JavaScript III

functions

Functions

A JavaScript function is a block of code designed to perform a particular task.

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

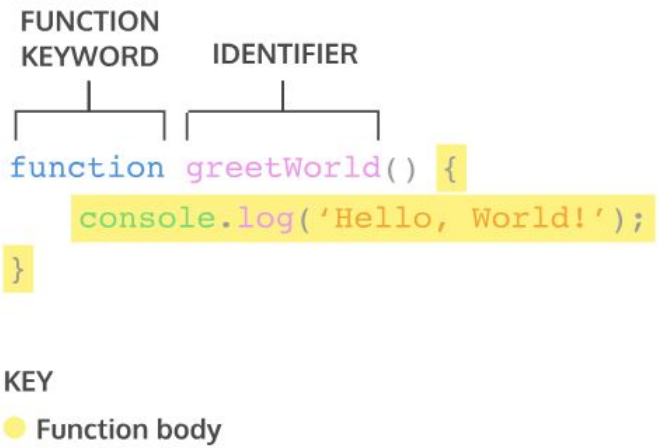
A JavaScript function is executed when "something" invokes it (calls it).

```
myFunction(4, 3);
```

Syntax

There a few ways to use and declare functions in JavaScript, including function expression, concise expression, arrow expression, etc.

This is the most basic and common function declaration:



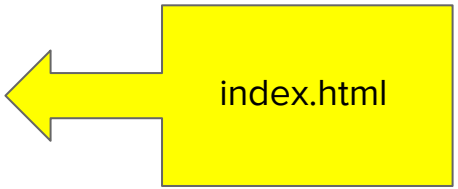
A function declaration consists of:

- The `function` keyword.
- The name of the function, or its identifier, followed by parentheses.
- A function body, or the block of statements required to perform a specific task, enclosed in the function's curly brackets, `{ }`.

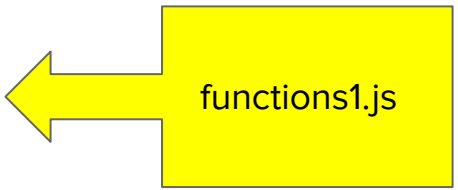
Basic Example

Let's write a Hello World function:

```
1   <!DOCTYPE html>
2   <html>
3   <head>
4       <title>JS Functions</title>
5   </head>
6
7   <body>
8
9   <div id="section1">
10  </div>
11
12  <script type="text/javascript" src="functions1.js"></script>
13
14  </body>
15  </html>
```




```
1   function HelloWorld(){
2       document.getElementById("section1").innerHTML = "Hello World!";
3   }
```



Calling a function

In order to execute the code in a function we have to call it.

We can call a function just by referring to it's name and adding an opening and closing parentheses + ending semi-colon:

Call your function

In your “functions1.js” call your Hello World function.
And preview the “index.html” file to see how it works.

Now, delete the HelloWorld(); function call from functions1.js
Now, let’s trigger the function call with a button click, Create a “functions1.html” file:

```
<!DOCTYPE html>
<html>
<head>
    <title>functions1</title>
</head>
<body>

<div id="section1">

</div>

<button onclick="HelloWorld();">submit</button>


<script type="text/javascript" src="functions1.js"></script>

</body>
</html>
```

* hoisting

The convention in most C-based languages is that functions have to be defined and declared before they can be called. JavaScript is an exception to this rule.

The hoisting feature in JavaScript which allows access to function declarations before they're defined.


```
1   HelloWorld();
2
3   function HelloWorld(){
4       document.getElementById("section1").innerHTML = "Hello World!";
5   }
```

Your turn

- In your JS file create another function called getReminder()
- This function should use the .innerhtml method to output the string ‘Water the Plants’ into a div with the id of “section2”. (You’ll need to create this div in your .html file.
- In your JS file create another function called greetInSpanish()
- This function should use the .innerhtml method to display an appropriate image into a div with the id of “section3”. (You’ll need to create this div in your .html file.
- In your html, create three more buttons to trigger each of these functions.

Functional utility

One of the reasons functions are powerful and important concepts in programming is because we can wrap up complicated or laborious code as a single function and call it as many times as we need to. Let's demo this in a functions2.js file:

```
1  function sayThanks(){
2    console.log('Thank you for your purchase! We appreciate your business.');
3  }
4  sayThanks();
5  sayThanks();
6  sayThanks();
```