

Hands-on JavaScript Functions



Purpose of the this hands-on training is to teach the students JavaScript Functions.

Learning Outcomes

At the end of the this hands-on training, students will be learned;

- What is Functions?
- Function Expression
- Calling/Invoking Function
- Function Scope
- Arrow Function

Part 1 - What is Functions?

- Functions are the main “building blocks” of the program. They allow the code to be called many times without repetition and designed to perform a particular task.

Syntax :

```
function nameOfFunction(param1, param2 ...) {  
    statement;  
}  
/* This is Function Declaration
```

Part 2 - Function Expression

- A JavaScript function can also be defined using an expression and a function expression can be stored in a variable.

Syntax :

```
var nameOfFunc = function (param1 , param2 ... ) {  
    statement  
};
```

- The function above is actually an **anonymous** function (a function without a name).

Part 3 - Calling/Invoking Function

- Defining a function does not execute it. Defining it simply names the function and specifies what to do when the function is called. We need to call the function that we define.

Syntax :

```
nameOfFunc(param1, param2 ....);
```

- Example without paramater

- Example with paramater

Part 4 - Function Scope

- Variables defined inside a function cannot be accessed from anywhere outside the function, because the variable is defined only in the scope of the function. However, a function can access all variables and functions defined inside the scope in which it is defined.

```
// The following variables are global variables.
var num1 = 24;
word1 = "Edward";

function sub() {
  // The following variables are local variables.
  // You can use global variables in this scope.
  let num2 = 8;
  word2 = "Benedict";
  return num1 - num2;
}
// You can not use local variables in this scope.
```

Part 5 - Arrow Function

- Arrow functions allows a short syntax for writing function expressions.
- You don't need the function keyword, the return keyword, and the curly brackets.

Syntax :

```
// ES5
var res = function (x, y) {
  return x * y;
};

// ES6
const result = (x, y) => x * y;
```