



netlink
digital solutions



www.netlink.com



JavaScript

JavaScript : Function

- 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).
- **JavaScript functions** are used to perform operations. We can call JavaScript function many times to reuse the code.

JavaScript Function Syntax:

```
function functionName([arg1, arg2, ...argN]){  
  //code to be executed  
}
```

JavaScript : Function

- When a value is passed when declaring a function, it is called **parameter**. And when the function is called, the value passed is called **argument**.

```
// program to add two numbers using a function
// declaring a function
function add(a, b) {
    console.log(a + b);
}

// calling functions
add(3,4);
add(2,9);
```

- you can call a function as many times as you want. You can write one function and then call it multiple times with different arguments.

JavaScript : Declaration

- A function declaration also known as a function statement declares a function with a function keyword. The function declaration must have a function name.
- Function declaration does not require a variable assignment as they are standalone constructs.
- These are executed before any other code.
- The function in function declaration can be accessed before and after the function definition.
- The **function declaration** (function statement) defines a function with the specified parameters.

Syntax:

```
function getInfo(paramA, paramB)  
{ // Set of statements }
```

JavaScript : Expression

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

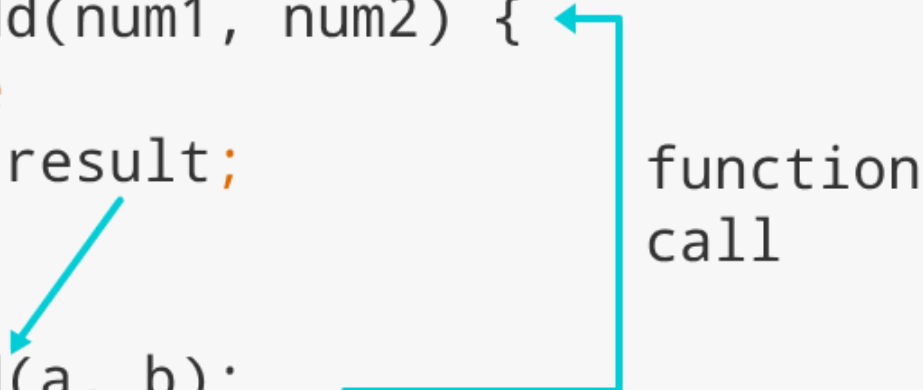
```
var x = function (a, b) {return a * b};
```

- Functions stored in variables do not need function names. They are always invoked (called) using the variable name.

JavaScript : Return Function

- The **return** statement can be used to return the value to a function call.
- The **return** statement denotes that the function has ended. Any code after **return** is not executed.
- When JavaScript reaches a **return** statement, the function will stop executing.

```
function add(num1, num2) {  
    // code  
    return result;  
}  
  
let x = add(a, b);  
// code
```



The diagram illustrates the execution flow of a function call. A teal arrow originates from the opening curly brace of the `function add` block and points to the `return result;` statement. Another teal arrow starts from the `add(a, b)` part of the `let x = add(a, b);` line and points back to the opening curly brace of the `function add` block. To the right of these arrows, the text "function call" is written, indicating the relationship between the function definition and its invocation.



Thank you

