Types of functions

First-class function:

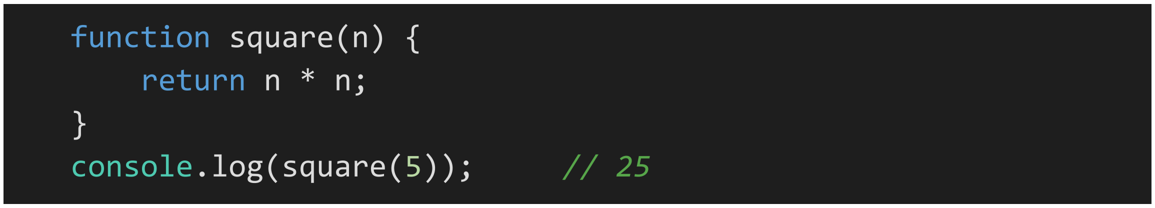
First-class functions mean when functions in that language are treated like any other variable. This means that these functions can be

- passed as arguments to other functions
- returned as values from other functions
- assigned to variables
- Stored as a variable in an array, objects

[Refer first_class_functions.js for examples]

First-order function:

A first-order function is a function that doesn't accept other functions as arguments and doesn't return a function as its result. They operate on fixed data types. *Example:*



```
function square(n) {
    return n * n;
}
console.log(square(5));      // 25
```

<u>Note</u>: In JavaScript, all functions are first class functions. However, not all functions are first-order functions. For example, the `setTimeout()` function is a first class function, but it is not a first-order function because it takes a function as an argument.

Higher order function:

A higher-order function is a function that can take one or more functions as arguments, and return a function as a result.
Examples: map(),filter() and reduce()

Pure function:

A pure function is a function where the return value is only determined by its arguments without any side effects. i.e, If you call a function with the same arguments then it will always return the same value.*Example:*

```
function add(a, b) {
   return a + b;
}
```

- Arrow function:

An arrow function is a shorter syntax for a function expression and does not have its own this, arguments, super, or new.target. These functions are best suited for non-method functions, and they cannot be used as constructors. *Example:*

```
    const greet = name => `Hello, ${name}!`;
    console.log(greet('Alice'));               // Hello, Alice!
```