

# HTML

Codes	Functions
<!DOCTYPE html>	<ul style="list-style-type: none"> <li>- Tells the browser that this file is using <b>HTML5</b>.</li> <li>- Ensures the page displays correctly.</li> </ul>
<html lang="en">	<ul style="list-style-type: none"> <li>- Wraps the entire webpage.</li> <li>- <b>lang="en"</b> tells the browser the page is in English (good for accessibility + SEO).</li> </ul>
<head> section	
<title>...</title>	Sets the tab name of the webpage.
<meta charset="utf-8">	Makes sure the website supports international characters like emojis, symbols, languages.
<meta name="viewport" ...>	<ul style="list-style-type: none"> <li>- Ensures the website scales properly on phones, tablets, and desktops.</li> <li>- Makes the site responsive.</li> </ul>
<link rel="stylesheet" href="./style.css">	Connects to the <b>CSS</b> file
<script src="" defer></script>	<ul style="list-style-type: none"> <li>- Loads <b>JavaScript</b> without blocking the page.</li> <li>- defer improves page speed by running script after the <b>HTML</b> is loaded</li> </ul>
<body>	Holds all the visible content of the webpage.
<h1>My Terrarium</h1>	Main heading of the page. Good for structure and accessibility.
<div id="page">	<ul style="list-style-type: none"> <li>- A wrapper container.</li> <li>- Groups everything inside to control layout with <b>CSS</b></li> </ul>
<div id="left-container" class="container">  <div id="right-container" class="container">	<ul style="list-style-type: none"> <li>- Two containers that likely split the plants into left and right sides.</li> <li>- <b>class="container"</b> is used in <b>CSS</b> to style them consistently.</li> </ul>
<div class="plant-holder">	<ul style="list-style-type: none"> <li>- Wraps each plant image.</li> <li>- Lets you position plants neatly inside their individual boxes.</li> </ul>
<img ...>	
class="plant"	or <b>CSS</b> styling
id="plant1" ...	unique ID for <b>JavaScript</b> interactions
alt="..."	text description for accessibility or if the image fails to load
src="./images/...png"	file path of the image

# CSS

Codes	Functions
body {         font-family: helvetica, arial, sans-serif;     }	Sets the font for the whole webpage.
h1 {         color:brown;         text-align: center;     }	<ul style="list-style-type: none"> <li>- Makes the title brown.</li> <li>- Centers the heading.</li> </ul>
h1:hover {         color:aqua;     }	<ul style="list-style-type: none"> <li>- When mouse hovers over the title, it changes to aqua.</li> <li>- Adds interactivity.</li> </ul>
#left-container {         background-color: #9fd6d0;         width: 15%;         left: 0px;         top: 0px;         position: absolute;         height: 100%;         padding: 10px;     }	<ul style="list-style-type: none"> <li>- Creates the left panel of the terrarium.</li> <li>- </li> <li>- Fixed on the left side using position: absolute.</li> <li>- </li> <li>- Full height (100%).</li> <li>- </li> <li>- Light blue background.responsive.</li> </ul>
.container {         background-color:burlywood;         width: 15%;         top: 0px;         position: absolute;         height: 100%;         padding: 10px;     }	<ul style="list-style-type: none"> <li>- A general styling template for container sections.</li> <li>- Burlywood background.</li> <li>- Full-height panel.</li> </ul> <p>(Note: You override the background for left &amp; right via their IDs.)</p>
.plant-holder {         position: relative;         height: 13%;         left: -10px;     }	<ul style="list-style-type: none"> <li>- Holds each plant image.</li> <li>- position: relative allows plant images inside to be positioned accurately.</li> <li>- Adjusts position slightly to the left.</li> </ul>
.plant {         position: absolute;         max-width: 150%;         max-height: 150%;         z-index: 2;     }	<ul style="list-style-type: none"> <li>- Positions images inside the plant-holder.</li> <li>- Allows plants to be bigger than their container (up to 150%).</li> <li>- z-index: 2 ensures plants appear above the jar graphics.</li> </ul>

<pre>#left-container div img {     width: 50px; }</pre>	<ul style="list-style-type: none"> <li>- Makes plant images smaller on the left panel.</li> <li>- Only affects images inside left container.</li> </ul>
<pre>.jar-walls {     height: 80%;     width: 60%;     background: #2eafa0;     border-radius: 1rem;     position: absolute;     bottom: 0.5%;     left: 20%;     opacity: 0.5;     z-index: 1; }</pre>	<ul style="list-style-type: none"> <li>- Creates the main curved glass body of the jar.</li> <li>- Semi-transparent (opacity: 0.5).</li> <li>- Positioned in the center</li> </ul>
<pre>.jar-top {     width: 50%;     height: 5%;     background: #a8ffff;     position: absolute;     bottom: 80.5%;     left: 25%;     opacity: 0.7;     z-index: 1; }</pre>	<ul style="list-style-type: none"> <li>- Represents the top opening of the jar.</li> <li>- Positioned above the jar walls.</li> </ul>
<pre>.jar-bottom {     width: 50%;     height: 1%;     background: #3f615d;     position: absolute;     bottom: 0%;     left: 25%;     opacity: 0.7; }</pre>	<ul style="list-style-type: none"> <li>- Small strip representing the bottom edge of the jar.</li> </ul>
<pre>.dirt {     width: 60%;     height: 5%;     background: #3a241d;     position: absolute;     border-radius: 0 0 1rem 1rem;     bottom: 1%;     left: 20%;     opacity: 0.7;     z-index: -1; }</pre>	<ul style="list-style-type: none"> <li>- Brown soil at the bottom of the jar.</li> <li>- Behind the plants (z-index: -1).</li> </ul>

<pre>.jar-glossy-long {     width: 3%;     height: 20%;     border-radius: 2rem;     background: #ddfbff;     position: absolute;     bottom: 20%;     left: 5%; }</pre>	Long vertical glossy highlight on the jar wall to make it look shiny.
<pre>.jar-glossy-short {     width: 3%;     height: 5%;     border-radius: 2rem;     background: #ddfbff;     position: absolute;     bottom: 45%;     left: 5%; }</pre>	Smaller glossy highlight to add realism.

## **CSS (Part 2)**

<b>Codes</b>	<b>Functions</b>
<pre>@keyframes slide-in { /* animation */   0% {     transform: translateX(-100%);     opacity: 0;   }   100% {     transform: translateX(0);     opacity: 1;   } }  .plant {   animation: slide-in 1s ease-out; }</pre>	Animation of icons when loading in
<pre>.plant { /* transition */   transition: width 0.3s, height 1s, background-color 1s; }</pre>	Transition of the icons between the starting (animation) and the ending stage (where the icons are supposed to be at)
<pre>.plant:hover {   width: 200px;   height: 200px;   background-color: rgb(222, 240, 243); }</pre>	Similar to hover

# **JavaScript**

Document Object Model (DOM) – Represents the structure of a web page as a tree of objects. Javascript allows for interaction with the DOM to create interactivity.

Event Listeners – Functions that wait for specific events to occur and then execute code in response.

- Event Types: Clicks, key presses, and changes in the documents
- Targets: Every event is associated with a particular element in the DOM
- Event handler function: The code that should be executed when the event occurs

Closures – Allows an inner function to access variables from its outer function; useful for managing state and keeping track of data over time. Closures are also functions that remembers the variables from the place where it was created even when the function is executed.

<b>Codes</b>	<b>Functions</b>
<b>Commands</b>	
let	Used for most cases
const	Used if the value always stays the same
var	
<b>Function</b>	<b>Allows to group instructions into a single unit that can be executed when called</b>
	-
	-

# Resources

**LinkedIn Learning -** [https://www.linkedin.com/learning-login/share?account=76881922&forceAccount=false&redirect=https%3A%2F%2Fwww.linkedin.com%2Flearning%2Fhtml-css-and-javascript-building-the-web%3Ftrk%3Dshare\\_ent\\_url%26shareId%3DXWkUFj%252FOQwSNup6nIL4Lew%253D%253D](https://www.linkedin.com/learning-login/share?account=76881922&forceAccount=false&redirect=https%3A%2F%2Fwww.linkedin.com%2Flearning%2Fhtml-css-and-javascript-building-the-web%3Ftrk%3Dshare_ent_url%26shareId%3DXWkUFj%252FOQwSNup6nIL4Lew%253D%253D)

The screenshot shows the LinkedIn Learning platform interface. On the left, there's a sidebar with a navigation menu and a list of course modules. The main content area features a video player showing two women, Nitya Narasimhan and Jasmine Greenaway, smiling. A subtitle at the bottom of the video frame reads "Plus, you get to solve problems and think creatively." Below the video, there are sections for "Instructors", "Related to this course", and "Related courses". The "Related courses" section includes links to "GitHub Codespaces" and "JavaScript Essential Training". At the bottom of the page, there are buttons for "Ask a question" and a note about AI-powered learning.

**YouTube –** <https://www.youtube.com/watch?v=cHkN82X3KNU&t=14s>

The screenshot shows a YouTube video player. In the center, a digital calculator application is displayed, showing the calculation "100\*25". The calculator has a standard layout with a numeric keypad, arithmetic operators (+, -, \*, /), and function keys (AC, DEL, %). Below the calculator, a progress bar indicates the video is at 0:16 of 12:42. The video title at the bottom is "Create a Responsive Calculator in HTML CSS & JavaScript" and it's associated with the channel "CodingLab". The YouTube interface includes standard controls like play/pause, volume, and sharing options.

## W3Schools - [https://www.w3schools.com/html/html\\_youtube.asp](https://www.w3schools.com/html/html_youtube.asp)

The screenshot shows the W3Schools navigation bar at the top with links for Tutorials, References, Exercises, Certificates, and a search bar. The main content area is titled "HTML YouTube Videos". It features a sidebar with links to various HTML categories like Forms, Graphics, Media, APIs, Examples, and YouTube. The main content includes sections on playing YouTube videos in HTML, struggling with video formats, and YouTube video IDs. A sidebar on the right offers a "Get Certified in HTML" course for \$95.

## Hook Agency - <https://hookagency.com/blog/website-color-schemes/>

The screenshot shows a blog post by Hook Agency. The header includes the Hook Agency logo, navigation links for Pricing, Website Results, About, Services, Resources, Reviews, and Contact Us, and a phone number 612-324-7905 with a "Schedule Intro Call" button. The main content features a section titled "Curated colors in context." with a color palette tool and a "Happy Hues" color palette. A sidebar on the right is titled "Growth-mode contractors play the long game." with text about monthly updates, custom websites, and full access to accounts, along with a "Book an intro Call" button.

## StackOverflow - <https://stackoverflow.com/questions/73485179/is-there-a-way-to-add-my-own-html-and-css-files-into-github-pages-without-having>

stackoverflow About Products For Teams Search... Log in Sign up

Home Questions AI Assist Tags Challenges Chat Articles Users Companies COLLECTIVES + Communities for your favorite technologies [Explore all Collectives](#)

STACK INTERNAL Try for free Learn more

Is there a way to add my own HTML and CSS files into GitHub-Pages without having to use Jekyll layouts?

Asked 3 years, 5 months ago Modified 3 years, 5 months ago Viewed 637 times

I want to create my own website and have been working on the page using Bootstrap and my own CSS style sheet. Is there a way of just pushing my HTML and CSS files straight to GitHub-pages without using Jekyll?

1 Here is a simple project structure of my files:

```
graph TD; index[index.html] --> info[info.html]; index --> archives[archives.html]; index --> contact[contact.html]; reset[reset.css]; website[website.css]
```

The index.html is the home page and the other pages are accessed from the nav-bar. The CSS files are linked to each HTML file as well.

Any information or links to tutorials is appreciated.

github github-pages

Share Improve this question Follow asked Aug 25, 2022 at 9:32 mmiguiuell 107 1 12

Add a comment

1 Answer Sorted by: Highest score (default)

The Overflow Blog

- Don't let your backend write checks your frontend can't cache
- Is that allowed? Authentication and authorization in Model Context Protocol

Featured on Meta

- Community Asks Sprint Announcement – January 2026: Custom site-specific badges!
- AI Assist: recent updates (January 2026)
- All users on Stack Overflow can now participate in chat
- Policy: Generative AI (e.g., ChatGPT) is banned
- Stack Overflow now uses machine learning to flag spam automatically
- No, I do not believe this is the end

AD Community activity Last 1 hr

- 4940 users online
- 7 questions
- 10 answers
- 13 comments
- 122 upvotes

Popular tags