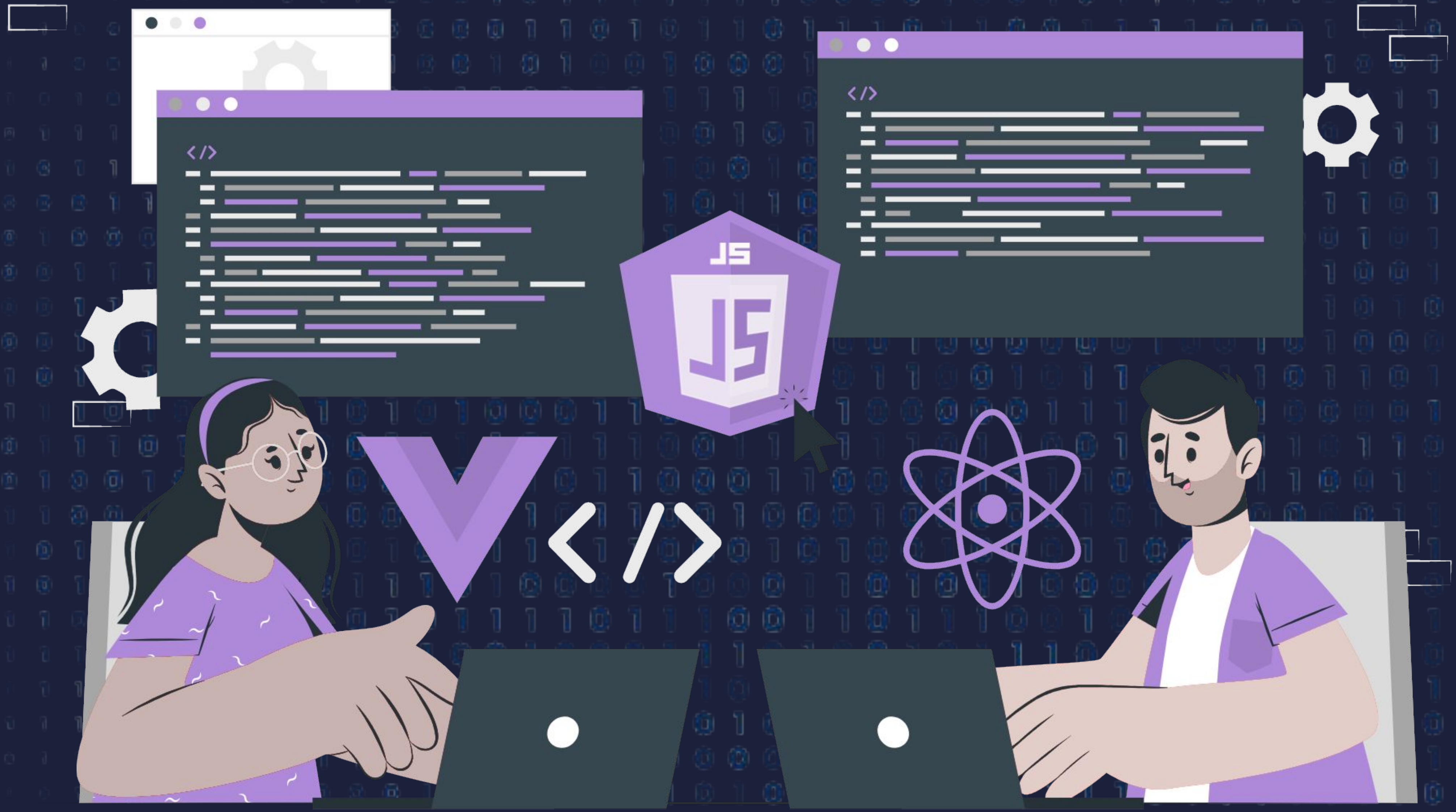
PW SKILLS

Execution Context



Lecture CheckList

1. Introduction.
2. Execution Context.
3. Types of Execution Context.
4. Eval() function.
5. Creation of execution context.

Introduction

Execution context is an important concept that helps us understand how code runs in JavaScript. Execution context refers to the environment in which a piece of code is executed. This environment includes information about variables, functions, objects, and whatever the js engine will execute. Understanding the execution context is essential because it helps us write code that runs efficiently and accurately. By managing the execution context, we can control the behavior of our code and ensure that it executes in the way we intend it to. We'll be exploring the execution context in more detail, in order to gain a better understanding of this fundamental concept in JavaScript.

Execution Context

Execution context is the environment in which a piece of code is executed in JavaScript. It includes all the information the code needs to run properly, such as variables, functions, and objects. By managing the execution context, we can control the behavior of their code and ensure that it executes correctly.

Types of the execution context

There are three types of execution context in JavaScript:

1. Global execution context.
2. Local/Function execution context.
3. Eval execution context.

Eval() Function

eval() is a built-in function in JavaScript that allows you to execute a string of code as if it were part of the current script. When the eval() function is called, it takes a string argument that contains the code to be executed.

In modern JavaScript, the use of eval() is discouraged, and there are usually safer alternatives to achieve the same result. Instead of using eval(), it is generally recommended to use other JavaScript features like functions or object literals.

Creation of Execution Context

Execution Context is created in 2 phases

1. Memory Creation Phase

2. Code Execution Phase

SKILLS

THANK YOU