What is HTML?

Think of a document that you would create in a word processor like Microsoft Word or Google Docs. They usually consist of more than one style.

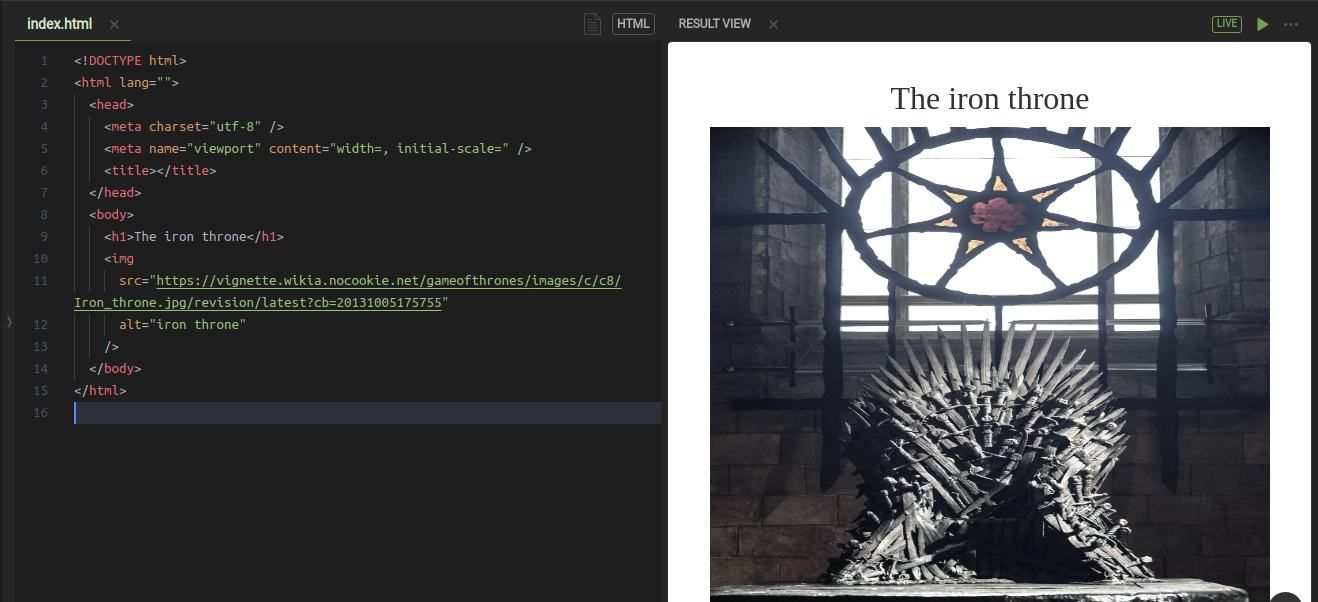
They use different font sizes to indicate different section of the text, like headers, body paragraphs, footers, table of contents, captions, and so on.

Unlike humans who can simply look at a document and understand the difference between a heading and a paragraph, computers don’t have that intuition. In order to correctly render a web page, it must be explicitly told what each piece of content is.

So how exactly do we instruct the browser to display web content? This is where Hypertext Markup Language(or HTML for short) can come in handy.

HTML is the language in which most websites are written with. It is used to create web pages and ensure their functionality.

* **Hypertext** defines the link between the web pages.
* **Markup language** is used to define the text document within a **tag** which determines the structure of web pages.



# What is HTML?

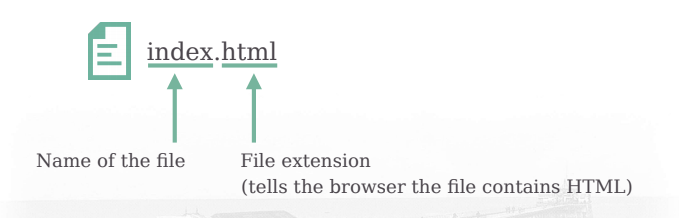
HTML allows you to take a plain text document created in any simple text editor and organize it into lists. It also creates links to other webpages, includes images, and much more.

So in other words, HTML outlines the structure or the skeleton of our webpage using **tags**.

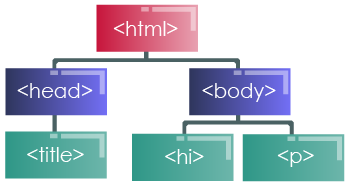
Just like most programming languages, we type a bunch of HTML into a file (aka. document) so we can send it around.

HTML documents are files that end with a **.html** or **.htm** extension. You can view them using any web browser (such as Google Chrome, Safari, or Mozilla Firefox). The browser reads the HTML file and renders its content so that internet users can view it.

The only thing that we have to do is to change the file extension into **.html**



# HTML tree

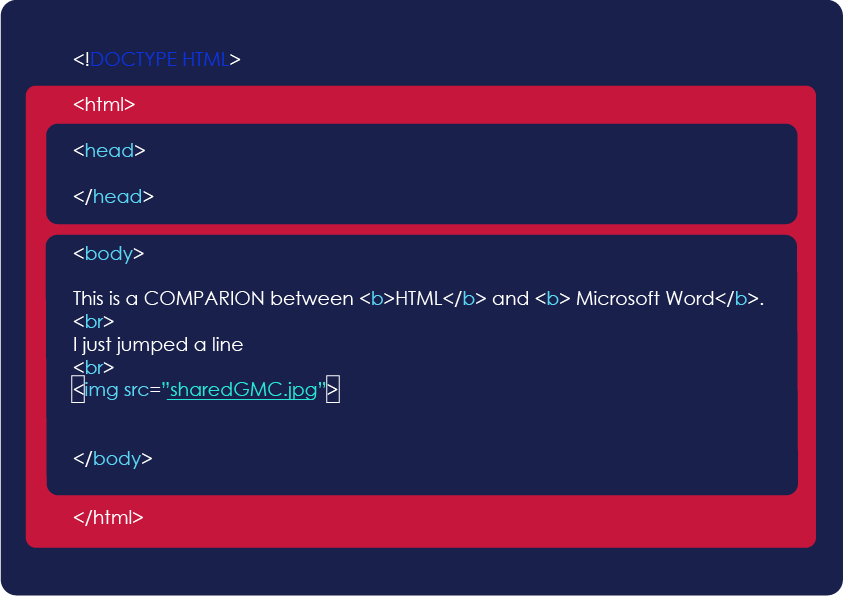
Since HTML is the skeleton of our web page, it should be very organized and well-structured to make it easy for the browser to display it.  
Here HTML is represented as a tree of tags. The root element of this tree is always the <html> tag.  


# HTML code structure:

As we have seen, HTML is based on tags. (<>)

We have to know that:

* All HTML documents must start with a document type declaration: <!DOCTYPE html>.
* The HTML document itself begins with < html > and ends with </ html>.
* The visible part of the HTML document is between < body > and </body >.
* A tag is closed this way: </ tagname>. For example, < head > is closed with < /head >.
* Changes are held between tags (For example, the code between < b > and < /b > becomes bold.



# Let's start Coding

Now, let’s make our hands a little bit dirty!

Remember the folder structure that we have talked about in the previous chapter? It’s time to use it.

Inside the GoMyCode folder, we are going to create a file named **index.html**

1. Open this file in VSCode and paste the following code.

<!DOCTYPE html>

<html>

<head>

<title> Hello world ! </title>

</head>

<body>

<h1> This is my first application ! </h1>

*<!-- this is a comment -->*

</body>

</html>

1. Save your changes.
2. Open **index.html** in the browser and see the result.

* No need to overthink it because we are going to understand things step by step.
* Everything in html is built using tags. There are two types of tags:
  + **Paired tag**: it has to be closed, it is composed of an opening tag and a closing tag with content inside of it.
  + **Unpaired tag**: this one does not have to be closed because it is a self closing tag.
* Every **element** in our html tree is composed of an opening tag <h1> a content and a closing tag </h1> .
* Comments are very important for the developers to understand their code and the others’ code.  
  The comment does not appear to the user. It exists only in the code file.  
  To create a comment in html, you only need to put it inside a comment tag <!-- whatever you want to say --> .
* One last thing, the HTML file structure is always the same

# The HTML tag:

### **As we have seen earlier, everything is contained inside the < html > tag.**

### **It is a crucial tag. It tells the browser that the content is located between < html > and < /html>.**

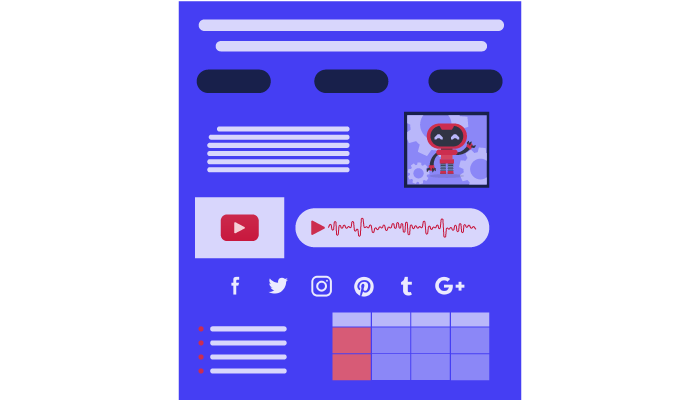
### **It is also called the “root” element.**

<html>

*<!--Everything-->*

</html>

## **The Body Tag:**



The body element appears after the head element in the page. It should contain all the content of your web page: text, images, and so on.

All web pages have one single body element, in order to understand the tags inside the body, we will divide them into:

* Headings
* Text Formatting and Line Breaking Tags
* Paragraphs
* Images
* Audio and Video
* Links
* Lists
* Tables
* Buttons
* Forms and Inputs

Let’s get to know them one by one.

# The head tag:

<head>

<title>My Beautiful Page</title>

<link rel="icon" href="favicon.ico" type="image/x-icon"/>

</head>

The head element contains information about the web page. You can put many elements inside of it such as:  
Title => Sets the title of the page  
Link => links elements to html, such as the page icon, CSS files...Etc.  
We will see the other tags later on in the course.  
Here’s an example down below,” rel” and “href” are called tag attributes, we will get back to them later in the course.  
For now, to set the icon, you need to set rel to “icon” and href to the link to your image.\*\*

# Heading tags:

The HTML standard has five additional text heading elements, appropriately defined from < h1 > to < h6 >. They are considered as Paired Tags (We have to close them )

<h1>This is a h1 Heading</h1>

# This is a h1 Heading

<h2>This is a h2 Heading</h2>

## **This is a h2 Heading**

<h3>This is a h3 Heading</h3>

### **This is a h3 Heading**

<h4>This is a h4 Heading</h4>

#### **This is a h4 Heading**

<h5>This is a h5 Heading</h5>

##### **This is a h5 Heading**

<h6>This is a h6 Heading</h6>

###### **This is a h6 Heading**

# Paired tag:

This is a <b>bold</b> tag test

This is a **bold** tag test

This is an <i>italic</i> tag test

This is an italic tag test

This is a <del>deleted</del> tag test

This is a ~~deleted~~ tag test

* Paired Tags: ( if the text is placed between a tag and its companion tag.) Example :  
  <b>: Bold Text  
  <i>: Italic Text  
  <small>: displays smaller text  
  <del>: defines deleted text
* Unpaired tag: (does not have a companion tag. In other words it doesn’t close)  
  <br>: breaks a line

Note: HTML doesn’t read spaces, you need to use <br> to jump a line

# Paragraph Tag:

The < p > Tag defines a paragraph in HTML.  
the browser breaks a line without the need of the < br> tag.  
Paragraphs are automatically separated by a line break.  
It's similar to writing a paragraph in a word document, paragraphs need to be separated from each other for them to be coherent and understandable.

Here’s an example :

<p>My first paragraph</p>

<p>My second paragraph</p>

#### **My first paragraph**

#### **My second paragraph**

# HTML attribute

HTML attributes provide additional information about a HTML element. Attributes can be considered as properties of the element. An element can have a single attribute, many attributes, or no attributes at all.  
Attributes are placed in the opening tag, with a space after the element declaration (or another attribute, if there are multiple).

Let’s take a look at an example heading with a title attribute:

<h2 title="This is a subheading">Hello, World!</h2>

Here's the output that we should have :

## **Hello, World!**

# The image tag:

The < img > tag is responsible for adding images to your HTML page.,Basically, the “src” attribute takes the URL of your image or its path on your computer. The “alt” attribute is your plan B if your image fails to load.  
Unlike most of the elements we have encountered thus far, the img element does not have a closing tag .

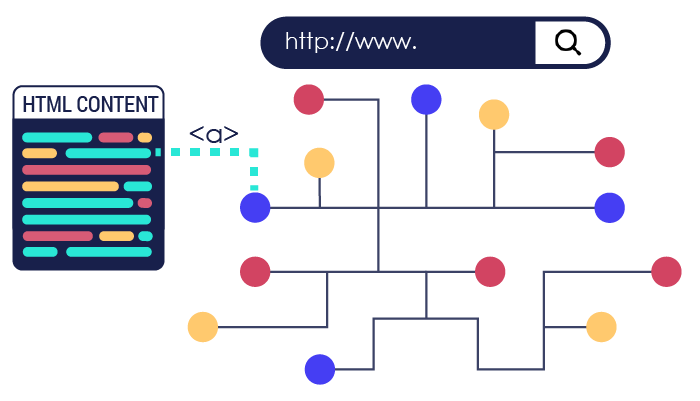
*<!-- Incorrect img declaration -->*

<img src="path/to/image/dog.jpg" alt="A dog" > </img>

*<!-- Correct img declaration -->*

<img src="path/to/image/dog.jpg" alt="A dog" >

# Linked page



One of the most important aspects of the World Wide Web is the ability to connect with other parts of the web.  
Without a method to redirect our HTML page to other web addresses, the web as we know it would cease to exist.  
We can connect a HTML page to other web pages by creating a hyperlink using the anchor tag.

# The a tag:

<a href="https://gomycode.tn/">Take me to GoMyCode's Website</a>

[Take me to GoMyCode's Website](https://gomycode.tn/)

Links allow users to navigate between different web pages. If you click on a link , you can jump to another document.  
Links are defined with the < a > tag also called anchor element.  
The “href” attribute defines the path once it’s clicked on.

# Internal link:

Links can be local links too. It will allow users to navigate on the same web page,  
In other words you can click on a link and jump to the end of the page, or to an element on your web page.

<body>

<a name="TopOfThePage"> </a>

<h1>Link Up Top</h1>

<p>

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Iure in ducimus,

tenetur facere repellendus ipsa beatae minus est laudantium nobis tempora

suscipit sunt tempore reprehenderit nulla necessitatibus! Eius sed

provident iusto dicta corrupti itaque, ducimus quo illo cupiditate,

quaerat blanditiis, quod accusamus doloremque. <br />

Neque unde nostrum dignissimos ad quam ratione.

</p>

<img src="Gomycode.png" alt="Gomycode" />

<p>

Lorem ipsum dolor sit amet consectetur adipisicing elit.

<br>

Quia, consequuntur!

</p>

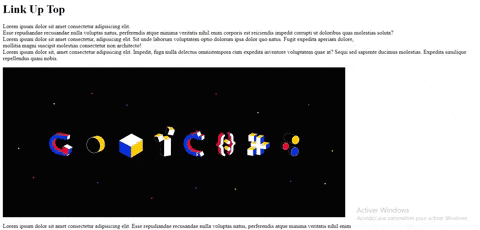
<img src="Track.jpg" alt="Gomycode">

<a href="#TopOfThePage"> Let's go up Top </a>

</body>

# Internal link:

This is what it should look like. As previously mentioned, local links makes it easier to navigate in your web pages.



# The audio tag:

To play an audio html file, use the < audio > element:  
The controls attribute is necessary. It adds audio controls like play, pause, and volume control.  
The < source > element allows you to specify alternative audio files from which the browser can choose from.  
The browser will use the first recognized format.  
The text between < audio >< /audio> will be displayed only if the audio file is not supported by the browser.

<audio controls>

<source src="test.mp3" type="audio/mpeg">

Please Check the audio Extension

</audio>

# The video tag:

We can also add videos to our web page by using the < video > element:  
It is preferable to include height and width attributes in your tag. In addition, to automatically start a video, we use the autoplay attribute:

<video width="320" height="240" autoplay>

<source src="test.mp4" type="video/mp4" />

Please Check the video extension.

</video>

# The iframe tag:

We can also add videos from youtube by simply using the iframe tag. Just go to Youtube, Pick a video , go to Share -> Embed -> Copy tag < iframe >

<iframe

width="420"

height="315"

src="https://www.youtube.com/embed/zcTFG\_F0FRs"

></iframe>

# HTML lists :

Lists are used to arrange related pieces of information so they are connected, understandable and easy to read. In modern web development, lists are workhorse elements, frequently used for navigation as well as general content.

Lists are advantageous from a structural point of view as they help create a well-structured, highly accessible, and easy-to-maintain document. They are also useful because they provide specialized elements to which you can attach CSS styles. Finally, semantically correct lists help visitors read your web site, and they simplify maintenance when your pages need to be updated.

There are many types of list. Let's have a look at the most used ones:

* Ordered list
* Unordered list
* Description list

# Unordered list:

To create an unordered list, we use the html tag <ul> typically rendered as a bulleted list but we can change the style of the list using the type attribute. Every element of the list is wrapped with tag <li>

<h1>Web Languages</h1>

<ul>

<li>HTML</li>

<li>CSS</li>

<li>JS</li>

</ul>

Our code output would be :

# Web Languages:

* HTML
* CSS
* JS

We can also change the styling type by using type attribute:

<h1>Tracks At GoMyCode</h1>

*<!-- Disc bullets -->*

<ul type="disc">

<li>AI Track</li>

<li>Web Track</li>

<li>Game Track</li>

<li>Data Science Track</li>

</ul>

*<!-- Square bullets -->*

<ul type="square">

<li>AI Track</li>

<li>Web Track</li>

<li>Game Track</li>

<li>Data Science Track</li>

</ul>

# Ordered List:

The HTML <ol> element represents an ordered list of item, typically rendered as a numbered list but like the unordered list it can be changed.

<h1>Web Ordered Languages:</h1>

<ol>

<li>HTML</li>

<li>CSS</li>

<li>JS</li>

</ol>

Our code output would be :

# Web Ordered Languages:

1. HTML
2. CSS
3. JS

Ordered list type values can be used to change the numbering scheme, and include the following:

* 1: Default numeric scheme
* I: Upper-case Roman numerals
* i: Lower-case Roman numerals
* A: Upper-case Alphabet
* a: Lower-case Alphabet

Here’s an example. You can try to do the rest by yourself.

*<!-- Upper-case Roman Numerals -->*

<ol type="I">

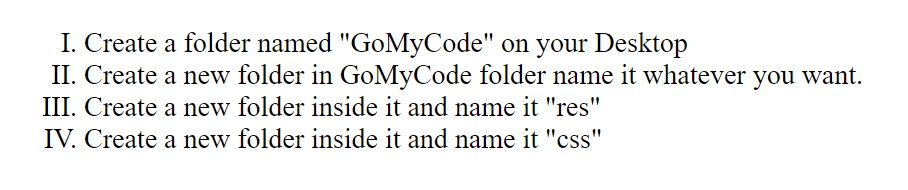
<li>Create a folder named "GoMyCode" on your Desktop</li>

<li>Create a new folder in GoMyCode folder name it whatever you want.</li>

<li>Create a new folder inside it and name it "res"</li>

<li>Create a new folder inside it and name it "css"</li>

</ol>



# Description list:

The HTML <dl> element represents a description list. The element encloses a list of groups that contain terms (specified using the <dt> element) and descriptions (provided by <dd> elements). The common uses for this element are: implementing a glossary or displaying metadata (a list of key-value pairs).

<p>The House Sigils In GOT:</p>

<dl>

<dt>House Stark</dt>

<dd>The Direwolf</dd>

<dt>House Lannister</dt>

<dd>The Lion</dd>

<dt>House Targaryen</dt>

<dd>The Three-Headed Dragon</dd>

<dt>House Baratheon</dt>

<dd>The Stag</dd>

</dl>



# Div tag:

<div> is certainly one of the most useful tags in HTML.  
The <div>, a block-level element,like a component, or a bag. It can contain many HTML elements in order to apply changes on them as a whole section.

<div> helps in organizing your code as well as in being creative using CSS. We will see how it does that later in the course.

<div>

<h1>This is the first division</h1>

</div>

<div>

<h1>This is the second division</h1>

</div>

Our code output would be :

## **This is the first division**

## **This is the second division**

Table

There are plenty of situations where you’ll want to present a table of data on your web page.  
If you haven’t worked with tabular data before, it will be useful to know that a table consists of rows and columns. Each row/column pair has a piece of data corresponding to it, referred to as a table cell.

Let’s dive straight in and convert the table below into HTML.

| **HTML** | **CSS** | **JS** |
| --- | --- | --- |
| Paragraphs | Selectors | DOM |
| Tables | Styles | Elements |

# The table tags:

HTML Tables are defined with the < table > tag.

* < th > tags are column names
* < tr > tags are rows
* < td > tags are column values of the row

<table border="1">

<tr>

<th>HTML</th>

<th>CSS</th>

<th>JS</th>

</tr>

<tr>

<td>Paragraphs</td>

<td>Selectors</td>

<td>DOM</td>

</tr>

<tr>

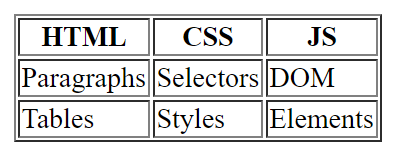
<td>Tables</td>

<td>Styles</td>

<td>Elements</td>

</tr>

</table>



# What is a semantic tag?

Up until now, we have focused on using HTML to structure our web pages and provide a clear presentation of the content. Following the HTML5 standard, you need to clearly indicate the meaning of each component of your web page’s content, that’s what we call that Semantic HTML.

Typically focusing on using HTML to semantically structure your web content gives you several advantages, including:

* Making your web content vastly more accessible to readers with disabilities.
* Applying styles with CSS will become more consistent and predictable.
* Search engines will use the semantic information to optimize and better understand your web pages.

# The semantic tags:

As we have already mentioned, Semantic tags are used to add meaning to our HTML document. A code should be comprehensible by whoever reads it.

Our document is usually divided into three different sections:

### **< header >**

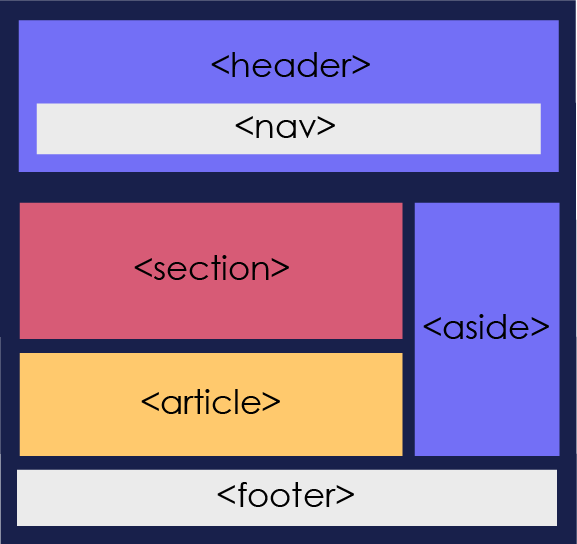
It's typically a group of introductory or navigational aids. It can contain some elements such as a logo, a search form, a slogan, etc.

### **< main >**

It can contain the main section of our website, we will explain this further in the next slides.

### **< footer >**

A footer typically contains information about the author, contact information, copyright data, etc.



# Nav tag:

Now that we have our page divided into header , main and footer, let’s move on to more semantic tags that can be useful to our code.

<nav> Element: it defines a set of navigation links.

<nav>

<a href="/html/> HTML </a>

<a href="/css/> CSS </a>

<a href="/js/> JavaScript </a>

<a href="/jquery/> jQuery </a>

</nav>

# Section tag:

The <section> element defines a section in a document.

For example, if we want to split our home page into several sections for: introduction, content, and contact information, etc...

<section>

<h1> Contact information <h1>

<p> Go My Code is an EdTech Startup.. </p>

</section>

# Article tag:

The <article> element defines an article in a document.

It can mean a blog entry, a news/scholarly article, or a forum post. These all remain good examples of content that would be semantically appropriate to store in an article element.

<article>

<h1> What is the perks of being a Web Developer? <h1>

*<!-- Article contents -->*

</article>

# Aside tag:

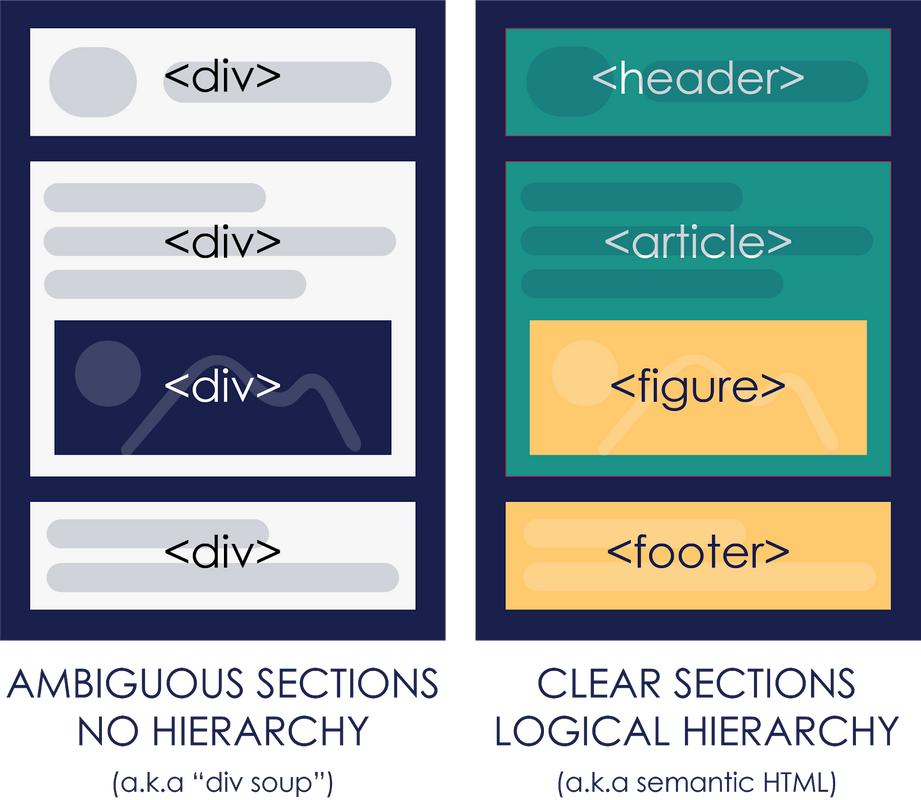
HTML offers many semantic elements to define distinct parts of a web page.

We’ll list the two most used semantic elements :

* The <aside> element : it defines whatever content that’s beside the content it is placed in (like a sidebar).
* <figure> and <figcaption> elements : its purpose is to regroup images and caption in one element.

# Advantages of semantic tag:

By adding semantic tags to your document, you provide extra information about that document, which helps in communication. Precisely, semantic tags produce the meaning of a page and its content, making it extremely clear to the browser.



# What is an html form?

In HTML, forms are a way of receiving input from users and they are very useful for collecting website data. Forms are highly relevant in today’s modern age as they are commonly used for sign-ups, authentication, and comments.

A form can be created with the <form> tag.

<html>

<head>

</head>

<body>

<form>

A form

</form>

</body>

</html>

The code output would be :  
A form

# Quick look

Here’s a quick look on how a form would look like using only HTML. Don’t mind the tags inside the <form> , we are going to have a closer look at them later.

<form>

<label>Name:</label>

<input type="text" name="first name" placeholder="Type your name here" />

<br />

<label>Number:</label>

<input type="number" name="number" value="23123456" />

<br />

<label>Birth Date:</label>

<input type="date" name="birthday" />

<br />

<label>Password:</label>

<input type="password" name="password" />

<br />

<input type="submit" value="Submit" />

</form>

Quick look

Here’s a quick look on how a form would look like using only HTML. Don’t mind the tags inside the <form> , we are going to have a closer look at them later.

<form>

<label>Name:</label>

<input type="text" name="first name" placeholder="Type your name here" />

<br />

<label>Number:</label>

<input type="number" name="number" value="23123456" />

<br />

<label>Birth Date:</label>

<input type="date" name="birthday" />

<br />

<label>Password:</label>

<input type="password" name="password" />

<br />

<input type="submit" value="Submit" />

</form>

# Form input:

The most useful component of a form is the input tag, which creates a text field where users can enter data. Here’s an example:

<form>

Search <input type = "text" name = "search" />

</form>

### **Output**



As seen in the code above, there is a type and name associated with inputs.  
The type defines the nature of the input (text, URL, email,...), while the name allows us to access the input data for future use. You can think of the name as a variable in which the input data is stored. Other properties of input include:

* size
* value
* maxlength
* readonly

# Selection inputs

You can use <select> (with nested <option>) elements to create a drop-down selection of items that a user can choose from.

<select>

<option value="WebTrack">Web Track</option>

<option value="AITrack">AI Track</option>

<option value="GameTrack">Game Track</option>

<option value="DataScienceTrack">Data Science Track</option>

</select>

# Text area:

If you want your user to be able to include new lines (by pressing return) in their text input, you can use a <textarea> element:

The rows attribute specifies the visible number of lines in a text area.

The columns (cols) attribute specifies the visible width of a text area.

<textarea name="message" cols="30" rows="5">

Welcome to GoMyCode

</textarea>

# Buttons:

A <button> element should be used whenever you want to create a clickable button to perform a certain action on the page.

<button> elements are simple to define, and have three different type values:

* submit: submits form data to a server
* reset: resets all the data in the current form
* button: no default behavior. This type of button will be more useful when we start dealing with Javascript.

<form>

<label>First Name:</label>

<input type="text" name="firstName" />

<br />

<label>Last Name:</label>

<input type="text" name="lastName" id="lastName" />

<br />

*<!-- This button will submit the form, causing the page to redirect -->*

<button type="submit">Submit Name</button>

</form>

# HTML RECAP

You’ve made quite the progress. Well Done!  
So up until now, we have been busy getting to know how important mastering HTML is.

What have we learned so far, you ask?  
Well, we have learned:

* How to build a webpage.
* How to design, plan and define the structure of the webpage.
* How to add content and use tags like a pro!

But, if you take a look at the end product, you will find that it’s still quite far from an actual web page. We have developed a static, unattractive web page with no aesthetic or forms output.

Right now, our number one priority is to solve that problem, and then we will start adding polish, style and elegance to our web page.