

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  
}
```



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":



A lambda expression is a concise way to define a short function in programming. It's commonly found in modern languages like Python, Ruby, JavaScript, and Java.

Essentially, it's just a small piece of code that creates a function.

Thanks to functions being treated as objects in JavaScript, they can be passed to and returned from other functions, allowing the creation of lambda functions.

```
let multiply = (a, b) => a * b;  
console.log(multiply(5, 9));
```



