JavaScript Closures

Before you learn about closures, you need to understand two concepts:

- Nested Function
- Returning a function

JavaScript Nested Function

In JavaScript, a function can also contain another function. This is called a nested function. For example,

```
// nested function example

// outer function
function greet(name) {

    // inner function
    function displayName() {
        console.log('Hi' + ' ' + name);
    }

    // calling inner function
    displayName();
}

// calling outer function
greet('John'); // Hi John
```

Run Code

In the above program, the `greet()` function contains the `displayName()` function inside of it.

Returning a Function

In JavaScript, you can also return a function within a function. For example,

```
function greet(name) {
    function displayName() {
        console.log('Hi' + ' ' + name);
    }

    // returning a function
    return displayName;
}

const g1 = greet('John');
console.log(g1); // returns the function definition
g1(); // calling the function
```

Run Code

Output

```
function displayName() {
      console.log('Hi' + ' ' + name);
  }
Hi John
```

In the above program, the `greet()` function is returning the `displayName` function definition.

Here, the returned function definition is assigned to the `g1` variable. When you print `g1` using `console.log(g1)`, you will get the function definition.

To call the function stored in the `g1` variable, we use `g1()` with parentheses.

JavaScript Closures

In JavaScript, closure provides access to the outer scope of a function from inside the inner function, even after the outer function has closed. For example,

```
// javascript closure example
```

```
// outer function
function greet() {

    // variable defined outside the inner function
    let name = 'John';

    // inner function
    function displayName() {

        // accessing name variable
        return 'Hi' + ' ' + name;

    }

    return displayName;
}

const g1 = greet();
console.log(g1); // returns the function definition
console.log(g1()); // returns the value
```

Run Code

Output

```
function displayName() {
      // accessing name variable
      return 'Hi' + ' ' + name;
  }
Hi John
```

In the above example, when `greet()` function is called, it returns the function definition of `displayName`.