HTML

Hyper text markup language ⇒ Is not a programming language

If a language has design making capability, then it qualify as a programming language.

HTML cannot think, HTML is dumb🥺.

Hyper text ⇒ A special type of text document in which you can add hyperlink of another document.

Markup ⇒ Medium of writing and formatting documents.

- Why HTML was invented ?

  ⇒ HTML was invented to write research paper. For documentation purpose only HTML was made.

  ⇒ In morden website, now a days, water you see as a structure of the web app, everything is build by HTML.

  ⇒ HTML provides the skeleton to our web app.

The building blocks of a HTML is `tag.`

Tag is going to be a simple syntax, using which you can tell what is this part of document is referring to.

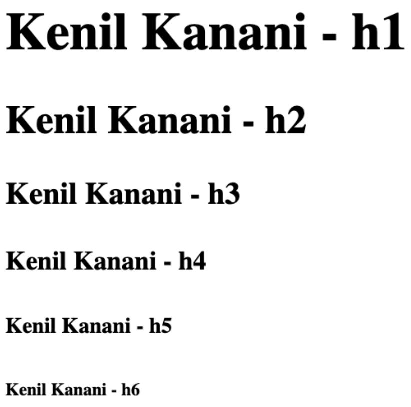
A simple mechanism, using which you can tell what is this part of document actually meaning.

```
<tag-name> content  </tag-name>   => Tag

<tag-name/>  => Self closing tag
```

h1-h6 tag

```
<h1>Kenil Kanani - h1</h1>
<h2>Kenil Kanani - h2</h2>
<h3>Kenil Kanani - h3</h3>
```

```
<h4>Kenil Kanani - h4</h4>
<h5>Kenil Kanani - h5</h5>
<h6>Kenil Kanani - h6</h6>

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



Kenil Kanani - h1

Kenil Kanani - h2

Kenil Kanani - h3

Kenil Kanani - h4

Kenil Kanani - h5

Kenil Kanani - h6

- When we write normal text in file without any tags, then it shows in Browser. Then why we do need a tag?

  ⇒ Even you write a plain text, the browser will try to wrap it in a HTML tag only, so it’s better to have things segregated in proper tags.

Lists in HTML

- Two types of List ⇒

  1. Unordered list

```
<ul>
    <li>List item 1</li>
    <li>List item 2</li>
    <li>List item 3</li>
    <li>List item 4</li>
    <ul>
        <li>Sublist item 1</li>
        <li>Sublist item 2</li>
        <li>Sublist item 3</li>
        <li>Sublist item 4</li>
    </ul>
</ul>
```

2. Ordered list

```
<ol>
    <li>List item 1</li>
    <li>List item 2</li>
    <li>List item 3</li>
    <li>List item 4</li>
    <ol>
        <li>Sublist item 1</li>
        <li>Sublist item 2</li>
        <li>Sublist item 3</li>
        <li>Sublist item 4</li>
    </ol>
</ol>
```

img tag in HTML

```
<img src="" alt="" />
```

- Here `src` and `alt` are attributes of `img` tag. Attributes are extra information about a tag.
- The `src` attribute specifies the source URL of the image, while the `alt` attribute provides alternative text that is displayed if the image cannot be loaded. It is important to provide descriptive and meaningful alternative text for accessibility purposes.
- Many people have disabilities, like not being able to see. When they use computer devices like phones or computers, there are special tools called screen readers in

web browsers and other software to help them.This screen readers are reading this `alt` attribute’s text.

- Additionally, we can add other attributes such as `width` and `height` to control the size of the image on the webpage.

Inline tag vs Block

```
<div>
  Any Content
 </div>
```

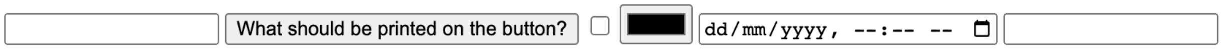
- Inline tags are used to wrap smaller portions of content, such as a `span` or a link, and they do `not create line breaks.`
- On the other hand, block tags, like `div` or `p`, are used to group larger sections of content and `create line breaks` before and after the content.
- Block tags are commonly used for structuring and organising the layout of a webpage.

marquee tag

```
<marquee behavior="scroll" direction="left">Hello, i am Kenil...</marquee>
```

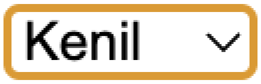
Input tag

```
<input type="text">
<input type="button" value="What should be printed on the button?">
<input type="checkbox">
<input type="color">
<input type="datetime-local">
<input type="number">
```



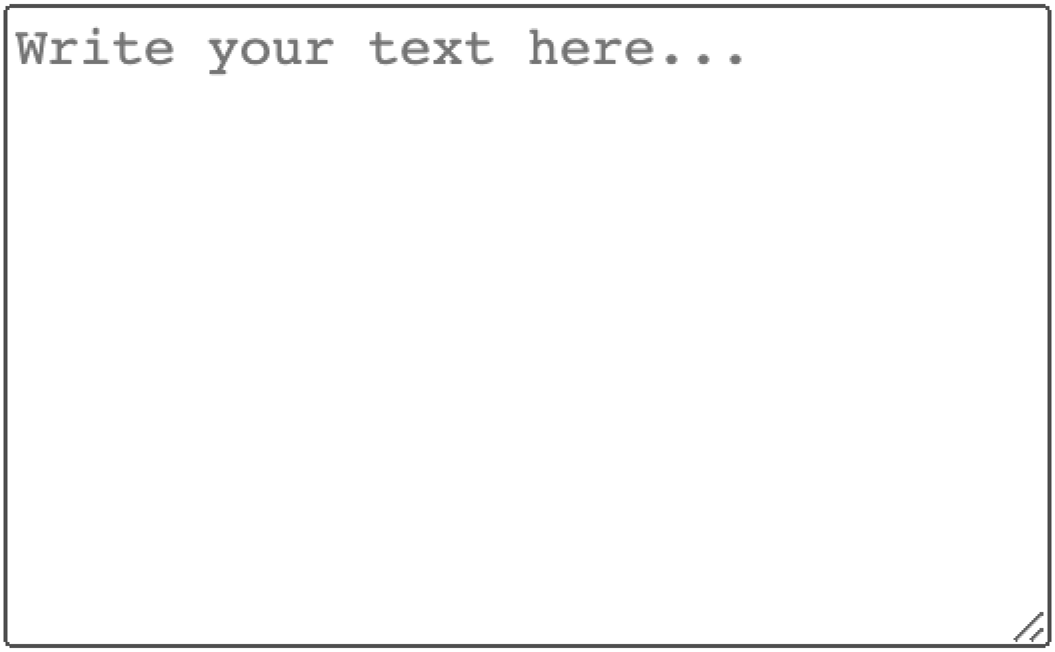
Select tag

```
<select>
  <option>Kenil</option>
  <option>Krish-1</option>
  <option>Krish-2</option>
</select>
```



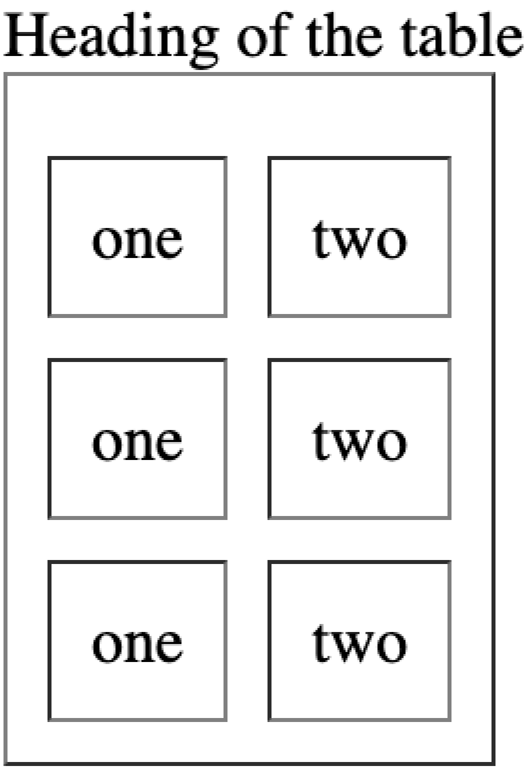
textarea

```
<textarea cols="30" rows="10" placeholder="Write your text here..."></textarea>
```



table

```
<table border="1" cellpadding="10px" cellspacing="10">
    <thead>
        <tr>Heading of the table</tr>
    </thead>
    <tbody>
        <tr>
            <td>one</td>
            <td>two</td>
        </tr>
        <tr>
            <td>one</td>
            <td>two</td>
        </tr>
        <tr>
            <td>one</td>
            <td>two</td>
        </tr>
    </tbody>
</table>
```



Heading of the table

one	two
one	two
one	two

Doctype

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>

</body>
</html>
```

The `<!DOCTYPE html>` declaration is used at the beginning of an HTML document to specify the version of HTML being used. In this case, `html` is the version and `lang="en"` specifies the language as English. This declaration helps the browser understand how to interpret and render the HTML content.

It is important to include the `<!DOCTYPE html>` declaration in every HTML document to ensure compatibility to web standards. Without this declaration, the browser may not render the document correctly.

The `<!DOCTYPE html>` declaration is followed by the `<html>` element, which serves as the root element of the HTML document. Inside the `<html>` element, we have the `<head>` and `<body>` elements. The `<head>` element contains meta-information about the document, such as the character encoding, viewport settings, and the document title. The `<body>` element contains the visible content of the webpage, including text, images, and other HTML elements.

Including the `<!DOCTYPE html>` declaration and structuring the HTML document correctly ensures proper rendering and compatibility across different browsers and devices.

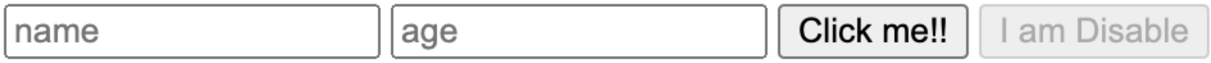
form

The `form` tag is used to create an interactive area on a webpage for user input. It allows users to enter data and submit it to a server for processing. The `form` tag contains various input elements such as text fields, checkboxes, radio buttons, and buttons that enable users to interact with the webpage.

By using the `form` tag, developers can collect user information, validate the data, and perform actions based on the user's input. It is an essential element in creating interactive and dynamic web applications.

input

```
<form>
        <input type="text" placeholder="name" name="firstname">
        <input type="number" placeholder="age" name="age">
        <input type="submit" value="Click me!!">
        <input type="submit" value="I am Disable" disabled>
</form>
```



- The present of `disabled` attribute is enough. we don't need to give it some value.
- Fun Fact : People can bypass the disabled property by simple JavaScript. So it's not enough to make check on UI part only. In Production, there are some checks on business logic.
- `name attribute` ⇒ In HTML, the `name` attribute is used to specify a name for an input element. This name is used to identify the input when the form is submitted to the server. It is important to provide a unique name for each input element within a form to ensure proper data handling on the server side.

HTML level Validation

- This validations are not useful in production, but it is good for starter.

```
<input type="text" placeholder="name" maxlength="10" autofocus>
```

- We can not write more then 10 characters in the input field. Even if we try to copy some sentence(with length more then 10), then also it paste upto 10 characters.
- `autofocus` also does not require any value, its presence is enough. When we refresh the page, then we automatically the input tag is on focus, which has `autofocus` attribute.
- When we have multiple `autofocus` in different-different tags, then it selects the first tag as autofocus.

Attributes Of Form

`action` ⇒ Write the url, wherever you want to submit the form.(i.e. ⇒ on your backend)

`method` ⇒

> The `method` attribute of the `<form>` tag specifies the HTTP method to be used when submitting the form data. The most common values for this attribute are `GET` and `POST`.

The `GET` method sends the form data as URL parameters, which can be seen in the browser's address bar. This method is typically used for forms with fewer inputs and for retrieving data from the server.

The `POST` method sends the form data in the body of the HTTP request, which is not visible in the browser's address bar. This method is used for forms with sensitive data, such as passwords, and for submitting larger amounts of data.

It is important to choose the appropriate method based on the requirements of your form and the sensitivity of the data being submitted.

Semantic HTML

- Generally, if you see the very old HTML, when not a lot of things introduced, most of the time you see a lots of `div` tags spreading around hear and their.
- And Technically, there is no problem with that, but it is not a very `clean` or `cool` way to implement a thing.
- So in order to write a clean `HTML` there is a concept called semantic HTML.
- Semantic HTML says that `you will write tags according to the context`.
- What you want to build, based on that you write the tags.
- It's not necessary to put `div` everywhere.
- It's give us the better search engine optimisation.
- Semantic HTML is not just recommendation, it's necessity.
    - `<header>`: Use this tag for the header section of our webpage. It typically contains the logo, navigation menu, and other introductory elements.
    - `<nav>`: This tag is used for the navigation menu of our webpage. It should contain links to different sections or pages of your website.
    - `<main>`: Use this tag to wrap the main content of our webpage. It represents the central content area and should not include headers, footers, or sidebars.
    - `<article>`: It can represent a blog post, news article, forum post, or any other independent piece of content.
    - `<section>`: It helps to organise the content and make it more structured.

- `<aside>`: This tag is used for content that is tangentially related to the main content but can be considered separate. It is often used for sidebars, pull quotes, or advertisements.
- `<footer>`: Use this tag for the footer section of our webpage. It typically contains copyright information, contact details, and other relevant information.
- `<figure>`: Use this tag to encapsulate media content, such as images or videos, along with their captions.
- `<figcaption>`: This tag is used as a caption for a `<figure>` element. It provides a description or explanation for the content within the `<figure>`.
- `<time>`: Use this tag to mark up a specific date, time, or duration. It is often used for publishing dates, event dates, or timestamps.

It's purely opinionated, that means if you want to use a particular type of an element, you can. If somebody doesn't want to use that, it's purely their choice. `HTML` is not going to restrict you for anything. Everything will be still shown on the browser. It's just that, how readability of the code is increasing.