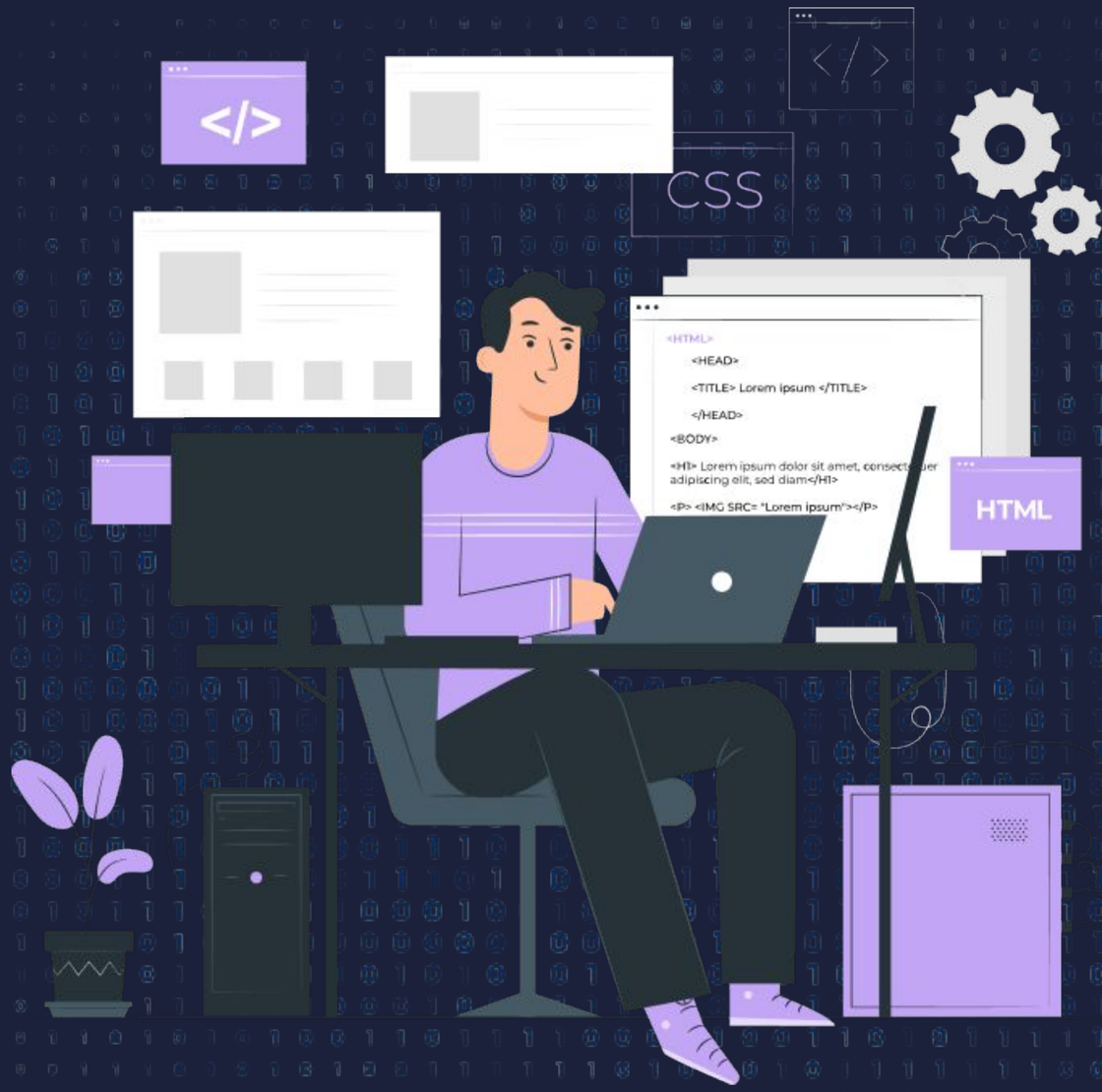
Arrow Function



Lecture CheckList

- Introduction to Arrow functions.
- Arrow Function Syntax.
- Why should developers use arrow functions?

Introduction to Arrow Functions

Arrow functions, introduced in ES6, is a straightforward and compact way of writing JavaScript functions. They improve the structure and readability of our code by providing a more concise syntax.

Arrow Function Syntax.

Arrow functions use a simplified syntax with the “=>” arrow. They don’t require the function keyword, and if the function body contains only a single expression, the curly braces and “return” statements can be omitted. If the function has a single parameter, the parentheses around the parameter can be omitted.

```
(parameter1, parameter2, ..., parameterN )=>{
// function body
return expression; // optional
}
```

Examples of arrow function

```
// 1. One parameter, and a single return statement

const square = x => x * x;
```

```
// 2. Multiple parameters, and a single return expression

const sumOftwoNumbers = (x, y) => x + y;
```

```
// 3. Multiple statements in function expression

const sum = (x, y) => {
  console.log(`Adding ${x} and ${y}`);
  return x + y;
};
```

```
// 4. Returning an object

const sumAndDifference = (x, y) => ({
sum: x + y, difference: x - y });
```

Features of Arrow Function Syntax

- Parentheses are optional in the case of a single parameter.
- Must use parentheses in case of multiple parameters.
- The return keyword is not required for a single return expression in the function body.
- The return keyword is required in case of multiple statements in the function.
- To return an object notation, wrap it with parentheses.

Limitation for arrow function

- Arrow functions cannot access the argument object.
- Arrow functions do not have the prototype property
- Arrow functions cannot be used with the new keyword.
- Arrow functions cannot be used as constructors.
- These functions are anonymous, and it is hard to debug the code

Arrow function vs regular function

Syntax:

- Regular Function: Defined using the function keyword followed by a name, parameter list, and function body.
- Arrow function: Defined using a concise syntax with the “=>” arrow operator, omitting the function keyword and using a more compact structure.

Arguments object

- Regular function: Has access to the arguments object, which contains all the arguments passed to the function.
- Arrow function: Does not have its own arguments object instead it inherits the arguments object from the enclosing scope.

Binding of “this”

- Regular function: Has its, own “this” value, which is determined by how the function is called. The value of “this” can change dynamically depending on the context of invocation.
- Arrow function: Inherits the “this” value from the enclosing lexical scope. It does not have it's own “this” binding and captures the “this” value of the surrounding context.

Use for new keyword

- Regular function: can be used as constructor function with the new keyword to create an instance of objects
- Arrow function cannot be used as a constructor function Attention to use new with an arrow function will result in a runtime error.


SKILLS
THANK YOU