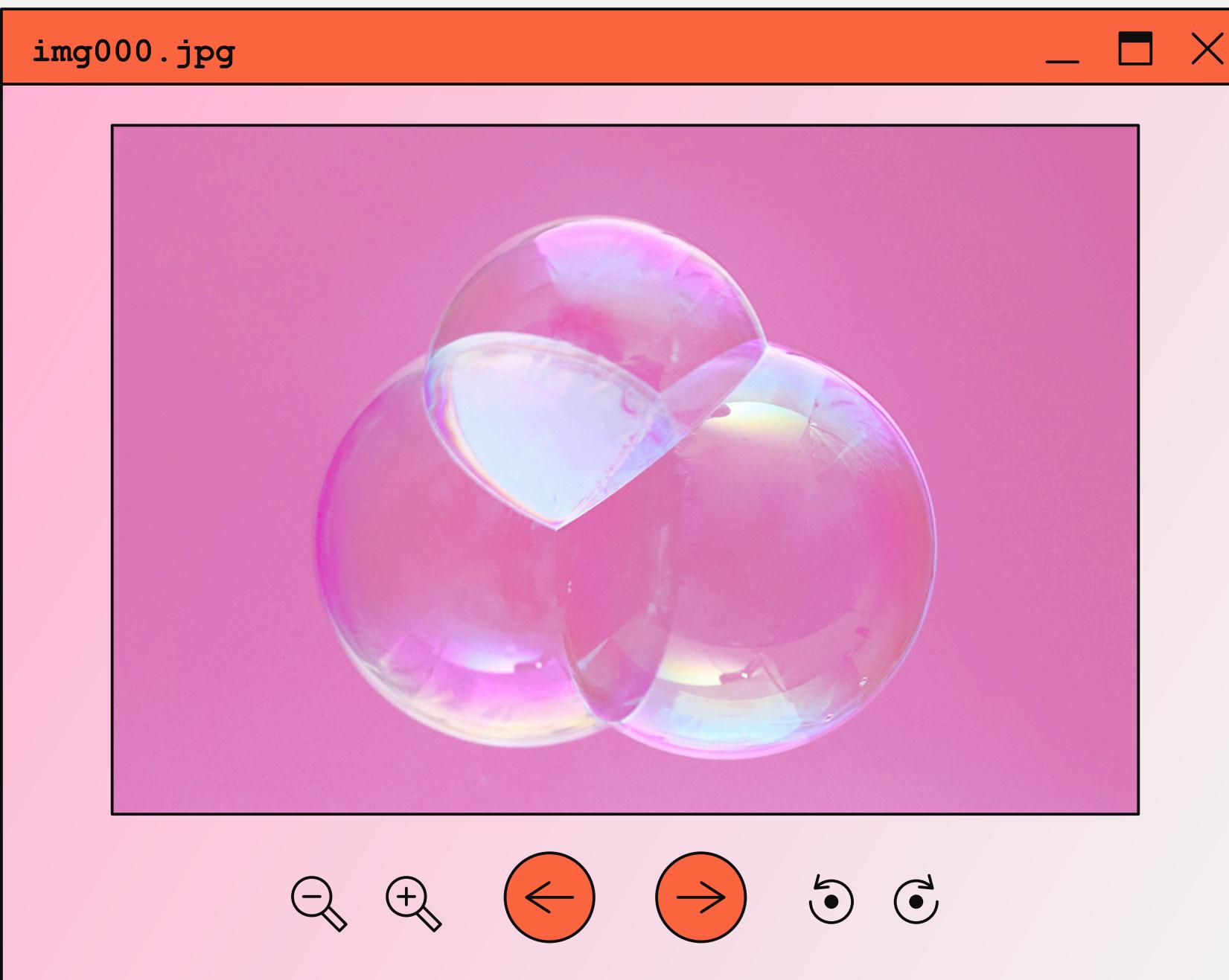


• • •

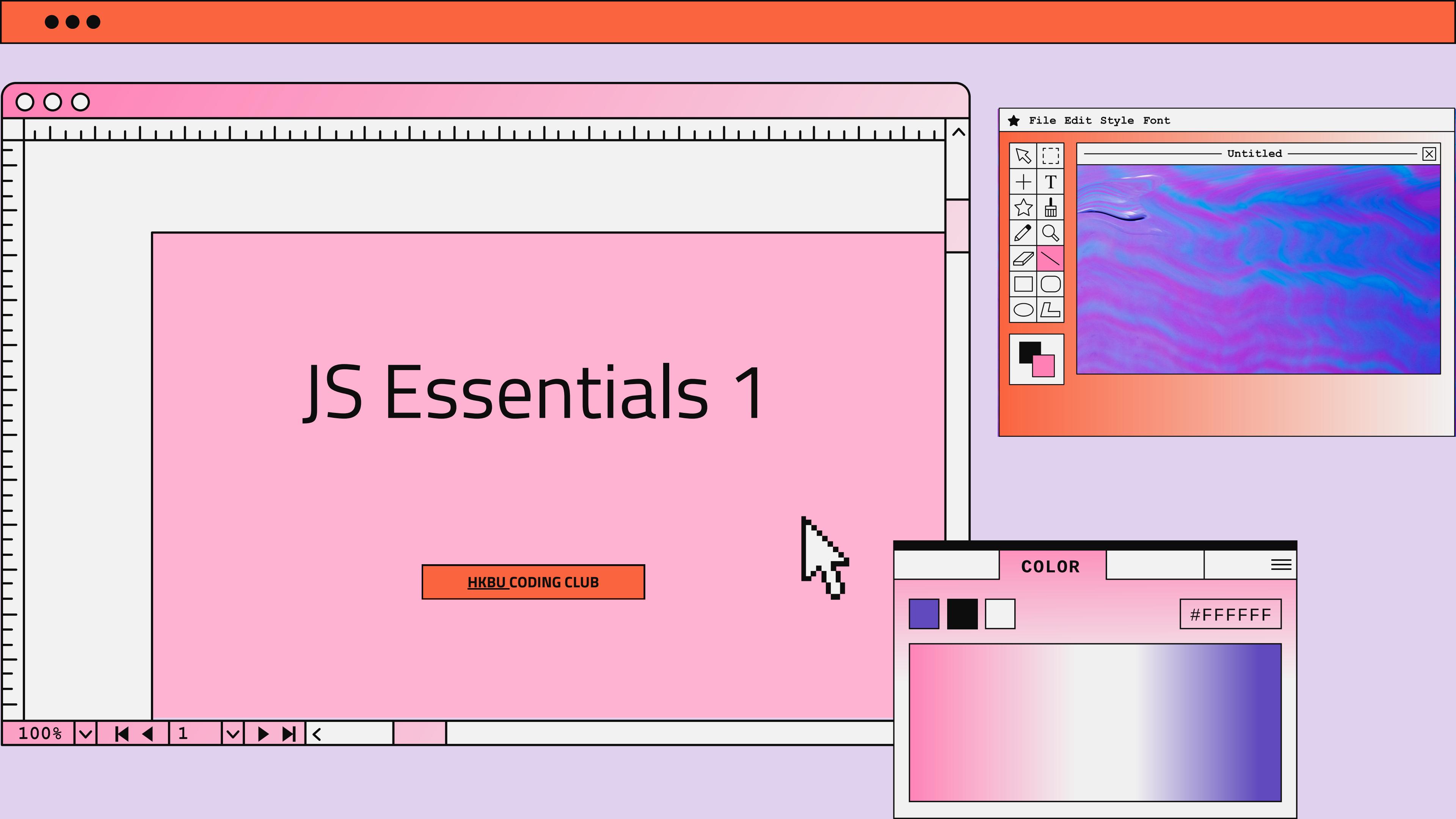
WORKSHOP PLAN



Untitled - TextEdit

File Edit View Help

01	INTRODUCTION TO HTML & CSS
02	Creating Modern Websites using Bootstrap and Gen-AI
03	JS Essentials 1
04	JS Essentials 2
05	Hike
06	Introduction to Git and Deployment



PLAN

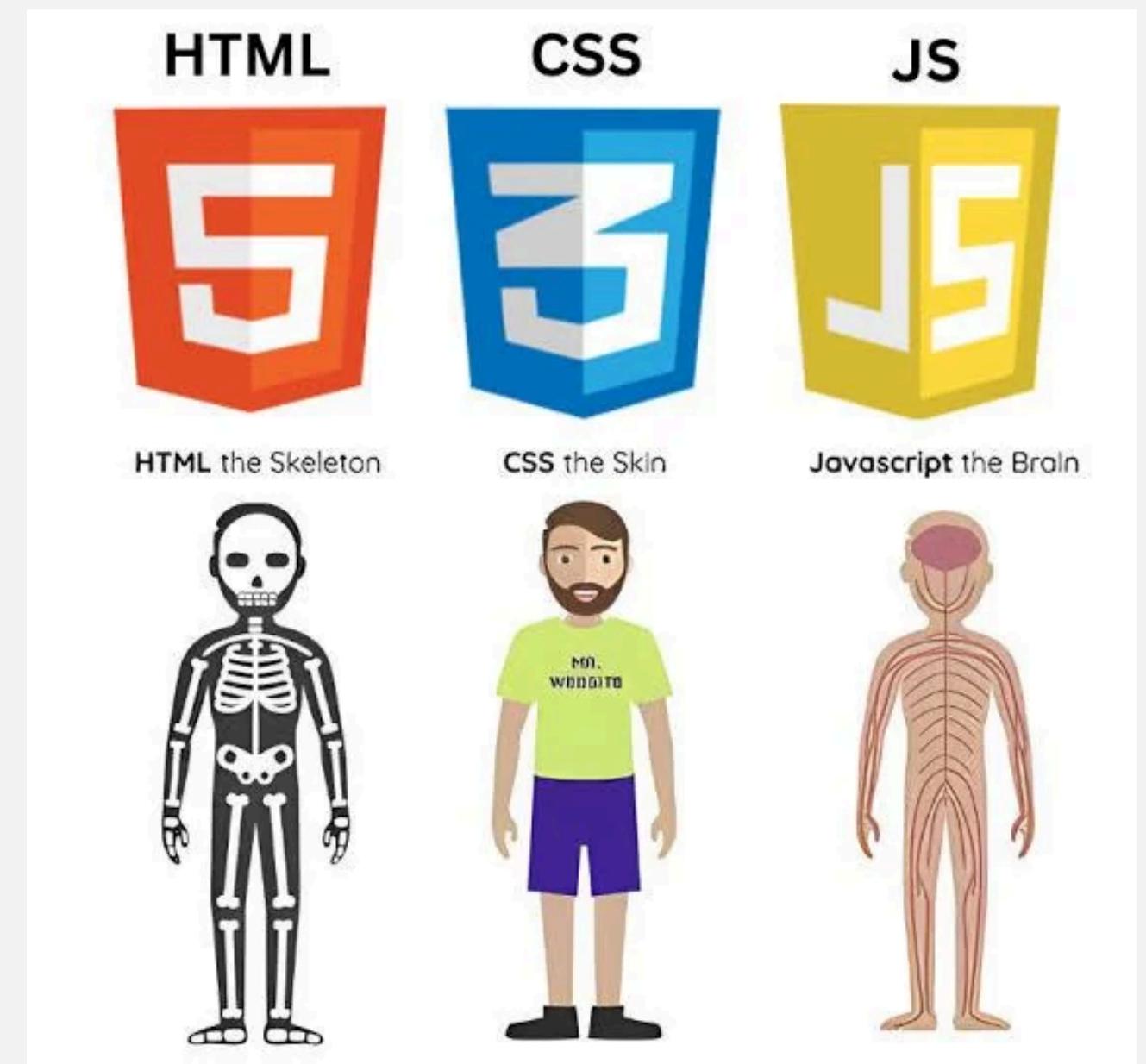
7:00 - 7:50 → Building Portfolio Site + Recap

8:00 - 8:50 → JS Basics

SOME BASICS

HOW TO MAKE WEBSITES?

- Use HTML to define elements
- Use CSS/ BOOTSTRAP to add style to the elements
- Use JavaScript to add functionality to the elements



SOME BASICS

HTML

HYPER TEXT MARKUP LANGUAGE

HTML is not a programming language; instead, it is a markup language that uses tags to define elements within a document.

Every piece of content on a web page - be it text, images, videos, or forms - is placed using HTML.

```
1 <html>
2   <head>
3     <title>Portfolio Website</title>
4   </head>
5
6   <body>
7
8     <h1>Portfolio Website</h1>
9
10   </body>
11
12 </html>
```

Portfolio Website

SOME BASICS

How to change
text color here?

Portfolio Website



Portfolio Website

USE WHAT?

SOME BASICS

```
<body>  
  <h1 style="color: #aqua;">Portfolio Website</h1>  
</body>
```

SOME BASICS

CSS

CASCADING STYLE SHEETS

CLASS

A CSS class is a reusable attribute used to apply the same styling rules to multiple HTML elements simultaneously

What if I have many headings?

Portfolio

Portfolio

Portfolio

Give them same class

```
<h1 class="headings">Portfolio</h1>
<h2 class="headings">Portfolio</h1>
<h3 class="headings">Portfolio</h3>
```

What if I have many headings?

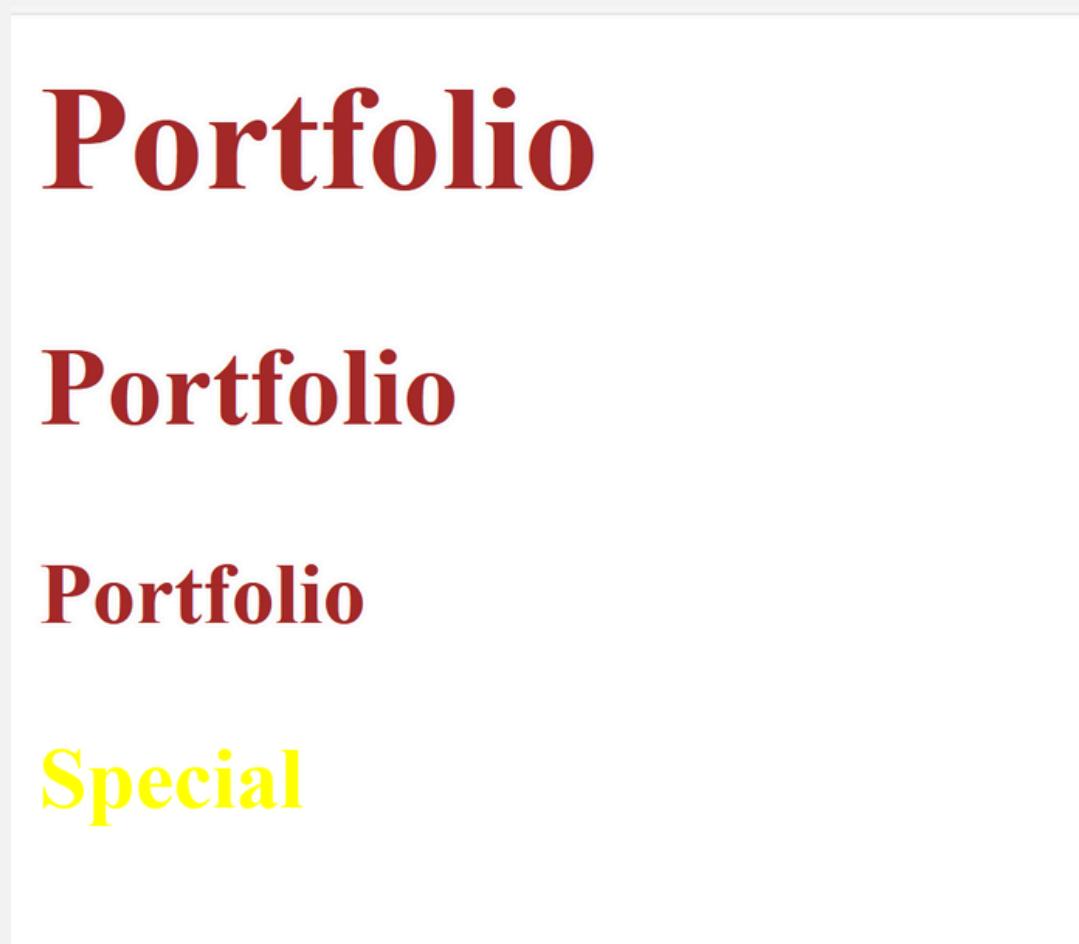
```
1 <html>
2   <head>
3     <title>Portfolio</title>
4     <style>
5       .headings{
6         color: brown;
7       }
8     </style>
9   </head>
10  <body>
11    <h1 class="headings">Portfolio</h1>
12    <h2 class="headings">Portfolio</h2>
13    <h3 class="headings">Portfolio</h3>
14  </body>
15 </html>
```

use **<style>** tag to write inline CSS

**.className {
 property: value
}**

ID

The CSS ID selector is used to target a single, unique HTML element on a webpage for styling. Use # to access.



```
1 <html>
2   <head>
3     <title>Portfolio</title>
4     <style>
5       .headings{
6         color: brown;
7       }
8       #special{
9         color: yellow;
10      }
11
12     </style>
13   </head>
14   <body>
15     <h1 class="headings">Portfolio</h1>
16     <h2 class="headings">Portfolio</h2>
17     <h3 class="headings">Portfolio</h3>
18     <h3 id="special">Special</h3>
19   </body>
```

• • •

STEP BY STEP MAKING PORTFOLIO SITE

1) Open folder On VS CODE

**2) Create main.html, Add Head,
Body, Title**

STYLE

3) Navigation

BODY

```
<header>
<nav>
<div class="logo">Bhawish Kumar</div>
<ul>
<li><a href="#about">About</a></li>
<li><a href="#projects">Projects</a></li>
<li><a href="#skills">Skills</a></li>
<li><a href="#contact">Contact</a></li>
</ul>
</nav>
</header>
```

```
<style>
/* Basic page styles */
body {
margin: 0;
font-family: Arial, sans-serif;
background: #0f172a;
color: #e2e8f0;
}

/* Navigation bar */
header { border-bottom: 1px solid #1f2937; }

nav {
max-width: 920px;
margin: 0 auto;
padding: 16px;
display: flex;
justify-content: space-between;
align-items: center;
}

/* Logo */
.logo { font-weight: bold; }

/* Links */
ul {
list-style: none;
margin: 0;
padding: 0;
display: flex;
gap: 16px;
}
a { color: #60a5fa; text-decoration: none; }
a:hover { text-decoration: underline; }

</style>
```

3) Heading + Button

BODY

```
<!-- Hero -->
<section class="hero">
  <h1>Bhawish Kumar</h1>
  <p>Full Stack Developer</p>

  <div class="buttons">
    <!-- Contact button jumps to the contact section -->
    <a class="btn primary" href="#contact">Contact Me</a>

    <!-- Download resume -->
    <a class="btn ghost" download>Download Resume</a>
  </div>
</section>
```

STYLE

```
/* Hero and Buttons */
```

```
.hero {
  max-width: 920px; margin: 32px auto; padding: 0 16px;
}
.hero h1 { margin: 0 0 8px; font-size: 32px; }
.hero p { margin: 0 0 16px; color: #94a3b8; }
```

```
/* Buttons are just styled links */
```

```
.buttons { display: flex; gap: 12px; }
.btn {
  display: inline-block;
  padding: 10px 14px;
  border-radius: 6px;
  text-decoration: none;
  font-weight: bold;
  border: 1px solid #1f2937;
}
.btn.primary { background: #60a5fa; color: #0b1220; border-color: transparent; }
.btn.ghost { background: transparent; color: #e2e8f0; }
```

4) About Me

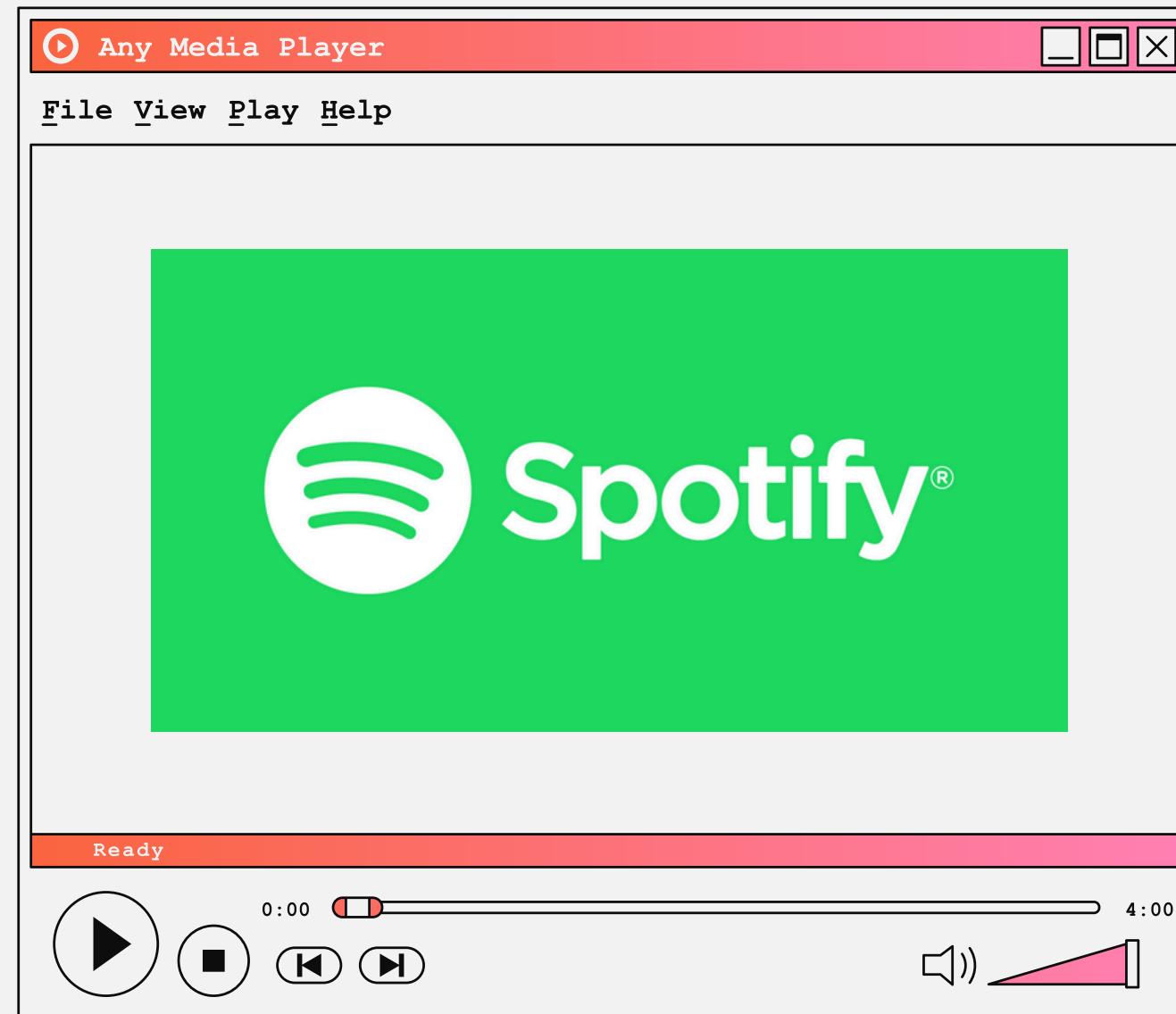
BODY

```
<section id="about" class="about">
  <h2>About Me</h2>
  <p>
    I'm Bhawish Kumar, a third-year Computer
    Science student in Hong Kong.
  </p>
  <ul>
    <li>Interests: Web dev, clean UI, tooling</li>
    <li>Stack: HTML, CSS, JavaScript, Python</li>
    <li>From Pakistan · Based in Hong Kong</li>
  </ul>
</section>
```

STYLE

```
/* --- About section --- */
.about {
  max-width: 920px;
  margin: 32px auto;
  padding: 0 16px;
}
```

ANALOGY

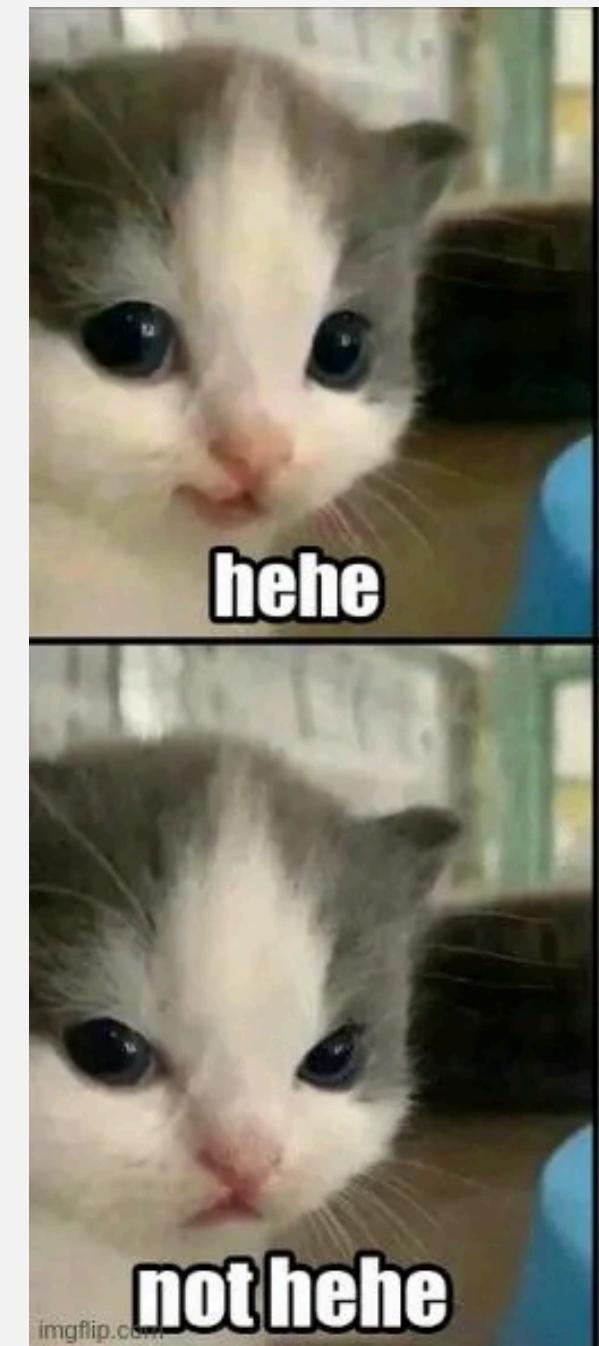


Play Button
Stop Button
Prev Button
Next Button

How it works?

What is JavaScript?

- JavaScript is a programming language that is used to make websites interactive and dynamic. Javascript can be used both in Frontend development and in Backend development.
- Gives functionality to UI
- Nothing like Java. Please don't compare!



What is JavaScript?

- In Frontend development, JavaScript works with HTML to make webpages dynamic, by getting data from APIs, managing user interaction like a mouse click, inputting data in a form, some button clicks, etc. events, and a lot more.
- Whereas in Backend development, JavaScript behaves as a full-time programming language. All it needs is a NodeJS setup.
- Javascript is slightly hard for beginner to learn that's why we teach Python first so you can get some idea. But JS is much cooler than Python 😊 and more applications overall!

A Little Bit Of JS Theory



JavaScript Features

Because I like theory!

Following are some of the most useful features of JavaScript:

- Light Weight
- Dynamically Typed
- Object-Based
- Functional
- Platform Independent
- Prototype-based
- Interpreted
- Asynchronous

JS in HTML

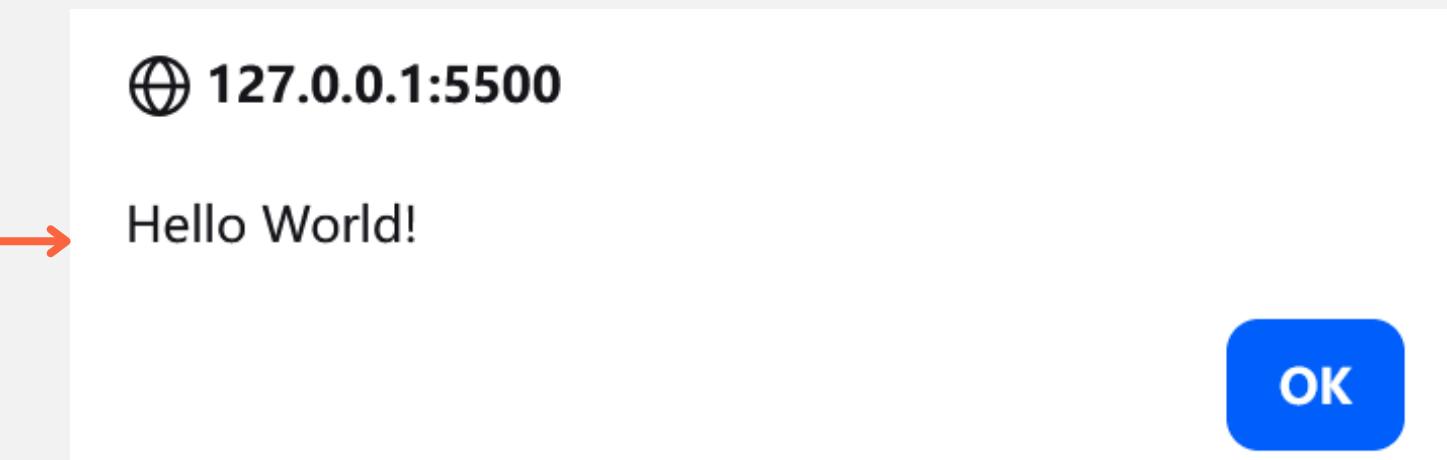
- You can write JavaScript code within your HTML file. You can create an HTML file with an extension .html and inside it, you can use the **<script> tag** to write JavaScript code.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>My HTML Webpage</title>
5      <script>
6          // JavaScript code
7          alert("Hello World!");
8      </script>
9  </head>
10 <body>
11     <!-- HTML code for webpage -->
12 </body>
13 </html>
```

JS in HTML

- You can write JavaScript code within your HTML file. You can create an HTML file with an extension .html and inside it, you can use the **<script> tag** to write JavaScript code.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>My HTML Webpage</title>
5   <script>
6     // JavaScript code
7     alert("Hello World!");
8   </script>
9 </head>
10 <body>
11   <!-- HTML code for webpage -->
12 </body>
13 </html>
```



- **don't worry about syntax, just try this!**

.js File

- You can also create a file with the extension .js to write JavaScript code. In a JavaScript file, you can directly write the JavaScript code, for example: I create 1.js and and then put it in html using <script> tag with src to the 1.js

```
3 <head>
4     <title>My HTML Webpage</title>
5     <script src="1.js"></script>
6
7 </head>
```

.js File

- You can also create a file with the extension .js to write JavaScript code. In a JavaScript file, you can directly write the JavaScript code, for example: I create 1.js and and then put it in html using <script> tag with src to the 1.js

```
3 <head>
4     <title>My HTML Webpage</title>
5     <script src="1.js"></script>
6
7 </head>
```

Basics of JS

1. Hello World

```
alert("Hello World");
```

2. Creating Variable

```
let bestTeacher= "Bhawish";
console.log(bestTeacher);
```

Basics of JS

1. Hello World

```
alert("Hello World");
```

2. Creating Variable

```
let bestTeacher= "Bhawish";  
console.log(bestTeacher);
```

In JavaScript, let is a keyword that is used to declare a local variable with block scope. In ECMAScript2016(ES6) the let keyword was introduced to define local scope variables.



Don't worry
about scope
right now

Basics of JS

No need to tell what type of a variable



Basics of JS



PRACTICE

3. If-else condition

```
4 let num1 = 10;  
5  
6 if(num1 > 0) {  
7   console.log("It's a +ve number");  
8 }  
9 else {  
10  console.log("It's a -ve number");  
11 }  
12
```

Basics of JS

4. Data Types

- When you create a variable in JavaScript, you do not have to worry about what type of data you will store in it because JavaScript is a Dynamically Typed language.
- A dynamically typed language is a language that infers data types at runtime. It allows storing different types of values in a variable.

Basics of JS

4. Data Types

```
let x = 5;      // x is a number  
x = "studytonight";    // x is string here.
```

- You could change the type of x anytime you want!

Basics of JS

4. Data Types

1. String
2. Boolean
3. Number
4. Undefined
5. Null

5. Composite Data Types

1. Object
2. Array

Check File:2.js

Basics of JS

String

- Basically text
- For example "HKBUCodingClub", "1234", "STU\$%12", etc.
- We can use both single or double quotes to create string values.

Basics of JS

Boolean

- The boolean type can have either true or false as a value. Boolean values are used when you need some value as a flag in your code, or in the case of conditional statements, etc.
- `let isOn = true; // bulb is on`

Basics of JS

Number

- Number data type means all numeric values, be it integer values, or fractional values. Integer values are whole numbers, and fractional values are valued with decimal points.
- `let x = 45; // Number without decimal point`

Basics of JS

Undefined

- When a variable is declared but not assigned any value, its value is undefined. Yes, undefined is a valid data type in JavaScript and it can have only one undefined value.
- `let a;`
- `console.log(a); // undefined`

Basics of JS

Null

- The null value is used to represent no value. It is neither similar to undefined, nor equal to empty value or zero value. The null datatype means, no value. You can assign a null value to a variable.
- The Null data type can have only one value, which is null. Let's take an example of this:
- `let a = null;`
- `console.log(a); // Output will be null`

Basics of JS

Array

- An array is a collection of values or a list of values. The values can be of the same type or of different types. To create an array in JavaScript, you can use square brackets [] and inside the square brackets, you can specify the comma-separated list of values.
- `let KhizerFavLanguage = ["JavaScript","Python"];`

Basics of JS

Object

- In JavaScript, an object is used to store a collection of data. An object is a collection of key:value pairs, where pairs of key-value are separated by commas and enclosed within curly braces {}.

Remember
Dictionaries?