Pakistan Blockchain Institute

MODULE-1

JAVASCRIPT CRASH COURSE

Class-1

Edversity.

Raja Rizwan Saleem
Lead Blockchain Trainer

Benefits Overview

Why We Use Computers

Enhancing Productivity and Connectivity

Communication

Computers facilitate seamless information exchange and collaboration, enhancing team interactions and project coordination.

Accuracy

Computers perform calculations and operations with precision, minimizing human errors and ensuring correctness.

Efficiency

Computers streamline tasks and processes, allowing for faster completion and resource optimization.

Reliability

Computers provide consistent performance and operation, reducing the likelihood of errors or downtime.

INTRODUCTION TO PROGRAMMING

- Computers have become part of our lives. We all have different reasons for wanting or needing to use computers.
- Computers can make our jobs become easier. They can be used for communication purposes (internet), to store and calculate data and to write up massive documents multiple times while only needing to write it up once

INTRODUCTION TO PROGRAMMING

Programming languages are the vocabulary and set of grammatical rules for instructing a computer to perform specific tasks. There are many different types of programming languages each having a unique set of keywords (words that it understands) and a special syntax (grammar)for organising program instructions.

Pakistan Blockchain Institute

WHY WE DEVELOP SOFTWARE APPLICATIONS

WHY WE DEVELOP SOFTWARE APPLICATION

- To solve our real world problem by converting it into computer program
- We convert our real world scenario/situation into software application
- So that it can be done more efficiently, reliably and accurately

REAL WORLD PROBLEM/ SITUATION

Letter

SOFTWARE SOLUTION

EMAIL

REAL WORLD PROBLEM/ SITUATION

Accounting Books

SOFTWARE SOLUTION

Excel/QuickBook

REAL WORLD PROBLEM/ SITUATION

Sorting/Arranging information

SOFTWARE SOLUTION

Excel/QuickBook

REAL WORLD PROBLEM/ SITUATION

Driving

SOFTWARE SOLUTION

Driverless Car

REAL WORLD PROBLEM/ SITUATION

Networking

SOFTWARE SOLUTION

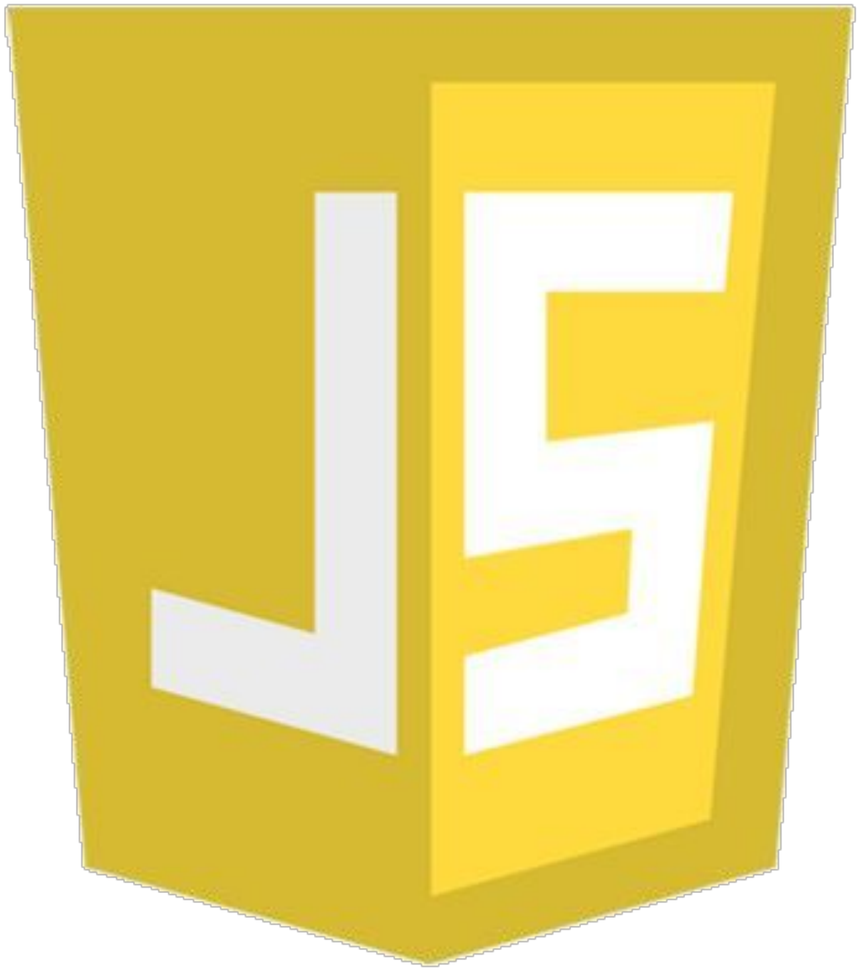
Facebook

Pakistan Blockchain Institute

HOW DO WE COMMUNICATE WITH COMPUTERS

HOW DO WE COMMUNICATE WITH COMPUTER

- Computer acts like our servent
- It will do whatever we ask computer to do.
- But the problem is computer don’t understand what we say
- It does not understand our plain English language
- Computer understands only 0s and 1s



JAVASCRIPT

BOOK WE

WILL

FOLLOW

BASIC REQUIREMENTS TO BUILD A WEBSITE

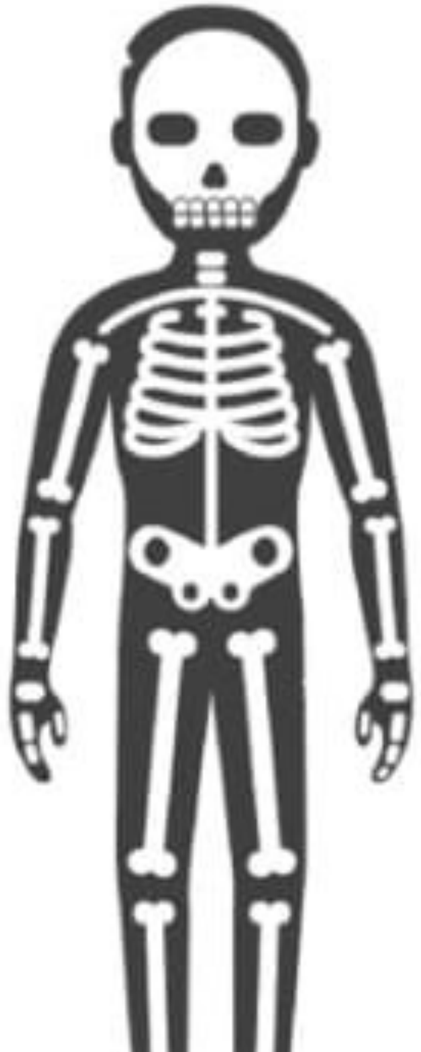
HTML

CSS

JAVASCRIPT

WEBSITE LANGUAGES (HTML, CSS, JS)

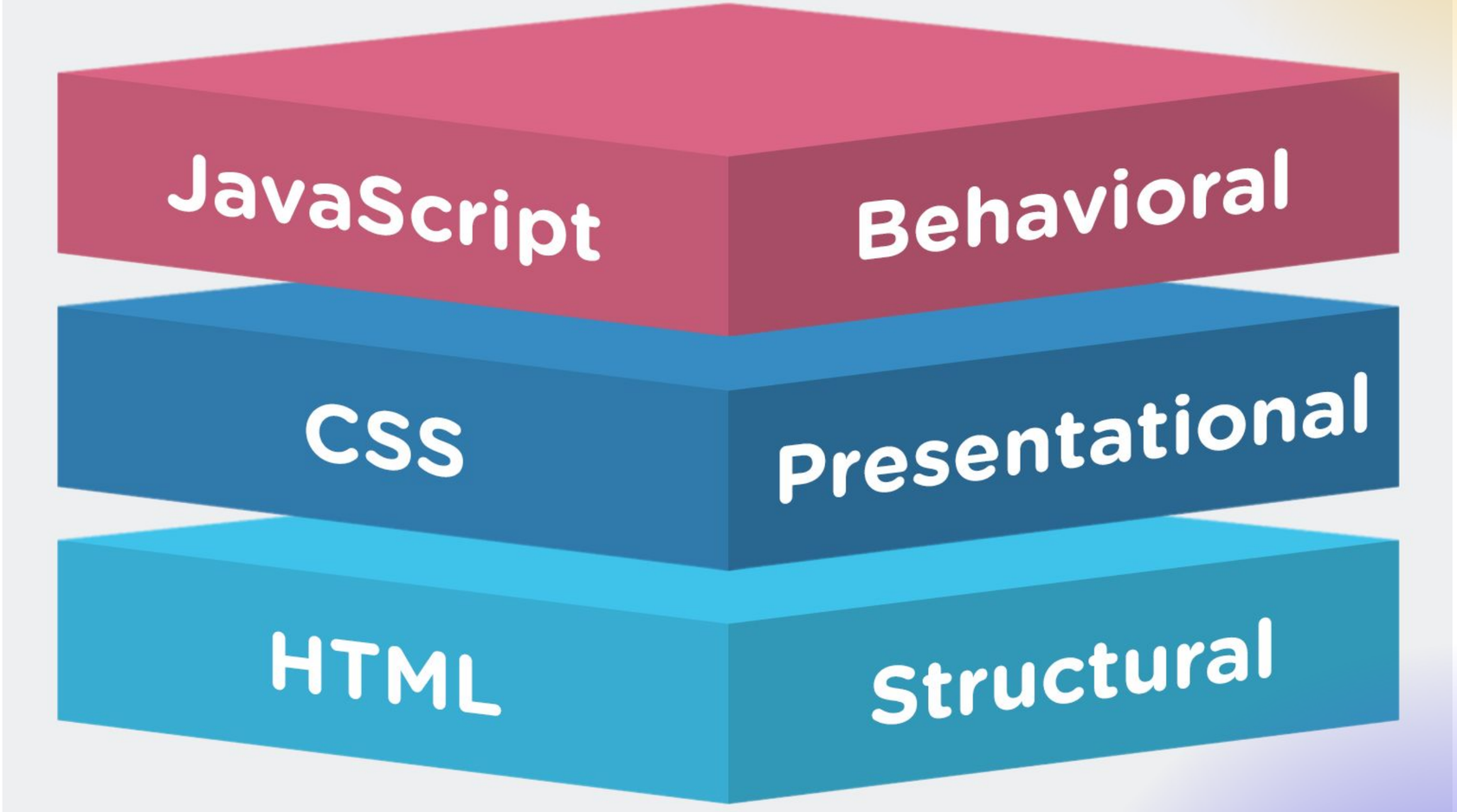
HTML the Skeleton



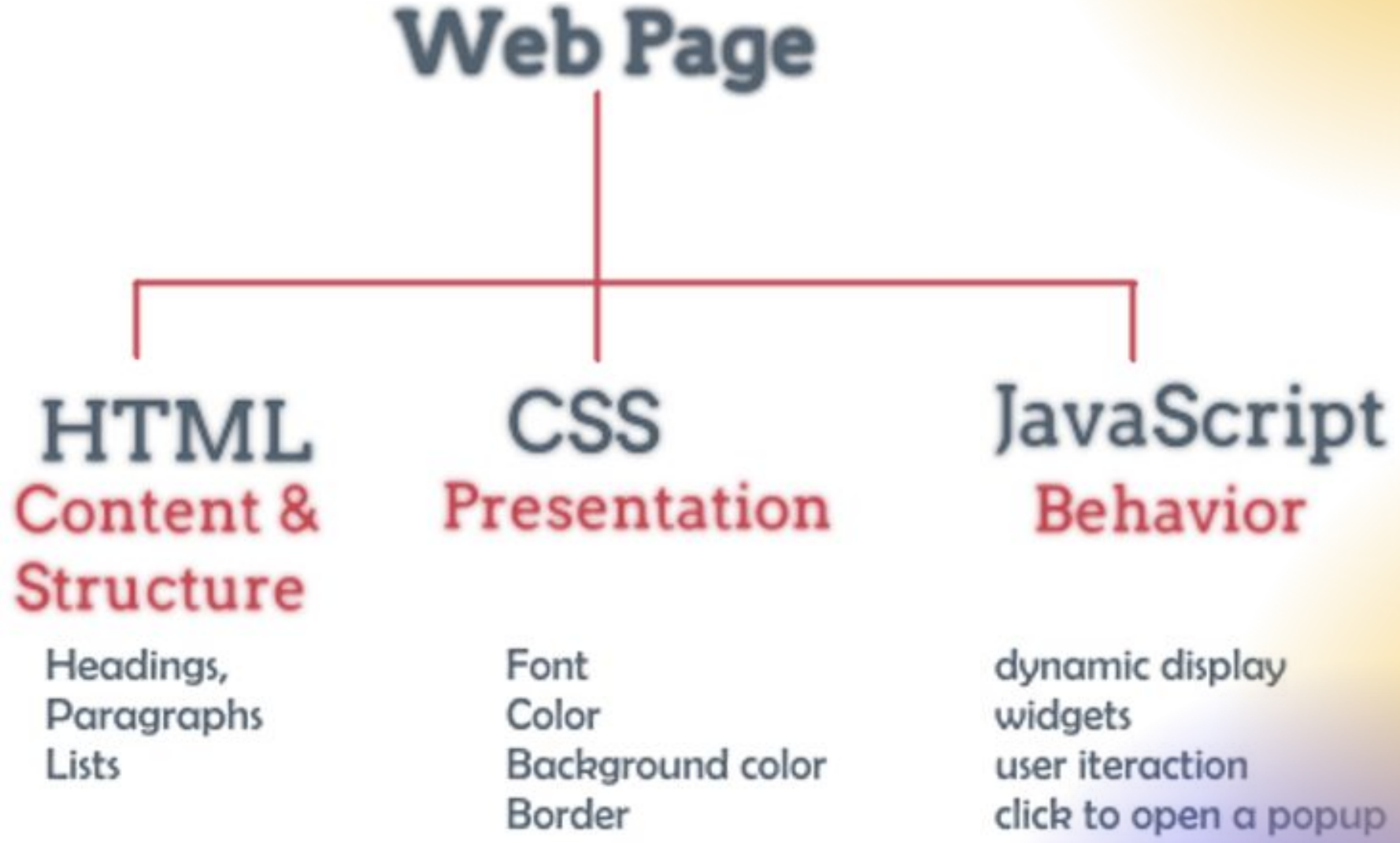
CSS the Skin

Javascript the Brain

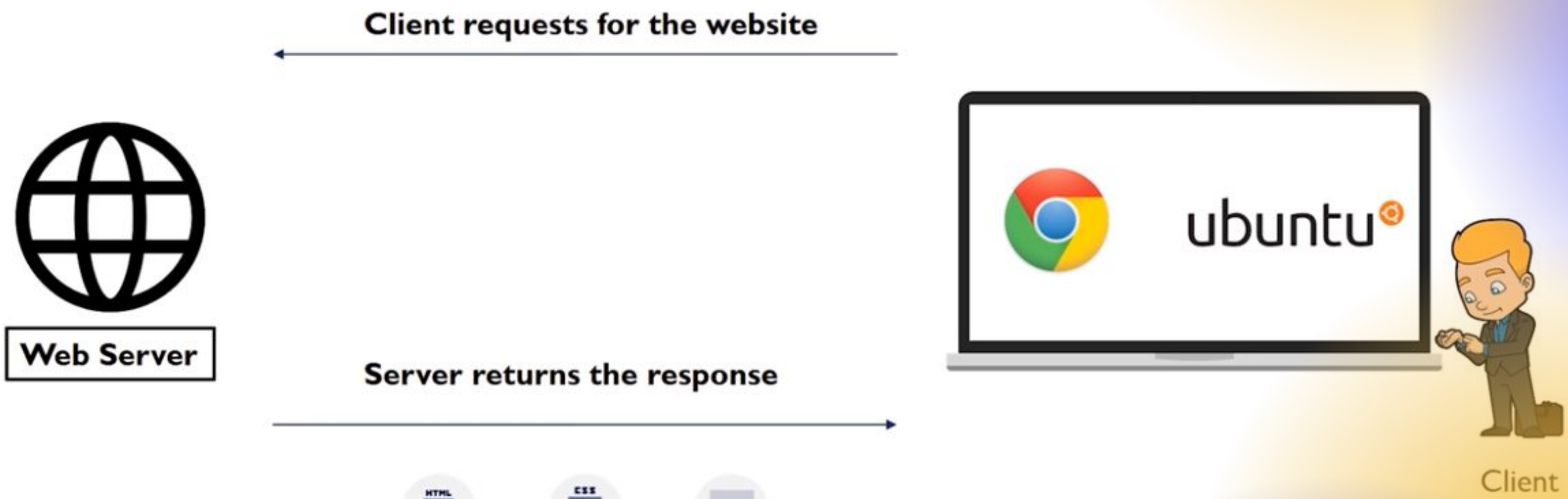
WEBSITE LANGUAGES, (HTML, CSS, JS)



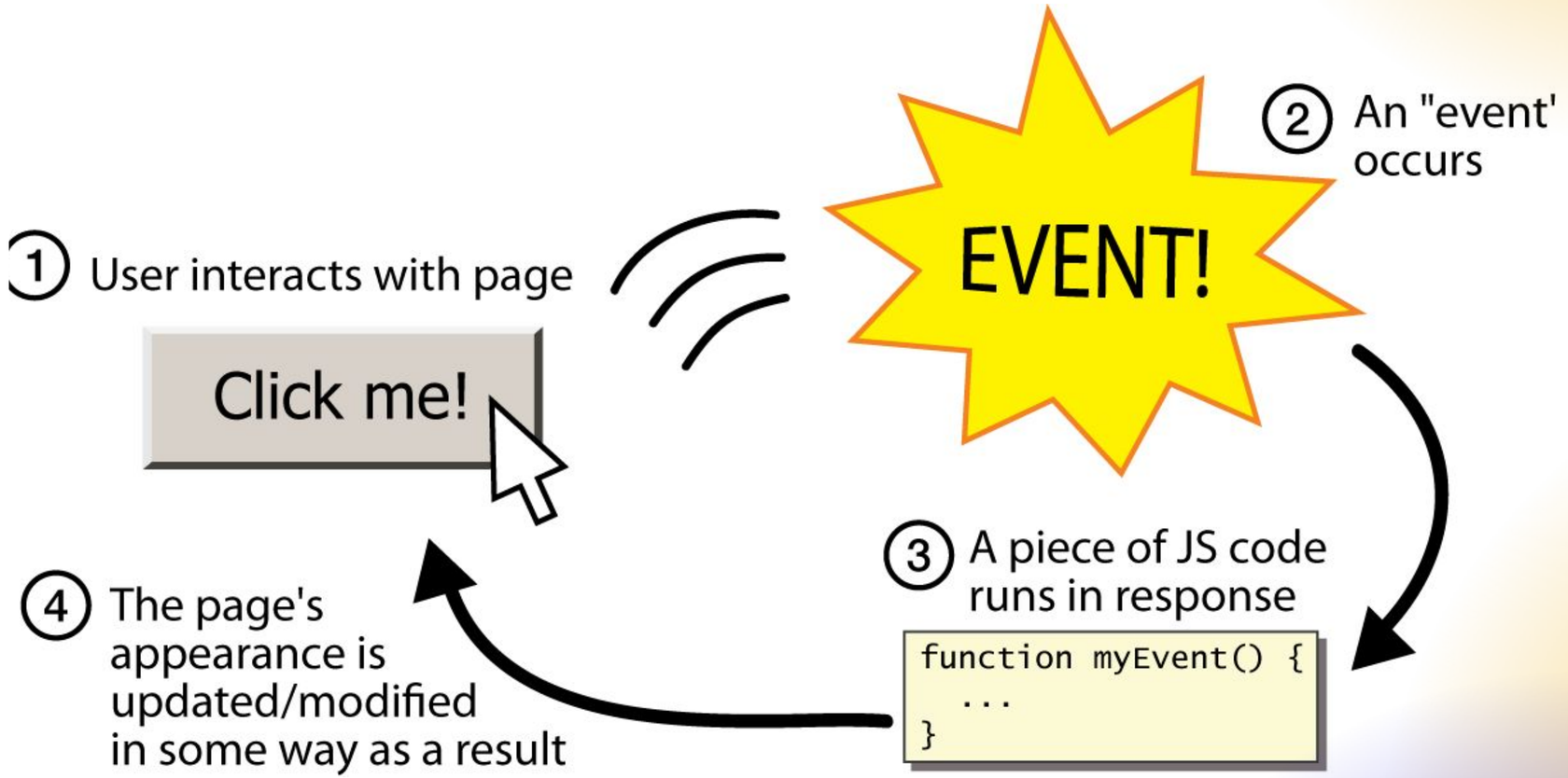
WEBSITE LANGUAGES (HTML, CSS, JS)



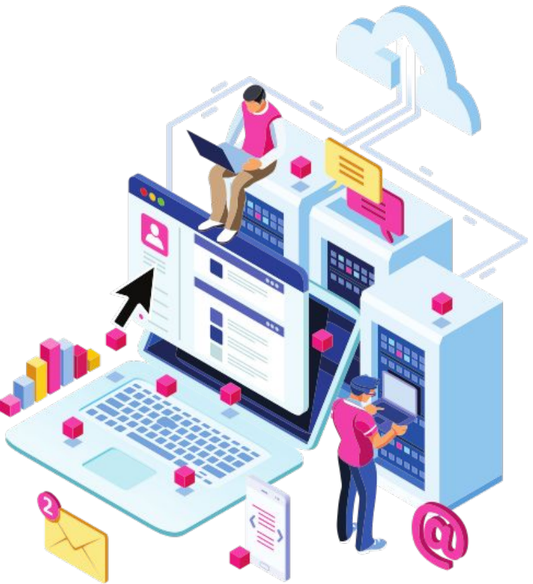
CLIENT SIDE VS SERVER SIDE

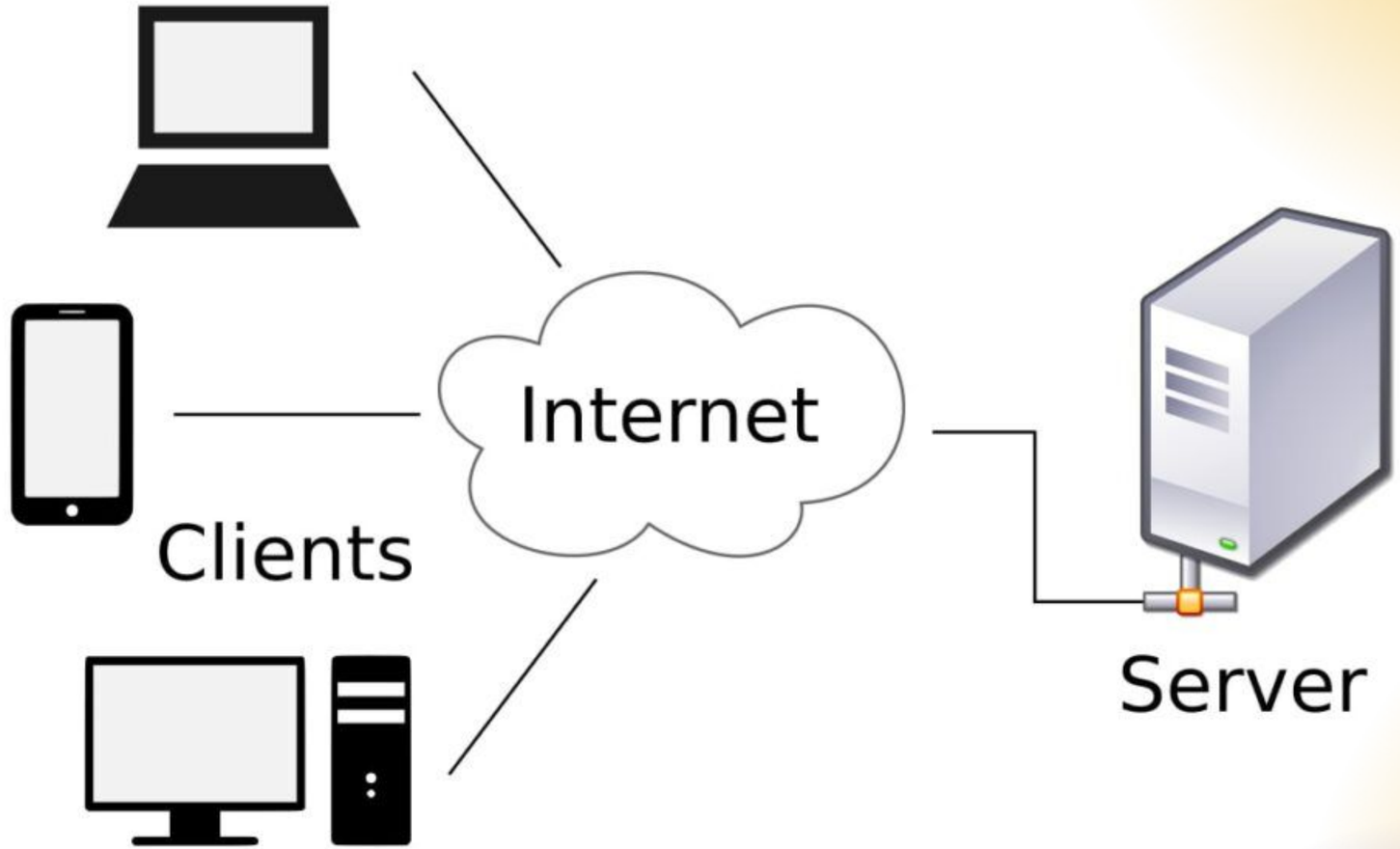


EVENT DRIVEN PROGRAMMING



CLIENT SIDE VS SERVER SIDE



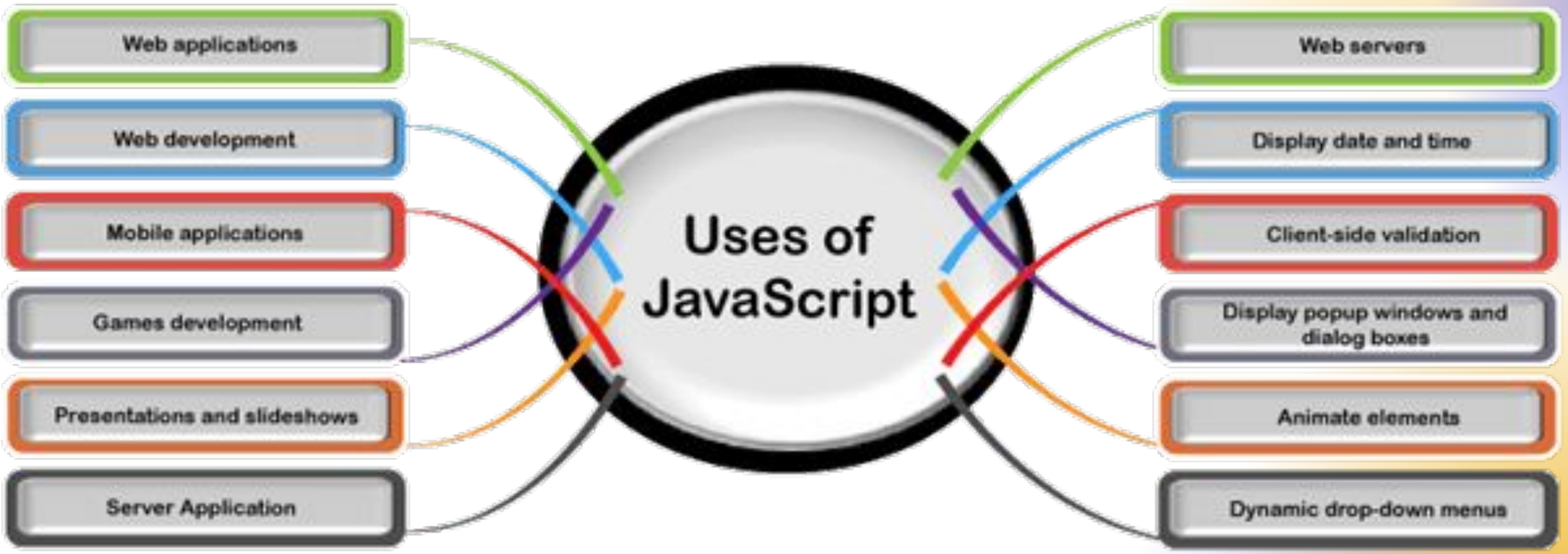


Client Server Network

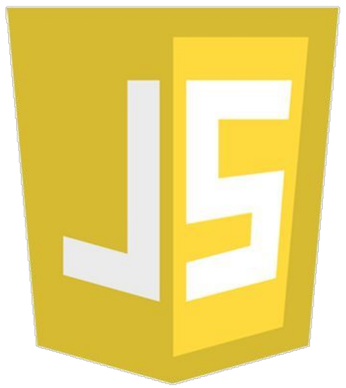
Pakistan Blockchain Institute

TODAY JAVASCRIPT IS MOST USED language and it has advantage to learn single language that can be used on both client and server

USES OF JAVASCRIPT



LETS DEEP DIVE IN WORLD OF



JAVASCRIPT

WHAT IS JAVASCRIPT

- JavaScript, which is abbreviated as JS, is a programming language that conforms to the ECMAScript specification.
- It was initially created to make web pages alive.
- It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages.
- JavaScript has no link with Java programming language. It is a fully independent language with its own specification.

WHAT IS JAVASCRIPT

- JavaScript is a very powerful client-side scripting language.
- JavaScript is used mainly for enhancing the interaction of a user with the webpage.
- You can make your webpage more lively and interactive, with the help of JavaScript.
- JavaScript is also being used widely in game development and mobile application development.

WHAT CAN BUILD USING JAVASCRIPT?

- **Websites:** JavaScript helps us to add behavior of our website. It helps users to interact with the website. For eg. clicking on buttons, saving details, uploading details on the website, etc.
- **Web Servers**: We can make robust server applications using JavaScript. To be precise we use JavaScript frameworks like Node.js and Express.js to build these servers.
- **Game Development:** In Game Development industry, JavaScript is used widely. With the addition of HTML5 Canvas, it's now possible to make 2D and 3D games in JavaScript very efficiently.
- **3D Drawings:** JavaScript in addition with HTML Canvas is used to make three-dimensional graphics.
- **Mobile Apps:** Mobile applications are the most popular modes of communicating these days. JavaScript also used to design mobile applications. There are many JavaScript frameworks using which we can make android, IOS, and hybrid apps.

INSTALLATION

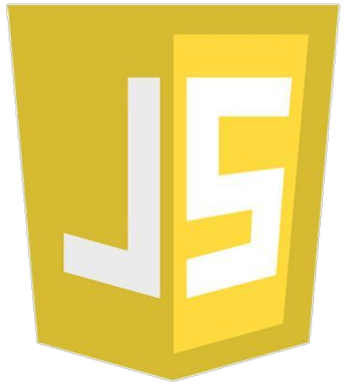
- To work with JavaScript you don’t need any software
- You just need to have Browser installed on your machine and you can use any text editor to write code.
- To manage projects and organize files we use IDE
- We will use Microsoft’s Visual Studio Code
- To run JavaScript on server side we will use Node.js

INSTALLATION AND SETUP CODING ENVIRONMENT

- Download and install Visual Studio code
  - https://code.visualstudio.com

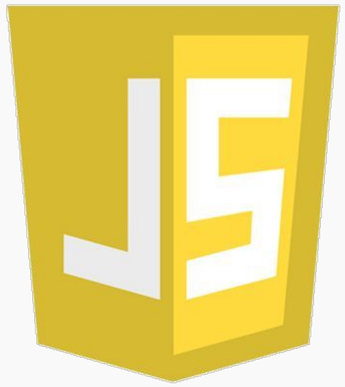
- Download install Node.js
  - https://nodejs.org/en/

- Use default setting while installing both tools



DEMO

HOW TO USE VISUAL STUDIO CODE



Demo
Create First JavaScript Example

JavaScript Integration in HTML

index.html File

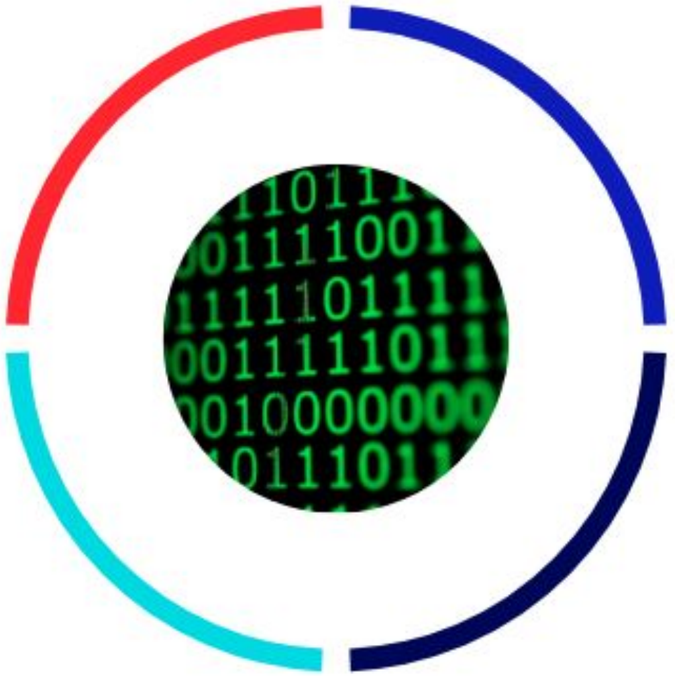
Including JavaScript in an HTML File

Output Display

Demonstrating 'Hello World' output in the body section of the HTML file.

HTML Structure

The basic structure of an HTML file with tag linking to JavaScript file.



JavaScript Execution

How JavaScript code is executed within the HTML file.

Script Tag

Usage of the tag to include external JavaScript file in the HTML document.

Methods to Implements Javascript

You can add JavaScript code in an HTML document by **employing the dedicated HTML tag <script> that wraps around JavaScript code**. The <script> tag can be placed in the <head> section of your HTML or in the <body> section, depending on when you want the JavaScript to load.

Methods to Implements Javascript

a. Inpage Javascript
b. External Javascript

Methods to Implements Javascript

- **Internal JS:** We can add JavaScript directly to our HTML file by writing the code inside the <script> tag. The <script> tag can either be placed inside the <head> or the <body> tag according to the requirement.
- **External JS:** We can write JavaScript code in other file having an extension .js and then link this file inside the <head> tag of the HTML file in which we want to add this code.

Methods to Implements Javascript

Inpage Javascript

```
<!DOCTYPE html>
<html>
        <head>
            <script>
            </script>
        </head>
<body>

        <h1>Pakistan Blockchain Institute</h1>
            <script>
            </script>

</body>
</html>
```

Methods to Implements Javascript

Inpage Javascript

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>
        Basic Example to Describe JavaScript
    </title>
</head>

<body>

    <!-- JavaScript code can be embedded inside
        head section or body section -->
    <script>
        console.log("Welcome to Ethereum 2.0 Course");
    </script>
</body>
</html>
```

Initial Code and Setup

1. Open Visual Studio code and create two files
    a. index.html
    b. Index.js

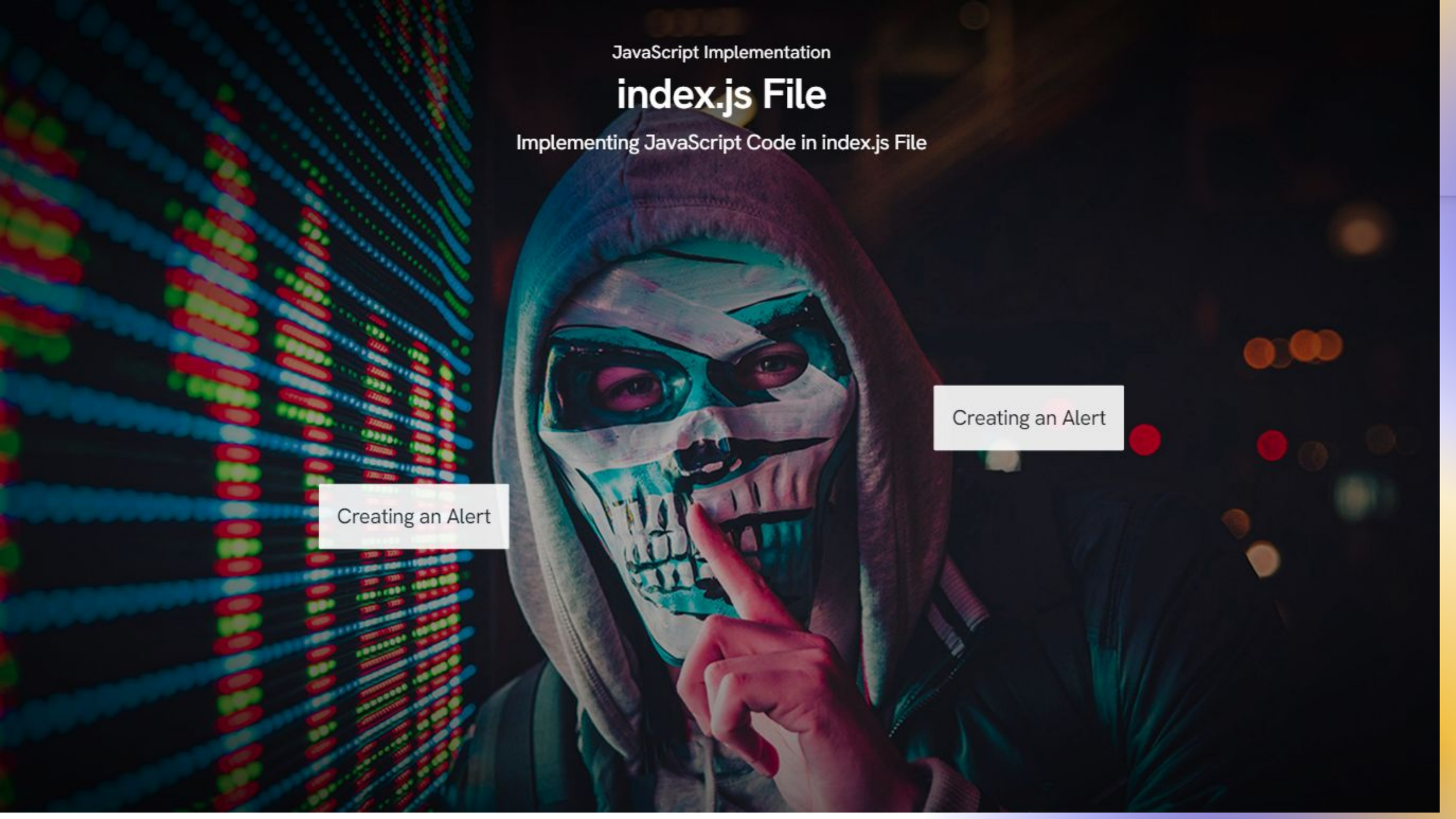
index.html file

External Javascript

```
<html>
    <head>
        <script src="index.js"></script>
    </head>
    <body>
        Hello World
    </body>
</html>
```

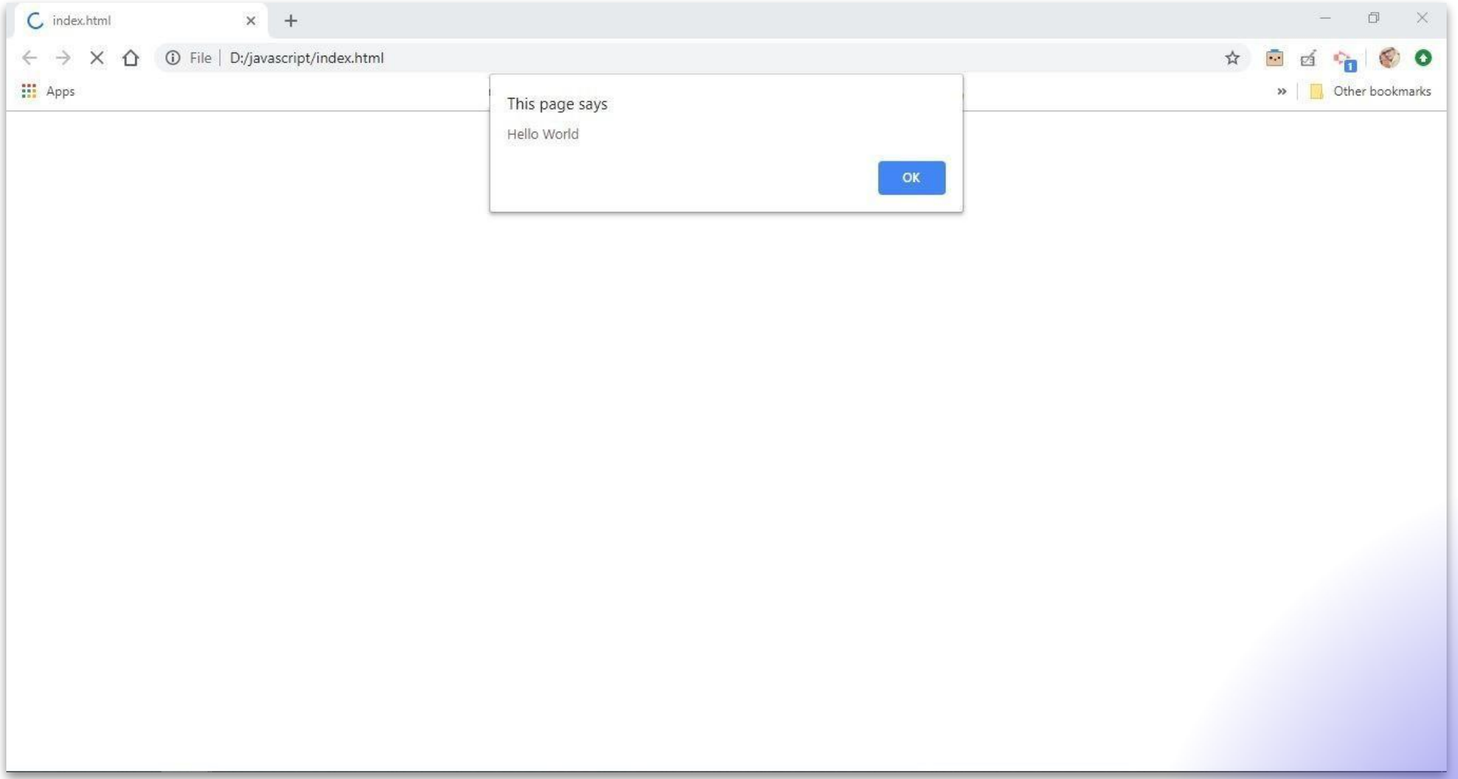
index.js file

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


JavaScript Implementation
index.js File
Implementing JavaScript Code in index.js File
Creating an Alert
Creating an Alert

Initial Code and Setup

1. Open file in browser and you will see alert in browser window



Alerts

1. An alert is a box that pops up to give the user a message.
2. Here's code for an alert that displays the message "Thanks for your input!"

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

3. alert is a keyword
4. The quoted text "Thanks for your input!" is called a text string or simply a string.

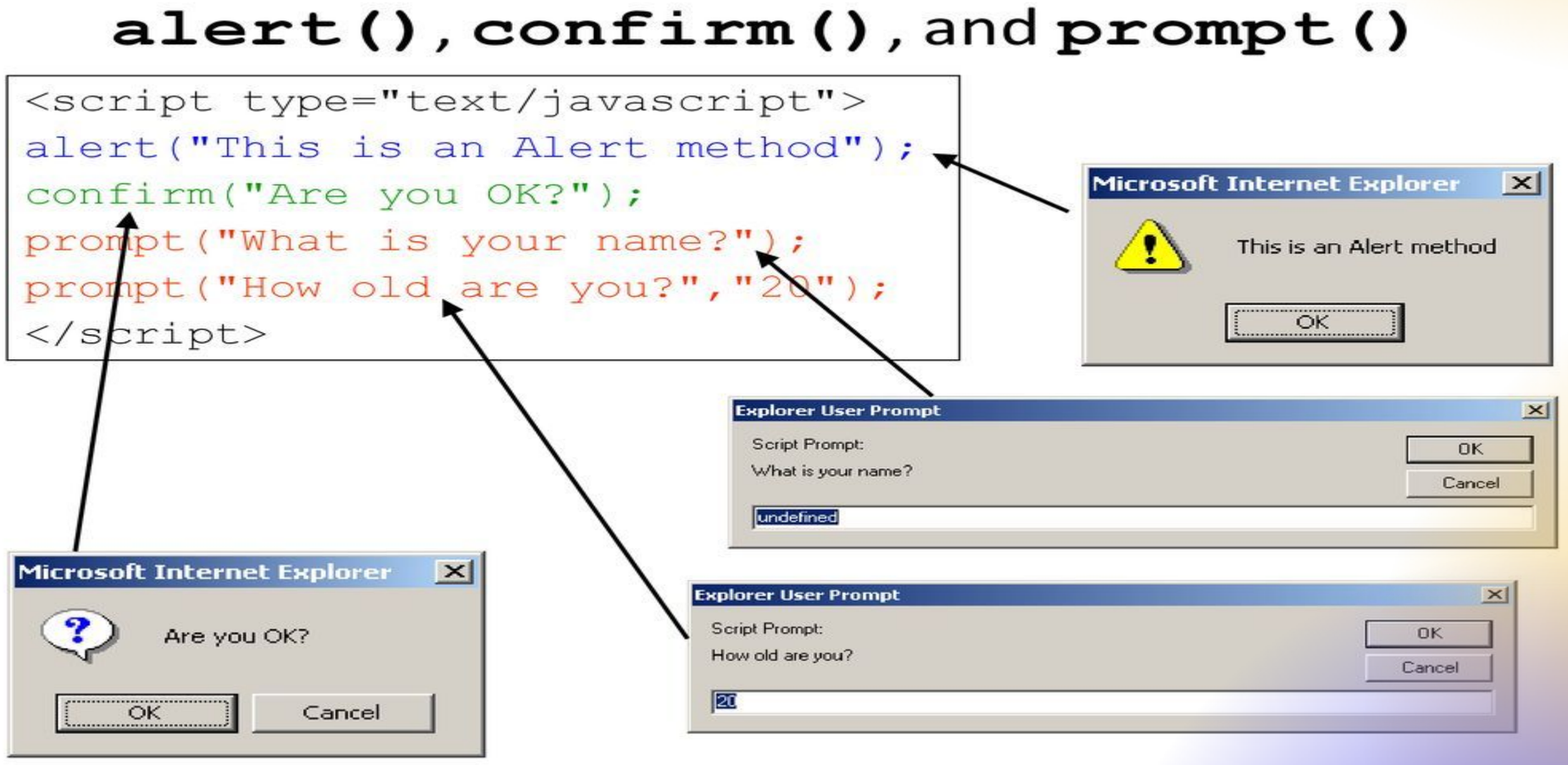
Alerts

1. ‘window.alert’ and ‘alert’ are the same
2. window is object in browser which contains many other objects and properties, and alert is one of them.
3. alert is a function that take any input and display it in popup to user
4. Alerts are not available when you are working with server side JavaScript in Node.js

Alerts Exercise

```
alert("JavaScript Alert! \n welcome to JS
land \n Happy coding!");
```

Alerts, Confirm and Prompt Exercise



Alerts, Confirm and Prompt Exercise

alert() and confirm()

```
alert("Text to be displayed");
```

- Display a message in a dialog box.
- The dialog box will block the browser.

```
var answer = confirm("Are you sure?");
```

- Display a message in a dialog box with two buttons: "OK" or "Cancel".
- `confirm()` returns `true` if the user click "OK". Otherwise it returns `false`.

Alerts, Confirm and Prompt Exercise

prompt()

```
prompt("What is your student id number?");
prompt("What is your name?", "No name");
```

- Display a message and allow the user to enter a value
- The second argument is the "default value" to be displayed in the input textfield.
- Without the default value, "undefined" is shown in the input textfield.

- If the user click the "OK" button, **`prompt()`** returns the value in the input textfield as a string.
- If the user click the "Cancel" button, **`prompt()`** returns null.

How to open Consol.log in chrome browser

We can open our console in the web browser by using: *Ctrl + Shift + K* or by Right-click on any webpage, click *Inspect*, and then we can see the innards of that site; its source code, the CSS that form its design, the JavaScript code that powers animations, and more. It has a console option as well, where we can run our JavaScript code.

Console.log

1. console.log is function that write message on console/terminal
2. Objective of console.log is to create logs for debugging
3. Instead of displaying text to user it shows output in browser’s developer tool
4. Also when working with server side javascript, we can use console.log for logging and output will be in terminal

Console.log

Console.log():-

This method is used to log(print) the output to the console. We can put anything inside the log(). It can be an array, object, string, boolean, etc.

Console.log

```
console.log("Hello World");
console.log(2+8);
```

document.write

1. For testing purpose you can use document.write to display message or text in browser window

```
document.write("Hello Word");
document.write(2+8);
```

This will be displayed in browser window

JavaScript Variables

What Variable is in JS?

In any programming language, we typically do lots of calculations. The calculation results are stored in the computer's memory. Just like human memory, the memory of the computer also consists of millions of cells. The calculated values are stored in these memory cells. To make the usage and retrieval of these values easy, these memory locations are given names. The names given to these locations are called variables.

Fundamentals

Variables in JavaScript

Understanding the Role of Variables in JavaScript Development

Variables Hold Data

Variables in JavaScript serve as containers that can store different types of data and can be modified during the program's execution.

1

Declaration with var, let, or const

JavaScript variables can be declared using keywords like var, let, or const, each with its own scope and behavior.

2

Local and Global Variables

JavaScript supports both local and global variables. Local variables are defined within functions and have function-level scope, while global variables are accessible throughout the program.

3

What Variable is in JS?

It holds the data or information which can be changed anytime. JavaScript use reserved keyword var to declare variables. In JavaScript, there are two types of variable and also it tells you where in your program you are allowed to use the variables and functions that you've defined.

Local Variable:

When you use JavaScript, local variables are variables that are defined within functions. They have local scope, which means that they can only be used within the functions that define them.

Global Variable:

In contrast, global variables are variables that are defined outside of functions. These variables have global scope, so they can be used by any function without passing them to the function as parameters.

Variables

1. Variable means anything that can vary.
2. JavaScript includes variables which hold the data value and it can be changed anytime.
3. Variable is the name of the storage location.
4. When we want to save some data, we store it in a variable
5. JavaScript uses reserved keyword **var** to declare a variable. A variable must have a unique name. You can assign a value to a variable using **equal to** (=) operator when you declare it or before using it.

What Variable is in JS?

- Before the advent of ES6, var declarations were used to declare a variable.
- The properties of var is that it has visibility linked to the function to which it belongs.
- We can change its value, and it can be redeclared in the same scope.
- Scope means where these variables are available for use. There are two types of scope, local and global.
- Var declarations are globally scoped, and when they are defined inside a function, they are locally scoped.

What Variable is in JS?

When the variable is declared, the JavaScript engine assigns it a memory or space. Because of this, once a variable is declared, it takes a value of undefined even before the assignment. We assigned data to the variable by using the assignment operator "=". Datatypes in JavaScript are:

1. Number i.e., 11,23,45,6
2. Strings, i.e., "Hello World."
3. Boolean, i.e., true, false
4. Undefined
5. Null
6. Any of the complex data structures (Array and Objects)

What Variable is in JS?

There are 3 ways to declare a JavaScript variable:

- Using var
- Using let
- Using const

Variables in JS (var, let & const)?

- var and let create variables that can be reassigned another value.
- const creates "constant" variables that cannot be reassigned another value.
- developers shouldn't use var anymore. They should use let or const instead.
- if you're not going to change the value of a variable, it is good practice to use const.

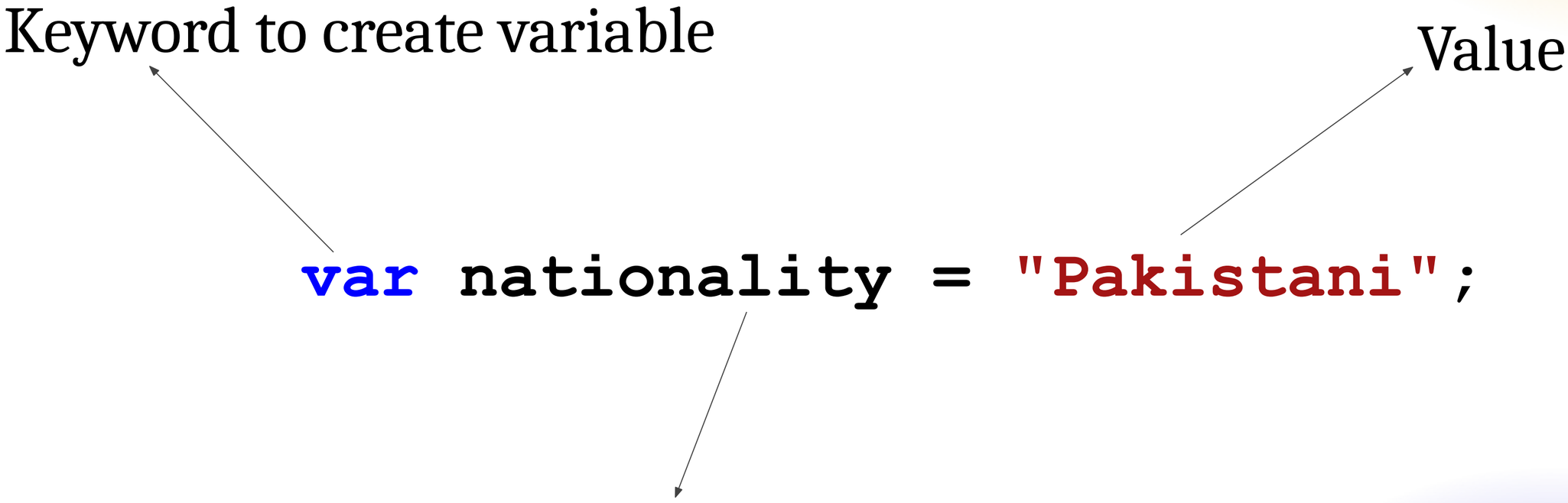
When to use JavaScript const?

As a general rule, always declare a variable with `const` unless you know that the value will change.

Use `const` when you declare:

- A new Array
- A new Object
- A new Function

Variables


Keyword to create variable
Value
var nationality = "Pakistani";
Variable name/identifier

Variables

```
var nationality = "Pakistani";

var age = 25;

var isFeePaid = true;

var weight = 60.55;
```

Declaration and Initialization

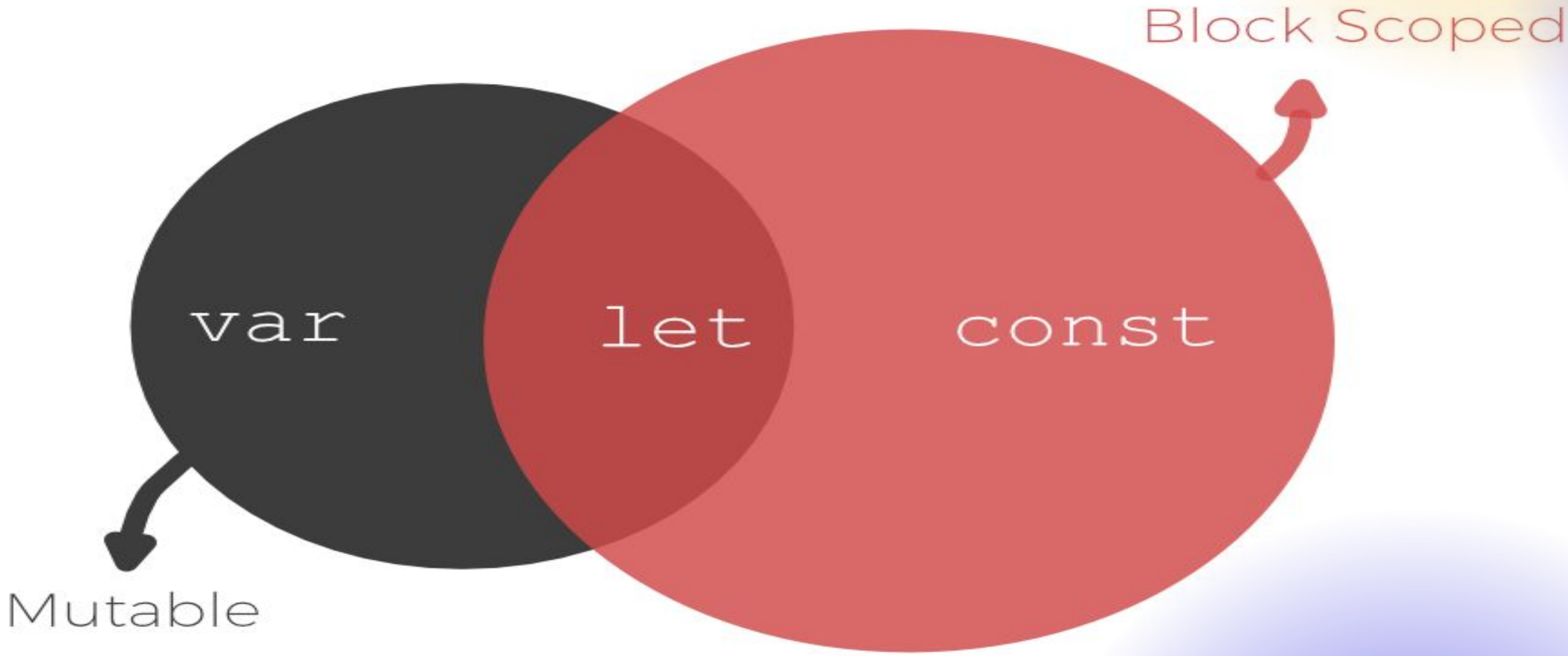
1. You can declare and initialize in single line or you can do that in two line

```
var age = 25;   OR

var age;     -- Declaration
age = 25;    -- Initialization
```

Only declaration will leave variable *undefined*

Variables' Types



Variables’ Types

Var:-

The *var* is the oldest keyword to declare a variable in JavaScript.

Scope:

Global scoped or function scoped. The scope of the *var* keyword is the global or function scope. It means variables defined outside the function can be accessed globally, and variables defined inside a particular function can be accessed within the function.

Variables' Types - Plz practice

```
// Declaring a variable using var
var name = "Rizwan";
console.log(name); // Output: Rizwan

// Reassigning the variable
name = "Edversity";
console.log(name); // Output: Edversity

// var is function-scoped
function greet() {
    var greeting = "Hello";
    console.log(greeting); // Output: Hello
}
greet();

// var can be redeclared
var name = "PBI";
console.log(name); // Output: PBI
```

Variables' Types - Plz practice

```
var a = 10
function f() {
    console.log(a)
}
f();
console.log(a);
```

Output:

```
10
10
```

Example 1: Variable 'a' is declared globally. So, the scope of the variable 'a' is global, and it can be accessible everywhere in the program. The output shown is in the console.

Variables’ Types

let:-

- **Let** is a keyword used to declare variables in javascript that are block scoped and this type let was introduced in ES6.
- It shares a lot of similarities with var, but unlike var, it has scope constraints. Its declaration and assignment are similar to var.
- The purpose of introducing let is to resolve all issues posed by variables scope, which developers face during development.
- We can change their values, but they cannot be redeclared in the same scope, unlike var.
- "let" helps us by making it easier to see where variables live in our code and make our code cleaner and easier to read.

Variables' Types

```
// Declaring a variable using let
let age = 25;
console.log(age); // Output: 25

// Reassigning the variable
age = 30;
console.log(age); // Output: 30

// let is block-scoped
if (true) {
    let city = "New York";
    console.log(city); // Output: New York
}
// console.log(city); // Uncaught ReferenceError: city is not defined

// let cannot be redeclared within the same scope
// let age = 35; // Uncaught SyntaxError: Identifier 'age' has already been declared
```

Variables' Types

The *let* keyword is an improved version of the *var* keyword.

Scope:

block scoped: The scope of a *let* variable is only block scoped. It can't be accessible outside the particular block ({block}). Let's see the below example.

Variables' Types = plz practice

```
let a = 10;
function f() {
    let b = 9
    console.log(b);
    console.log(a);
}
f();
```

Output:

```
9
10
```

Variables' Types – Practice Plz

```
let a = 10
if (true) {
    let a = 9
    console.log(a) // It prints 9
}
console.log(a) // It prints 10
```

Output:

```
9
10
```

Variables' Types

```
let a = 10;
function f() {
    if (true) {
        let b = 9

        // It prints 9
        console.log(b);
    }
     // It gives error as it
    // defined in if block
    console.log(b);
}
f()
// It prints 10
console.log(a)
```

Output:

```
9

ReferenceError: b is not defined
```

Example 2: The code returns an error because we are accessing the *let* variable outside the function block. The output is shown in the console.

Variables' Types

Const:-

- Const is also introduced in ES6.
- It is a variable type assigned to data whose value cannot and will not be changed throughout the program.
- Const is more strict as compared to var and let.
- Const is also limited to the scope in which it operates.
- We declare const just like var and let.
- We use const when we are sure a variable will not be redeclared.

Variables’ Types

The *const* keyword has all the properties that are the same as the *let* keyword, except the user cannot update it.

Scope:

block scoped: When users declare a *const* variable, they need to initialize it, otherwise, it returns an error. The user cannot update the *const* variable once it is declared.

Variables' Types

Example 1: We are changing the value of the const variable so that it returns an error. The output is shown in the console.

```
const a = 10;
function f() {
    a = 9

console.log(a)
}
f();
```

Output:

```
TypeError:Assignment to constant variable.
```

Differences between var, let & const

var	let	const
The scope of a *var* variable is functional scope.	The scope of a *let* variable is block scope.	The scope of a *const* variable is block scope.
It can be updated and re-declared into the scope.	It can be updated but cannot be re-declared into the scope.	It cannot be updated or re-declared into the scope.
It can be declared without initialization.	It can be declared without initialization.	It cannot be declared without initialization.
It can be accessed without initialization as its default value is “undefined”.	It cannot be accessed without initialization otherwise it will give ‘referenceError’.	It cannot be accessed without initialization, as it cannot be declared without initialization.

Pakistan Blockchain Institute

MODULE-1

THANK YOU

END OF LECTURE-1