-----------------------------------HTML-------------------------------------

**Inline Element:** An inline element does not start on a new line and only takes up as much width as necessary. eg. <span>

**Block Level Elements:** A block-level element always starts on a new line and stretches out to the left and right as far as it can. eg. <div>

* HTML stands for Hyper Text Markup Language
* HTML describes the structure of Web pages using markup
* HTML elements are represented by tags
* Browsers do not display the HTML tags, but use them to render the content of the page
* HTML is the standard markup language for creating Web pages
* The <!DOCTYPE html> declaration defines this document to be
* The <!DOCTYPE> declaration represents the document type, and helps browsers to display web pages correctly.
* It must only appear once, at the top of the page (before any HTML tags).
* The <!DOCTYPE> declaration is not case sensitive.
* HTML5The <html> element is the root element of an HTML page
* The <head> element contains meta information about the document
* The <title> element specifies a title for the document
* The <body> element contains the visible page content

## **HTML Elements**

* The <h1> element defines a large heading
* The <h6>element defines a small heading
* <p> element defines a paragraph
* <a> element defines a link, link specified in attribute href
* <img> element defines image
* <button> defines button tag
* <br>element defines a **line break**.
* <hr>

The <hr> tag defines a thematic break in an HTML page, and is most often displayed as a horizontal rule.

The <hr> element is used to separate content (or define a change) in an HTML page:

* <pre>:

The HTML <pre> element defines preformatted text.

* **HTML Lists:**

HTML lists are defined with the <ul> (unordered/bullet list) or the <ol> (ordered/numbered list) tag, followed by <li> tags (list items)

* The <iframe> tag specifies an inline frame.

An inline frame is used to embed another document within the current HTML document.

# HTML Attributes

* Attributes provide additional information about HTML elements.
* All HTML elements can have **attributes**
* Attributes provide **additional information** about an element
* Attributes are always specified in **the start tag**

HTML links are defined with the <a> tag. The link address is specified in the href attribute.

HTML images are defined with the <img> tag. The filename of the image source is specified in the src attribute

alt attribute specifies an alternative text to be used, when an image cannot be displayed.

The style attribute is used to specify the styling of an element, like color, font, size etc.

The language is declared with the lang attribute.

## **Empty HTML Elements**

HTML elements with no content are called empty elements.

<br> is an empty element without a closing tag (the <br> tag defines a line break):

HTML Formatting Element

HTML defines special element for defining text with special meaning.

* <b> - Bold text
* <strong> - Important text
* <i> - Italic text
* <em> - Emphasized text
* <mark> - Marked text
* <small> - Small text
* <del> - Deleted text
* <ins> - Inserted text
* <sub> - Subscript text
* <sup> - Superscript text

# HTML Styles

* Use the style attribute for styling HTML elements
* Use background-color for background color
* Use color for text colors
* Use font-family for text fonts
* Use font-size for text sizes
* Use text-align for text alignment

# HTML Quotation and Citation Elements

* <q>
* The HTML <q> element defines a short quotation.
* Browsers usually insert quotation marks around the <q> element
* Abbriviation:
* The HTML <abbr> element defines an abbreviation or an acronym.
* Marking abbreviations can give useful information to browsers, translation systems and search-engines.
* <address>
* The HTML <address> element defines contact information (author/owner) of a document or an article.
* The <address> element is usually displayed in italic. Most browsers will add a line break before and after the element.

HTML Comment

* You can add comments to your HTML source by using the following syntax:
* <!-- Write your comments here -->
* Notice that there is an exclamation point (!) in the opening tag, but not in the closing tag.
* With comments you can place notifications and reminders in your HTML:

Java Script

* JavaScript is the programming language of HTML and the Web

-----------------------------------Angular JS-------------------------------------

# AngularJS Introduction

AngularJS is a **JavaScript framework**. It can be added to an HTML page with a <script> tag.

AngularJS extends HTML attributes with **Directives**, and binds data to HTML with **Expressions**.

AngularJS is a JavaScript framework written in JavaScript.

AngularJS is distributed as a JavaScript file, and can be added to a web page with a script tag:

AngularJS extends HTML with **ng-directives**.

The **ng-app** directive defines an AngularJS application.

The **ng-model** directive binds the value of HTML controls (input, select, textarea) to application data.

The **ng-bind** directive binds application data to the HTML view.

## **AngularJS Applications**

AngularJS **modules** define AngularJS applications.

AngularJS **controllers** control AngularJS applications.

The **ng-app** directive defines the application, the **ng-controller** directive defines the controller.

AngularJS binds data to HTML using **Expressions**.

## **AngularJS Expressions**

AngularJS expressions can be written inside double braces: {{ expression }}.

AngularJS expressions can also be written inside a directive: ng-bind="expression".

AngularJS will resolve the expression, and return the result exactly where the expression is written.

**AngularJS expressions** are much like **JavaScript expressions:** They can contain literals, operators, and variables.

Example {{ 5 + 5 }} or {{ firstName + " " + lastName }}

# AngularJS Modules

An AngularJS module defines an application.

The module is a container for the different parts of an application.

The module is a container for the application controllers.

Controllers always belong to a module.

**AngularJS Directives:**

AngularJS lets you extend HTML with new attributes called **Directives**.

AngularJS has a set of built-in directives which offers functionality to your applications.

AngularJS also lets you define your own directives.

The ng-app directive initializes an AngularJS application.

The ng-init directive initializes application data.

The ng-model directive binds the value of HTML controls (input, select, textarea) to application data.

The ng-repeat directive used on an array of objects:

# AngularJS ng-model Directive

The ng-model directive binds the value of HTML controls (input, select, textarea) to application data.

# AngularJS Controllers

AngularJS controllers **control the data** of AngularJS applications.

AngularJS applications are controlled by controllers.

The **ng-controller** directive defines the application controller.

# AngularJS Scope

The scope is the binding part between the HTML (view) and the JavaScript (controller).

The scope is an object with the available properties and methods.

The scope is available for both the view and the controller.

When adding properties to the $scope object in the controller, the view (HTML) gets access to these properties.

**Data Binding**

Data binding in AngularJS is the synchronization between the model and the view.

## **HTML View**

The HTML container where the AngularJS application is displayed, is called the view.

**Filters**

Filters can be added in AngularJS to format data for display to users.

AngularJS provides filters to transform data:

* currency Format a number to a currency format.
* date Format a date to a specified format.
* filter Select a subset of items from an array.
* json Format an object to a JSON string.
* limitTo Limits an array/string, into a specified number of elements/characters.
* lowercase Format a string to lower case.
* number Format a number to a string.
* orderBy Orders an array by an expression.
* uppercase Format a string to upper case.

Service

AngularJS services are JavaScript functions for specific tasks, which can be reused throughout the application.

In AngularJS you can make your own service, or use one of the many built-in services.

In AngularJS, a service is a function, or object, that is available for, and limited to, your AngularJS application.

AngularJS has about 30 built-in services. One of them is the $location service.

The $location service has methods which return information about the location of the current web page

Here are the things that you need to do to get a entry level job as a Front-End developer:

if you don't have a degree: \* Learn html, css (grid, flexbox), javascript, react \* Build real projects. Things that people want to use. There are hundreds of people seeking out the same jobs, you must stand out \* Network. Remember it's not what you know, it's who you know. But it's not who you know, it's how you know them. \* Have an online presence. You must write blog post OR make youtube videos OR speak at conferences. Any of the above to stand out. \* Have your resume and linkedin looked at by a professional. MANY people make huge mistakes that prevents them from getting a job