**Table of Contents**

[**TRACKER** 2](#_Toc169077442)

[**Day 1** – HTML Basics, CSS Basics, Hovers, Transitions, Shadows 2](#_Toc169077443)

[**Day 2** – Chrome DevTools & CSS Box Model, Text Styles, The HTML Structure 2](#_Toc169077444)

[**Day 3** – Images and Text Boxes, CSS Display Property, The div Element 2](#_Toc169077445)

[**Day 4** – Nested Layouts Techniques, CSS Grid, Flexbox 2](#_Toc169077446)

[**Day 5** – Nested Flexbox, CSS Position, Position Absolute and Relative 2](#_Toc169077447)

[**Day 6** – Finish the Project, More CSS Features 2](#_Toc169077448)

[**INTRODUCTION TO HTML** 3](#_Toc169077449)

[What is HTML? 3](#_Toc169077450)

[What is an HTML Element? 3](#_Toc169077451)

[HTML Page Structure 4](#_Toc169077452)

[**HTML & CSS Full Course - Beginner to Pro** 5](#_Toc169077453)

[**Day 1** 5](#_Toc169077454)

[HTML Basics 5](#_Toc169077455)

[CSS Basics 5](#_Toc169077456)

[**Day 2** 7](#_Toc169077457)

[Chrome DevTools & CSS Box Model 7](#_Toc169077458)

[Text Styles 7](#_Toc169077459)

[HTML Structure 8](#_Toc169077460)

[**Day 3** 8](#_Toc169077461)

[Images and Text Boxes 8](#_Toc169077462)

[CSS Display Property 9](#_Toc169077463)

[The div Element 9](#_Toc169077464)

[**Day 4** 10](#_Toc169077465)

[Nested Layouts Techniques 10](#_Toc169077466)

[CSS Grid 10](#_Toc169077467)

# **TRACKER**

## **Day 1** – HTML Basics, CSS Basics, Hovers, Transitions, Shadows

## **Day 2** – Chrome DevTools & CSS Box Model, Text Styles, The HTML Structure

## **Day 3** – Images and Text Boxes, CSS Display Property, The div Element

## **Day 4** – Nested Layouts Techniques, CSS Grid, Flexbox

## **Day 5** – Nested Flexbox, CSS Position, Position Absolute and Relative

## **Day 6** – Finish the Project, More CSS Features

# **INTRODUCTION TO HTML**

### What is HTML?

* HTML stands for ***Hyper Text Markup Language***
* HTML is the standard markup ***language for creating Web pages***
* HTML describes the ***structure of a Web page***
* HTML consists of ***a series of elements***
* HTML elements tell the browser ***how to display the content***
* HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

**EXAMPLE OF HTML DOCUMENT**

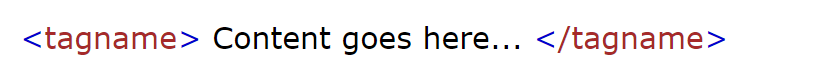
**A white background with black and white clouds

Description automatically generated**

* The <!DOCTYPE html> declaration defines that this document is an HTML5 document
* The <html> element is the root element of an HTML page
* The <head> element contains meta information about the HTML page
* The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
* The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
* The <h1> element defines a large heading
* The <p> element defines a paragraph

### What is an HTML Element?

An HTML element is defined by a start tag, some content, and an end tag:



The HTML **element** is everything from the start tag to the end tag:

A close-up of a text

Description automatically generated

### HTML Page Structure

A screenshot of a computer

Description automatically generated

# **HTML & CSS Full Course - Beginner to Pro**

## **Day 1**

### HTML Basics

* Instruction to create button - **<button> </button>**
* Instruction to create a paragraph of text - **<p> </p>**
* Buttons and paragraphs are types of elements
* **Syntax** means rules for writing code ( “<>” )
* **HTML Tag** tells the computer what to create (<button>, <p>)
* **HTML Tag** consists of opening and closing tag (< />)
* **Anchor element** means a link to another website (<a> </a>)
* **HTML Attribute** modifies how an elemen behaves (href=”” )
* The **<a>** tag defines a hyperlink. The **href** attribute specifies the URL of the page the link goes to (<a href="https://www.youtube.com"> Link to YouTube </a>)
* **Attribute syntax rule:** There has to be a space between the attribute and the tag name and there has to be an **=** and **“”**
* The **Target** attribute determines whether the link open at the current page or in a new tab. If you set the target attribute to **“\_blank”**, it will open in a new tab
* **Href and Target** attributes only work on the link element
* **Extra spaces** are ignored in HTML
* **Extra new lines** are also ignored

### CSS Basics

* **The style (<style> </style) element** allows authors to embed style information in their documents. The style element is one of several inputs to the styling processing model. The element does not represent content for the user.
* To change background colors - **background-color: color;**
* **CSS Syntax rules** are, it should have a CSS Selector that tells which elements we are targeting
* **“background-color:”** called the CSS Property. It tells the computer what we are changing while the “**color;”** called the CSS Value, it tells the computer what we’re changing the property to.
* To change text colors - **color: color;**
* To remove border property - **border: none;**
* To change height - **height: px;**
* To change width - **width: px;**
* To make rounded corners - **border-radius: px;** (The higher the px, the rounder the border)
* To make the cursor to pointer - **cursor: pointer;**
* To target styles to just one button, you go to opening tag and add the **“class”** attribute **(<button class="subscribe-button"> SUBSCRIBE </button>)** then rename the target class to the name you made **(subscribe-button)** then start with a dot **(.subscribe-button)**
* **GENERAL TECHNIQUE:** Create elements with HTML and Style with CSS one-by-one
* To change border color - **border-color: color;**
* To get solid color all around the border - **border-style: solid;**
* To change border width - **border-width: px;**
* To add space or margin - **margin-right: right;**
* To set the font into bold - **font-weight: bold;**
* To change the font size - **font-size: px;**
* To apply hover style **- .subscribe-button:hover {}**
* **:hover** is a pseudo-class style, it adds extra styles in a certain situation
* **:active** is a pseudo-class style**,** it activates when we click on the element
* Making the opacity 0 to 1 will be normal and It will start to fade in. Anything below 1 will have a fade-out effect when we hover.
* To have a fade-out effect when we click - **opacity: 0.5;**
* To transition smoothly when we hover - **transition: opacity 1s**
* To transition text color smoothly when we hover just put a comma and type what you want to transition and also put a duration - **transition: background-color 1s, color 1s;**
* To put a shadow **- box-shadow: 10px 10px 0 rgba(0,0,0,0);** ( horizontal, vertical, blur) and the “a” in the rgba value means the opacity of the color.
* Number from 0 to 1 of the opacity in rgba will make it just normal. Number from below 0 and 1, will make the opacity more transparent.

## **Day 2**

### Chrome DevTools & CSS Box Model

* Right click on an open space and click **inspect elements** to check and get the right styles.
* The first part of the box model that is facing of the outside element is called **Margin**
* Setting the height and the width is **not a good idea** when you have **a long text** because it will overflow
* The spaces inside of a button are called **Padding**
* **CSS Box Model** determines how much space an element takes up on the page and how far it is away from other elements
* With padding, even if you use more text, the element will resize with the text, but it will keep the spacing on the inside
* Normally the browser is text-based so it automatically aligns the text and not the elements. To align the elements - **vertical-align: top;**

### Text Styles

* To change the font style - **font-style: italic;**
* If we set a width, it’s going to force the text into second line
* To adjust the spacing between lines - **line-height: 20px;**
* HTML Entity lets us type special characters
* Paragraphs (<p>) by default, come with margin at the top and bottom
* We can avoid duplication by making new style - **p { font-family: Arial; }**
* CSS Specificity means if we have multiple line setting the same property, there’s a priority that the browser follows
* To know which has more priority, we have to look at the selector (p {} or .video-stats {}). The selector that’s more specific has higher priority.
* To have an underline effect when we hover on a text - **text-decoration: underline;**  but it will affect the entire paragraph. To target specific words, we need to learn about **text element** which means an element inside a line of text
* We uses a text element <span> </span> to modify a specific part of the text without affecting the whole thing

### HTML Structure

* Every html file are gonna start with **<!DOCTYPE html>**
* An element inside an element is called **Nesting**
* **<html>** represents an entire webpage
* **<body>** contains all elements that are visible
* **<title>** is inside the element **<header>** which will change the text shown in a tab
* To automatically save the code live, install an extension called **“Live Server”**
* **<style>** element should go inside the **<head>** element
* **<link>** element doesn’t require a closing tag and these types of elements are called **“Void Element”**.
* **<link> element** have two types of attributes: **rel** which means, attribute names a relationship of the linked document to the current document. The attribute must be a space-separated list of the link types values. The second one is **href**.
* To load the style elements from the css file - **<link rel="stylesheet" href="Buttons.css">**
* To link a file that’s inside a folder, give the name of the folder that’s beside the html file.

## **Day 3**

### Images and Text Boxes

* **<img>** element is another type of Void element and it takes one attribute which is **src =””** that tells the computer which image to display on a website
* A property that does makes sense for an image is **width** to change the size of it
* When we set the width of the image, the height also changes so it keeps the same dimensions or shape
* **Object-fit** Specifies how the contents of a replaced element should be scaled relative to the box established by its used height and width
* **Object-position** Determines the alignment of the replaced element inside its box.
* **object-fit: contain;** will shrink the image until it satisfies the width and height you set
* The **<input>** element represents a typed data field, usually with a form control to allow the user to edit the data
* **<input type="text">** is going to give us a text box
* The “Search” text inside the search bar will disappear once we type on it and it is called the label which tells the user what to put into the text box also known as **Placeholder**

### CSS Display Property

* The reason why the text box and the image are beside each other is because of the display property in css
* In HTML there are 3 types of elements: **Block element**, which means it takes up the entire line. Next is **the inline-block element, which** only takes up as much space as needed. Next is the **inline element,** they are text elements, that appear within a line of text
* Block to inline-block - **.video-author, .video-stats { display: inline-block;**
* Inline-block to block - **.thumbnail { width: 300px; display: block;**

### The div Element

* <div> means division or basically it’s just a box
* <div> is a block element
* <div> is useful because it can contain other elements
* <div> are meant to be containers
* Block elements take up the entire line in their container rather than on the page
* Inline-block elements only takes up as much space as needed
* Inline-block = the don’t take up the entire line and they appear beside each other

## **Day 4**

### Nested Layouts Techniques

* 2 types of nested layouts techniques are **vertical layout (Items are on top of each other)** and **horizontal layout (Items are beside each other)**

### CSS Grid

* It can create perfectly aligned horizontal layout
* Grid is a layout that has rows and columns
* To make a grid, we need to style the outer <div> or the container - **<div style=" display: grid; grid-template-columns: 100px 100px; ">**
* When an element is inside a grid, it’s actually place into the grid. It only takes up the entire column instead
* Grids maintain alignment much better
* If we set the grid-template-columns with only 3 columns, the 4th column will wrap around to the second row
* **Fr** means free space, it takes up the remaining amount of space on a grid
* **column-gap** means setting a space or a gap between a column

### Flexbox

* Flexbox is similar to CSS grid but it’s more flexible
* The layout is more rigid for grid
* if we change the order of the elements, the layout will change but it keeps its width even they move around
* justify-content property determines how elements are laid out horizontally
* align-items property determines how elements are aligned vertically

### Nested Flexbox

* Flexbox doesn’t go down two layers. It only affects elements on the first layer. That is why it needs to be turn into a flexbox in order to use the alignment-items: center
* .search-bar::placeholder to target an input element placeholder
* box-shadow: inset; to put the box shadow inside
* Since flexbox is flexible, it will start shrinking the items inside when there’s not enough space
* A flexbox special property that prevent certain items from shrinking is **flex-shrink: 0;** in contrast, **width: 0;** to shrink

## Day 5

### CSS Position

* It helps us do things like keep the header at top of the page while we scroll
* It also helps us keep the sidebar at the top of the page
* Whenever we see an element on top of another element, that’s using CSS position
* **Top** specifies how far away our element is from the top of the window
* We use top, left, bottom, right, to move an element around and resize it with **position fixed**.

## Position Absolute and Relative

* Position: fixed; are placed in the browser window
* Position: absolute; are placed on the page
* Z-index determines which elements appear in front and which elements appear behind