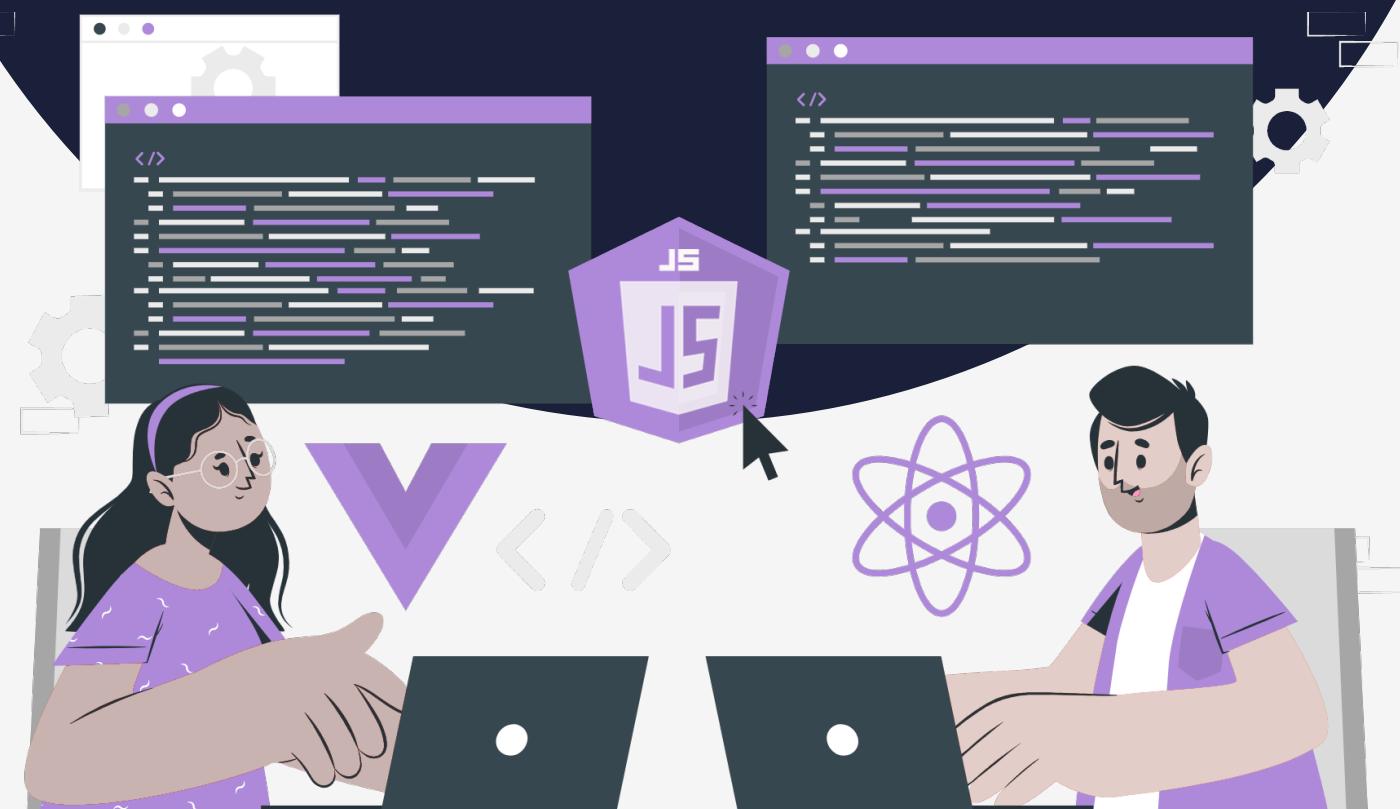


Lesson:

Function without a parameter and return



Topics Covered:

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

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 without a parameter and return

// Function declaration

function functionName() {
    // Function Body
}

// Calling a function

functionName();
```

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.

```
// Function Declaration

function sayMessage() {
  console.log("I am happy to learn Full Stack Web Development From PW Skills");
}

// Calling a Function

sayMessage();
sayMessage();
sayMessage();
sayMessage();

/*
OUTPUT:
I am happy to learn Full Stack Web Development From PW Skills
I am happy to learn Full Stack Web Development From PW Skills
I am happy to learn Full Stack Web Development From PW Skills
I am happy to learn Full Stack Web Development From PW Skills
*/
```

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.

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. Sometimes, the return statement can be used without a value, to return the control to the caller before the function has been completely executed.

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:

```
// Function returning value

// Declaration

function functionName() {
    // Function Body
    return expression/value;
}

// Calling a function

let return_value = functionName();

// The function would be executed and on executing the return statement the value/
expression are stored in the return_value variable
```

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.

```
// Function returning value

// Declaration

function passMessage() {
    return "I am happy to learn Full Stack Web Development from PW Skills";
}

// Calling a function

let message = passMessage();

// The function would be executed and on executing the return statement the returned
message is stored in the message variable

console.log(message);

// Function with no return value

function printName(name) {
    if (name == undefined) {
        return;
    }
    console.log(name);
}

printName(); // OUTPUT: NO OUTPUT
printName("PW Skills"); // OUTPUT: PW Skills
```

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.

The function `printName` takes in a parameter "name" and checks if it is undefined. If the condition is satisfied, the function ends without returning any value. If it is not undefined, the function logs the "name" to the console.

When the function is called with no argument passed in, it will not output anything as the condition inside the function is met. If the function is called with an argument passed in, such as "PW Skills", the function will log that name to the console.

We can further use the value stored throughout the program.

```
// Function returning value
// Declaration

function passMessage() {
    return "I am happy to learn Full Stack Web Development from PW Skills!!";
}

// Calling a function

let message = passMessage();

// The function would be executed and on executing the return statement the returned
message is stored in the message variable

console.log(message); // I am happy to learn Full Stack Web Development from PW
Skills!!

let newMessage = message + " Enroll Today";

console.log(newMessage); // I am happy to learn Full Stack Web Development from PW
Skills!! Enroll Today
```

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.