Lecture 3: Functions

What is a Function

- A **function** is a block of code designed to perform a specific task.
- It runs only when it is **called** or **invoked**
- Helps with **code reuse** – you can define a function once and use it multiple times.

Function Declaration

A **function declaration** defines a named function that can be invoked later.

Function declarations are **hoisted,** meaning they can be called before they are defined in the code.

```
functionName(1);

function functionName(parameters) {
    // code to be executed
}
```

Function Expression

- A **function expression** involves creating a function and assigning it to a variable.
- Function expressions are **not hoisted,** meaning they cannot be called before they are defined

```
const someFunction = function() {
    // code to be executed
}
```

Function Arguments

Arguments are the values passed to a function when it is invoked.

These values correspond to the **parameters** defined In the function.

You can pass **any** data type as arguments.

```
printName('Busa');

function printName(name) {
    console.log(name);
}
```

Default Parameters

Default parameters allow you to specify default values for function parameters.

If no argument or **undefined** is passed when the function is invoked, the parameter takes the default value

```
printName();

function printName(name = 'Busa') {
    console.log(name);
}
```

Callback

A **callback** is a function passed as an argument to another function and executed after the completion of that function.

```
invokeCallback(function () {
    console.log('YAAAY')
});

function invokeCallback(cb) {
    cb();
}
```

return keyword

The **return** keyword is used to **exit a function** and optionally pass back a value to the function's caller.

When a **return** statement is executed, the function **stops** running, and control is handed back to the caller

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

let sumValue = sum(1, 2);

console.log(sumValue); // 3
```

Higher-order Function

A **higher-order** function is a function that **returns another function** or **takes a function as an argument.**

```
let f = function() {
    return function() {
        console.log('from callback');
    }
}

f()();
```

Pattern: IIFE

IIFE stands for **Immediately Invoked Function Expression.**

It is a function that is **defined and executed immediately** after its creation.

The syntax involves wrapping the function inside parenthesis and immediately invoking it with ()

```
(function () {
    console.log('Hello world!')
})();
```

arrow function

Arrow functions are a more concise way to write functions.

Key Differences

- No need for the **function** keyword
- If the function body has only a **single expression,** the **return** keyword and braces { } can be omitted.

```
const arrowFunc = (a, b) => {
    return a + b;
}

const newArrowFunc = (a, b) => a + b;
```