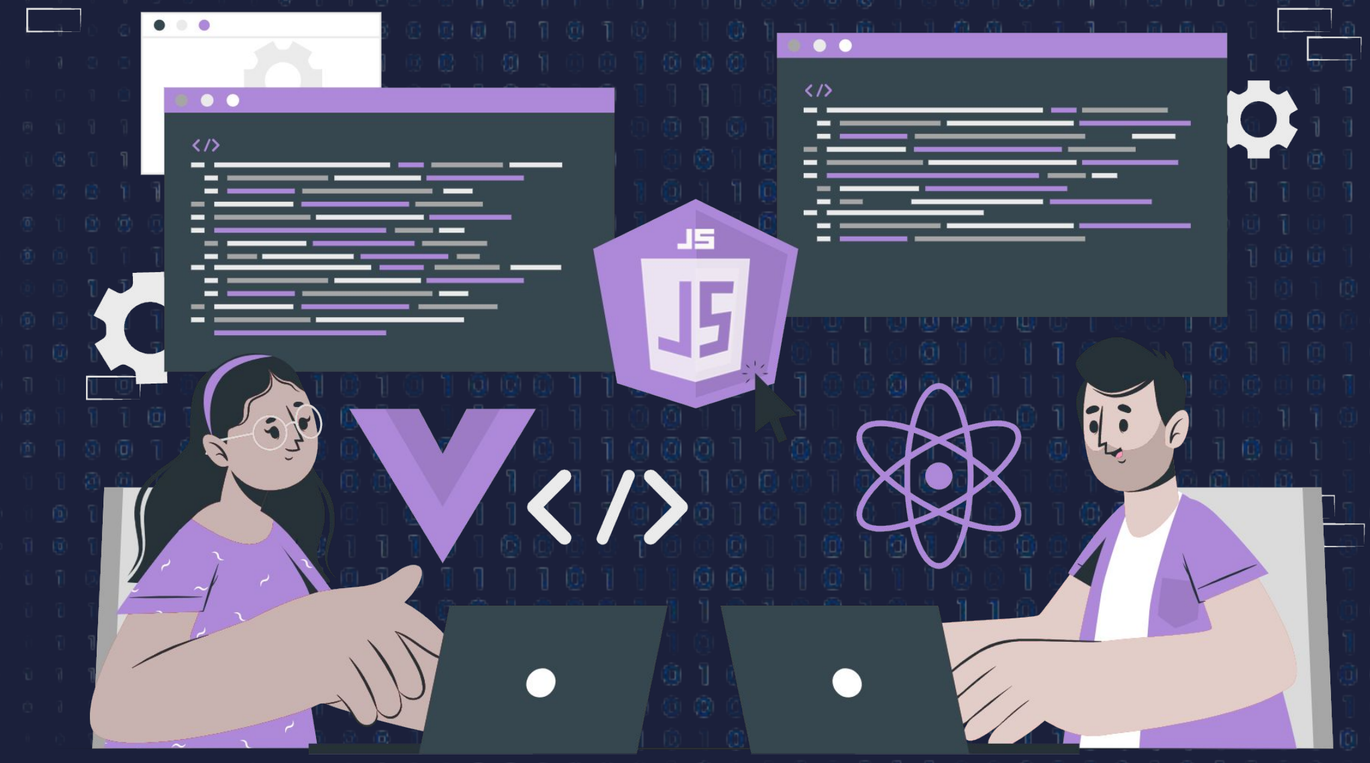
PW SKILLS

Working of function



Lecture CheckList

1. Introduction.
2. Function Hoisting in Javascript.
3. Function Expressions.
4. Arrow functions.

Introduction

Functions are really important in JavaScript programming. They let developers write code that can be used over and over again. Functions in JavaScript can be put in variables, passed as arguments to other functions, and received back as values from other functions. They make it easy to write complicated code by putting everything together and making it easy to understand and fix later.

Function Hoisting in Javascript

n the previous lecture, we looked into variable hoisting. Function hoisting is similar to variable hoisting in JavaScript, as both involve moving the declaration of a variable or function to the top of its scope during the code's creation phase. It gives us the advantage that we can use a function before it is declared.

How does JS Engine look at it?

When the JavaScript engine is given with the code, it first goes through a process called hoisting. During hoisting, the engine moves all function declarations to the top of their respective scopes, allowing you to call them before they are actually defined in the code.

Function Expressions

We all know that function expressions offer us a flexible and powerful way to define functions in JavaScript, allowing us to assign them to variables or pass them as arguments to other functions. But, function expressions, unlike function declarations, are not hoisted to the top of their respective scopes. This means that you cannot call a function expression before it is declared because it is not yet defined in the code.

Arrow functions

From the previous lectures, we know that Arrow functions are a concise way to write functions in JavaScript using the “=>” syntax. They are also a compact way of writing function expressions.

So do you think they can be hoisted? No, arrow functions are not hoisted in JavaScript.

THANK YOU