# JavaScript – Lecture 3: Functions, Scope, Regex, DOM

## Function Types

***1. Regular Function (Declaration):***

```
function greet(name) {
console.log("Welcome", name);
}
```

***2. Anonymous Function (Expression):***

```
const greet = function(name) {
console.log("Welcome", name);
};
```

***3. Arrow Function:***

```
const greet = name => console.log("Welcome", name);
```

***4. Callback Function:***

```
function doSomething(callback) {
callback();
}
```

***5. Self-Invoked Function:***

```
(function() {
console.log("I run myself");
})();
```

## Local Scope

Variables declared inside a function are only accessible within that function.

## Rest Operator (...)

Used to group multiple values into an array in function parameters.

Example:

function sum(...numbers) {

return numbers.reduce((acc, cur) => acc + cur, 0);

}

## Array Iteration Methods

***1. forEach():*** loops over array without return.

***2. map():*** returns new array after applying transformation.

***3. filter():*** returns elements matching condition.

***4. reduce():*** reduces array to a single value.

Example:

arr.map(item => item * 2);

## every() and some() Methods

***- every():*** checks if all items match a condition → returns true/false.

***- some():*** checks if at least one item matches a condition → returns true/false.

## Regular Expressions (Regex)

Used for pattern matching in strings.

Syntax: /pattern/flags

Common Patterns:

- \d : digit

- \w : word character

- \s : whitespace

- ^ : start of string

- $ : end of string

- [] : character class

- {min,max} : repetition

Example:

let pattern = /^\+20 1[0-2]{1}[0-9]{8}$/;

pattern.test("+20 1234567890");

## Intro to DOM

***DOM = Document Object Model: allows JS to interact with HTML elements.***

**Selection Methods:**

- document.getElementById("id")

- document.querySelector(".class" or "#id")

- document.querySelectorAll("tag")

- document.getElementsByTagName("tag")

- document.getElementsByClassName("class")

## Array Destructuring + Swapping

Destructuring allows unpacking values from arrays.

Example:

let [a, b] = [1, 2];

Swapping using destructuring:

[a, b] = [b, a];

## Object Destructuring

Allows extracting properties into variables.

Example:

let person = {name: 'Ali', age: 25};

let {name, age} = person;