information related to the query. It is a database server and used to manages information.
- MongoDB**:** MongoDB, the most popular NoSQL database, is an open source document-oriented database. The term 'NoSQL' means 'non-

relational'. It means that MongoDB isn't based on the table-like relational database structure but provides an altogether different mechanism for storage and retrieval of data.

- Sql**:** Structured Query Language is a standard Database language which is used to create, maintain and retrieve the relational database.

Popular Stacks:

- **MEAN Stack:** MongoDB, Express, AngularJS and Node.js.
- **MERN Stack:** MongoDB, Express, ReactJS and Node.js
- **Django Stack:** Django, python and MySQL as Database.
- **Rails or Ruby on Rails:** Uses Ruby, PHP and MySQL.
- **LAMP Stack:** Linux, Apache, MySQL and PHP.

HTML:

It is the language that browsers understand.

If we just write 'My cat is very grumpy' the browser would not understand if it's a head or paragraph. When we use , the browser understands that it is a paragraph. The paragraph will not take multiple lines, it will always render it in the same line.

INPUT:

** ** My cat is very grumpy.

My cat is very grumpy.

My cat is very grumpy.

My cat is very grumpy.

My cat is very grumpy.

** **

OUTPUT:

My cat is very grumpy. My cat is very grumpy. My cat is very grumpy. My cat is very grumpy. My cat is very grumpy.

Common HTML Tags

The following are some facts about HTML tags (plus a few facts about XHTML tags too):

- Web pages are just plain text. You can view or edit the source code using any text editor.
- "Tags" provide web browsers with instructions about the web page, such as where to display images, and how the document is structured.
- Tags are always enclosed in **angle brackets:** < >.
- Tags are comprised of **elements** and **attributes**. An element is an object on a page (such as a heading, paragraph, or image), and attributes are qualities that describe that element (such as width and height).
- Tags usually travel in pairs. An opening tag begins a section of page content, and a closing tag ends it. For example, to markup a section of text as a paragraph, you would open the paragraph with an opening paragraph tag ** ** and close it with a closing paragraph tag ** ** (closing tags always proceed the element with a /).
- A few tags are called *non-container tags*, because they don't contain any content - they stand alone. Examples are images and line breaks. XHTML is stricter than HTML, and requires that all open tags must be closed, even if they're not container tags. Therefore, non-container tags end in />. For example, the tag for

a line break is **
**. HTML does not have this same requirement, but it's a good habit to get into in case you ever need to code in XHTML.

- Tags in HTML are not case sensitive, but in XHTML all tags must be in lower case. Even when coding in HTML, you should get in the habit of writing tags in lower case.
- White space is ignored by web browsers. So, if you hit the space bar multiple times within a document, only one of those spaces will actually be displayed by the browser.
- Tags can be **nested**. For example:

  ```
  <div><p>This paragraph is nested inside a division</p></div>
  ```

  Note that the order of nested tags is important: The container tags surrounding any content should be **symmetrical**.

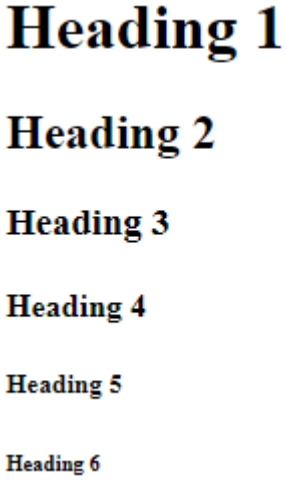
HTML Headings

HTML headings are defined with the <h1> to <h6> tags. <h1> defines the most important heading. <h6> defines the least important heading.

Input:

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
```

Output:



Headings Are Important

Search engines use the headings to index the structure and content of your web pages.

Users often skim a page by its headings. It is important to use headings to show the document structure.

<h1> headings should be used for main headings, followed by <h2> headings, then the less important <h3>, and so on.

Input:	Output:

`<h1>My Article Title</h1>`
 	
`<h2>Introduction</h2>`
`<p>paragraph</p>`
 	
`<h2>Some Content</h2>`
`<p>lorem ipsum dolor sit maet consect</p>`
	My Article Title
	Introduction
	paragraph
	Some Content
	lorem ipsum dolor sit maet consect

Bold Text:

Input:	Output:

`<p>This is my bold text</p>`
 	
`<p>This is my <strong>bold</strong> text</p>`
	This is my bold text
	This is my **bold** text

NOTE: Do not use <b> tag, as it outdated way.

Italic Text:

Input:	Output:

`<p>This is my text</p>`
 	
`<p>This is my <strong>bold <em>italic</em></strong> text</p>`
	This is my text
	This is my **bold *italic*** text

Attributes:

One important attribute is '*id*' attributes. It helps us in identify a particular element. It is used further in CSS even backend.

The id attribute specifies a unique id for an HTML element. The value of the id attribute must be unique within the HTML document.

The id attribute is used to point to a specific style declaration in a style sheet. It is also used by JavaScript to access and manipulate the element with the specific id.

Note: The id name is case sensitive!

Note: The id name must contain at least one character, and must not contain whitespaces (spaces, tabs, etc.)

```
<p id='myPara'>This is my <strong>bold <em>italic</em> </strong> text</p
>
```

If we use the attribute '**hidden="true"**' then it will not be visible on the webpage.

Nesting:

```
<p>My cat is <strong>very</strong> grumpy.</p>
```

```
<p>My cat is <strong>very grumpy.</p></strong>
```

Output:

My cat is **very** grumpy.

My cat is **very grumpy.**

IN HTML each element has an ending tag. But there are few exceptions.

Image Tag

<img src=‘’ alt=‘’>

The `src` Attribute

The <img> tag is used to embed an image in an HTML page. The src attribute specifies the path to the image to be displayed:

There are two ways to specify the URL in the src attribute:

1. Absolute URL - Links to an external image that is hosted on another website. Example: src="https://www.w3schools.com/images/img_girl.jpg".

Notes: External images might be under copyright. If you do not get permission to use it, you may be in violation of copyright laws. In addition, you cannot control external images; it can suddenly be removed or changed.

2. Relative URL - Links to an image that is hosted within the website. Here, the URL does not include the domain name. If the URL begins without a slash, it will be relative to the current page. Example: src="img_girl.jpg". If the URL begins with a slash, it will be relative to the domain. Example: src="/images/img_girl.jpg".

Tip: It is almost always best to use relative URLs. They will not break if you change domain.

The `alt` Attribute

The required alt attribute for the <img> tag specifies an alternate text for an image, if the image for some reason cannot be displayed. This can be due to slow connection, or an error in the src attribute, or if the user uses a screen reader.

The `width` and `height` Attributes

The <img> tag should also contain the width and height attributes, which specifies the width and height of the image

Links:

<a> - this is called a tag or anchor tag.

AttainU ➔ this takes us to the AttainU Website

Google

When we hover over the link, it shows what we have written inside the element and the link address.

Google

This will open the link in a new episode. We can even open other html files that are in the same folder.

Google

The HTML Document:

In vs code, if we type 'html:5' and it will write down the basic boiler plate completely.

All the content should be inside the <html> tag. Inside <html> tag, you have <head>

and <body> tags. In <head> everything related to document (meta-data) is found and in the <body> all the content is registered.

```
<!DOCTYPE html>
<htmml lang='en'>
    <head>
        <meta charset="UFT-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Document</title>
    </head>
    <body>
        <h1></h1>
    </body>
</html>
```

Metadata:

Metadata is data that describes other data. Meta is a prefix that -- in most information technology usages -- means "an underlying definition or description”. Metadata summarizes basic information about data, which can make finding and working with particular instances of data easier.

Metadata is data about data. A simple example of metadata for a document might include a collection of information like the author, file size, the date the document was created, and keywords to describe the document. Metadata for a music file might include the artist's name, the album, and the year it was released

<title> is not see on the web page but is visible on the tab.

** ** is used to have a logical separation or logical structure. It is used for division as well.

<article> The <article> element does not render as anything special in a browser. However, you can use CSS to style the <article> element. The <article> tag specifies independent, self-contained content. An article should make sense on its own and it should be possible to distribute it independently from the rest of the site.

ASCII :

UTF – 8 is another charset. ASCII has only 250 characters. Based on whatever characters we have in UTF-8, it would render accordingly.

Centre-Align: Center

NOTE: keep your HTML clean, only use CSS.