**JavaScript**

**Part-3**

# JavaScript Functions

* A JavaScript function is a block of code designed to perform a particular task.
* A JavaScript function is executed when "something" invokes it (calls it).

### **Example**

function myFunction(p1, p2) {
  return p1 \* p2;   // The function returns the product of p1 and p2
}

## **JavaScript Function Syntax**

* A JavaScript function is defined with the function keyword, followed by a **name**, followed by parentheses **()**.
* Function names can contain letters, digits, underscores, and dollar signs (same rules as variables).
* The parentheses may include parameter names separated by commas:
**(parameter1, parameter2, ...)**
* The code to be executed, by the function, is placed inside curly brackets: **{ }**

function name(parameter1, parameter2, parameter3) {
  // code to be executed
}

## **Function Invocation**

The code inside the function will execute when "something" **invokes** (calls) the function:

* When an event occurs (when a user clicks a button)
* When it is invoked (called) from JavaScript code
* Automatically (self invoked)

## **Function Return**

* When JavaScript reaches a return statement, the function will stop executing.
* If the function was invoked from a statement, JavaScript will "return" to execute the code after the invoking statement.
* Functions often compute a **return value**. The return value is "returned" back to the "caller":

### **Example**

Calculate the product of two numbers, and return the result:

var x = myFunction(4, 3);   // Function is called, return value will end up in x

function myFunction(a, b) {
  return a \* b;             // Function returns the product of a and b
}