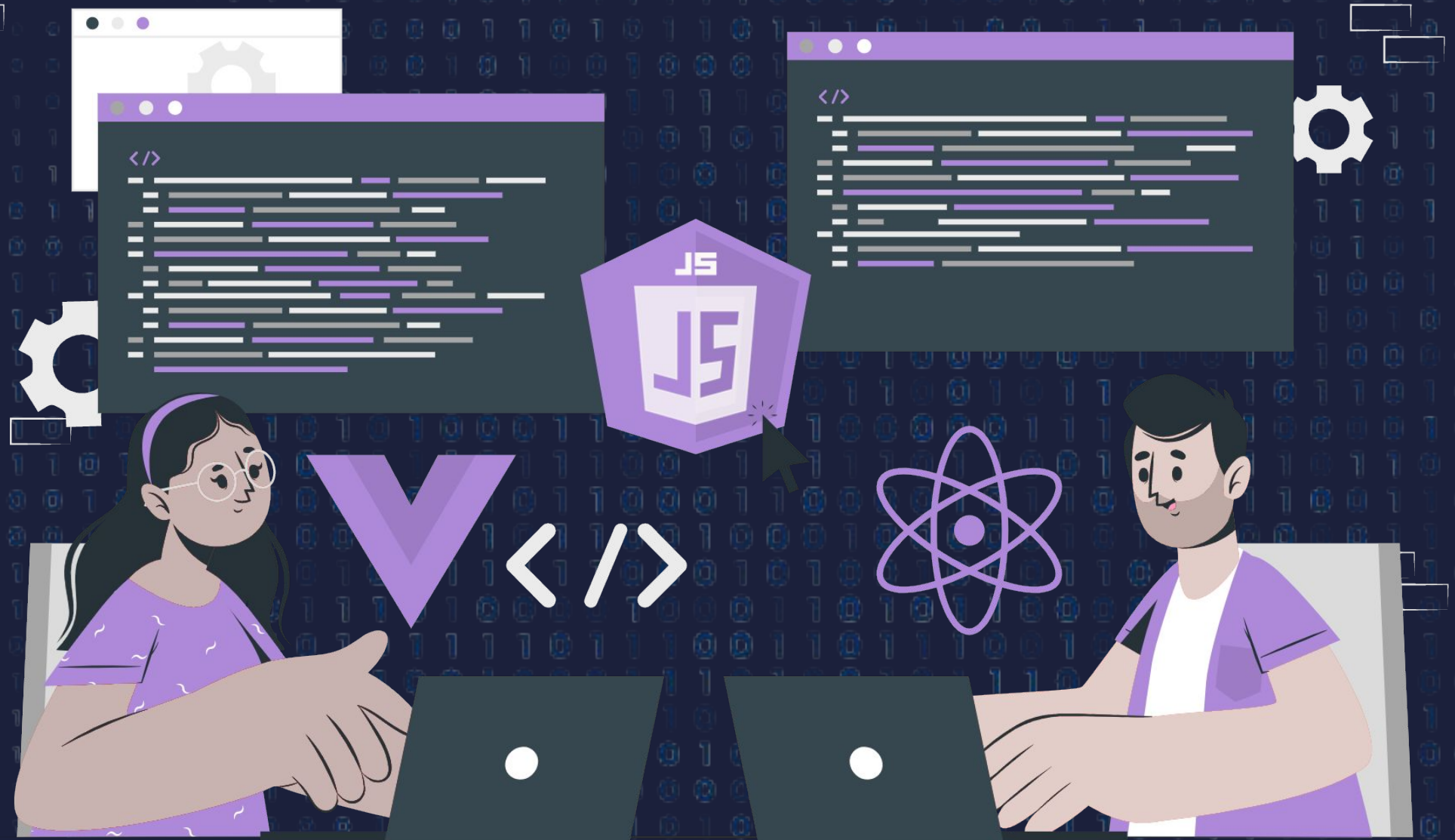




Function without a parameter and return



Lecture CheckList

1. Function without a parameter and return.
2. Syntax.
3. Implementation.
4. Code Walk Through.
5. Function returning value.
6. Syntax.
7. Implementation.
8. Code Walk Through

Function without a parameter and return.

In the previous lecture, we looked into some of the important terms such as parameters, arguments, and return statements.

One thing to note is that without parameters there won't be any existence of arguments.

The parameters and return statements inside a function are optional sometimes we wish to get some tasks done inside a function without taking any input parameters and without returning any output.

Syntax

```
// Function declaration
```

```
function functionName() {  
    // Function Body  
}
```

```
// Calling a function
```

```
functionName();
```


Implementation.

Let's look at an example to understand better.

Imagine you want to just print something onto the console and want to display the same message multiple times. Instead of writing the same statement multiple times we can write a function and call it as many times as we want.

By doing this the code repetition would be reduced and the code would be more neat and readable.

Implementation.

```

1  // Function Declaration
2
3  ✓ function sayMessage() {
4    |   console.log("I am happy to learn Full Stack Web Development From PW Skills");
5    | }
6
7  // Calling a Function
8
9  sayMessage();
10 sayMessage();
11 sayMessage();
12 sayMessage();
13
14 ✓ /*
15
16  OUTPUT:
17  I am happy to learn Full Stack Web Development From PW Skills
18  I am happy to learn Full Stack Web Development From PW Skills
19  I am happy to learn Full Stack Web Development From PW Skills
20  I am happy to learn Full Stack Web Development From PW Skills
21
22  */
23  |

```


Code Walk Through.

In the above example the function "sayMessage" is declared using the keyword "function". It does not take any parameters and has no return statement.

Inside the function, it uses `console.log` to display a message on the console.

The function is then called using the function name followed by parentheses, "sayMessage()".

The function is called four times, so the message "I am happy to learn Full Stack Web Development From PW Skills" will be displayed on the console four times.

Function returning value.

We know that the return statement in JavaScript is used to specify the value that a function should return when the function is called upon finishing the execution.

The return statement is followed by the value or expression that you want to return.

Functions can be declared without any parameters but can have a return statement. This is done to return some value when the function is called.

When the function is called and the return statement is executed, the function execution is stopped and the specified value is returned to the calling code.

Syntax

```
1  // Function returning value
2
3  // Declaration
4
5  function functionName() {
6      // Function Body
7      return expression / value;
8  }
9
10 // Calling a function
11
12 let return_value = functionName();
13
14 // The function would be executed and on executing the return statement the value / expression are stored in the return_value variable
15
```

Implementation

Let's look at an example where we want to return a message on calling the function instead of directly printing the message onto the console.

The message is returned using the return statement and the value is stored in the variable. The value stored can be used throughout the program.

Implementation

```
1  // Function returning value
2
3  // Declaration
4
5  function passMessage() {
6    | return "I am happy to learn Full Stack Web Development from PW Skills";
7  }
8
9  // Calling a function
10
11 let message = passMessage();
12
13 // The function would be executed and on executing the return statement the returned message is stored in the message variable
14
15 console.log(message);
16
```

Code Walk Through

The function "passMessage" is declared using the keyword "function" and it does not take any parameters. Inside the function, it uses the "return" keyword followed by a string that contains a message.

The function is then called using the function name followed by parentheses, "passMessage()", and the returned value is stored in the variable "message". The value stored in the "message" variable can be accessed and used later in the code, in this example it is logged to the console using `console.log(message)` which will display "I am happy to learn Full Stack Web Development from PW Skills" on the console.

Code Walk Through

```
1 // Function returning value
2
3 // Declaration
4
5 function passMessage() {
6   return "I am happy to learn Full Stack Web Development from PW Skills!!";
7 }
8
9 // Calling a function
10
11 // let message = passMessage();
12
13 // The function would be executed and on executing the return statement the returned message is stored in the message variable
14
15 console.log(message); // I am happy to learn Full Stack Web Development from PW Skills!!
16
17 let newMessage = message + " Enroll Today";
18
19 console.log(newMessage); // I am happy to learn Full Stack Web Development from PW Skills!! Enroll Today
20
```

Code Walk Through

Here the returned message is stored in the variable 'message' and the variable 'newMessage' uses the variable 'message' and adds it with the string "Enroll Today" and the final message is logged to the console using `console.log(newMessage)` which will display "I am happy to learn Full Stack Web Development from PW Skills!! Enroll Today" on the console.



▶ THANK YOU ◀