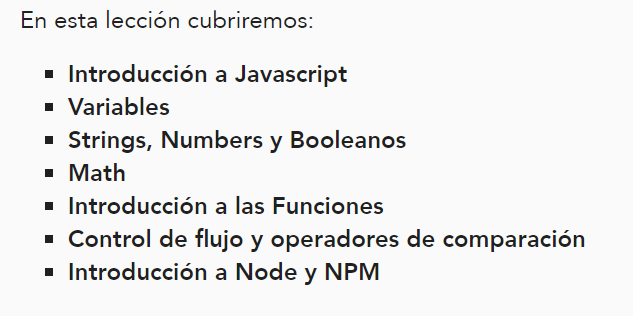
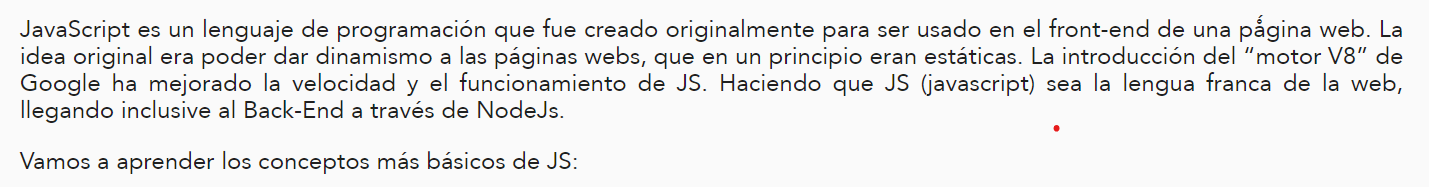
Introducción a JavaScript





**Variables**

Una variable es una forma de almacenar el valor de algo para usar más tarde (JavaScript es un lenguaje de tipado dinámico, una variable se puede configurar (y restablecer) a cualquier tipo, no necesitamos declarar el tipo al inicio de la variable)

Para crear una variable en JavaScript utilizamos la palabra reservada *var*, seguida de un espacio y el nombre de la variable

Tres tipos de variables:

**Var**

Es una palabra clave ES5

**Let**

Esta asignará una variable muy similar a var pero con un comportamiento un poco diferente

**Cons**

También es nuevo ES6, es una variable que no podrá cambiar su valor, es la abreviatura de constante.

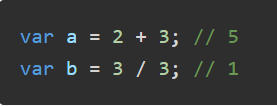
**Console.log();**

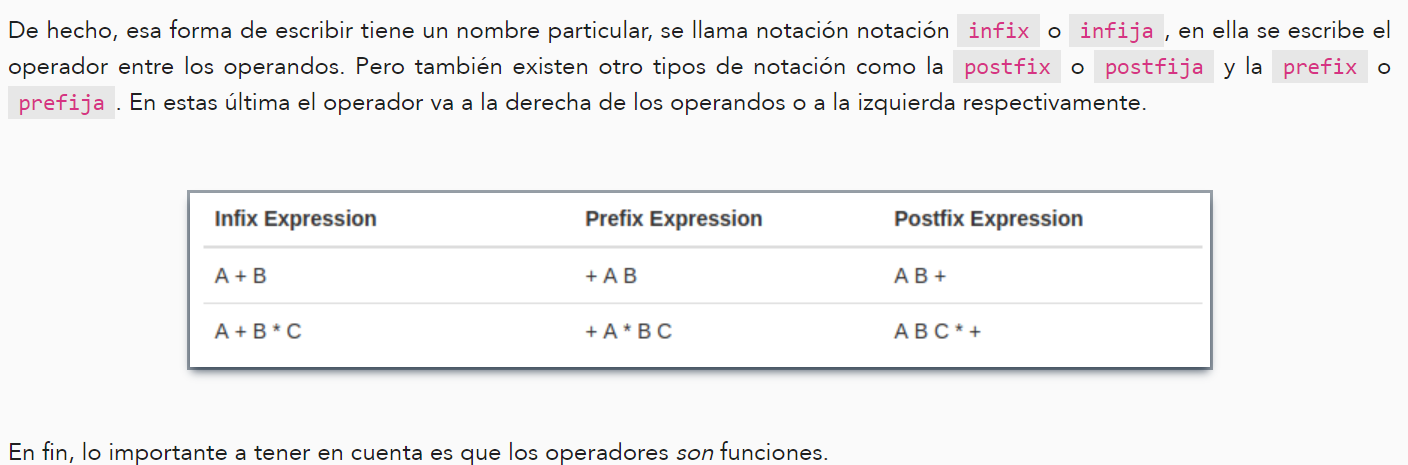
Es un método muy simple que nos ayudará a imprimir en la consola todo lo que pongamos entre paréntesis.

**Operadores**

Los operadores son los símbolos que ya conocemos (+, -, \*, /) que indican al interprete las operaciones que debe realizar.

Por ejemplo: Para el intérprete al ver el signo +, sabe que debe ejecutar la función de suma (que tiene internamente definida), y toma como parámetros los términos que estén a la derecha y a la izquierda del operador.





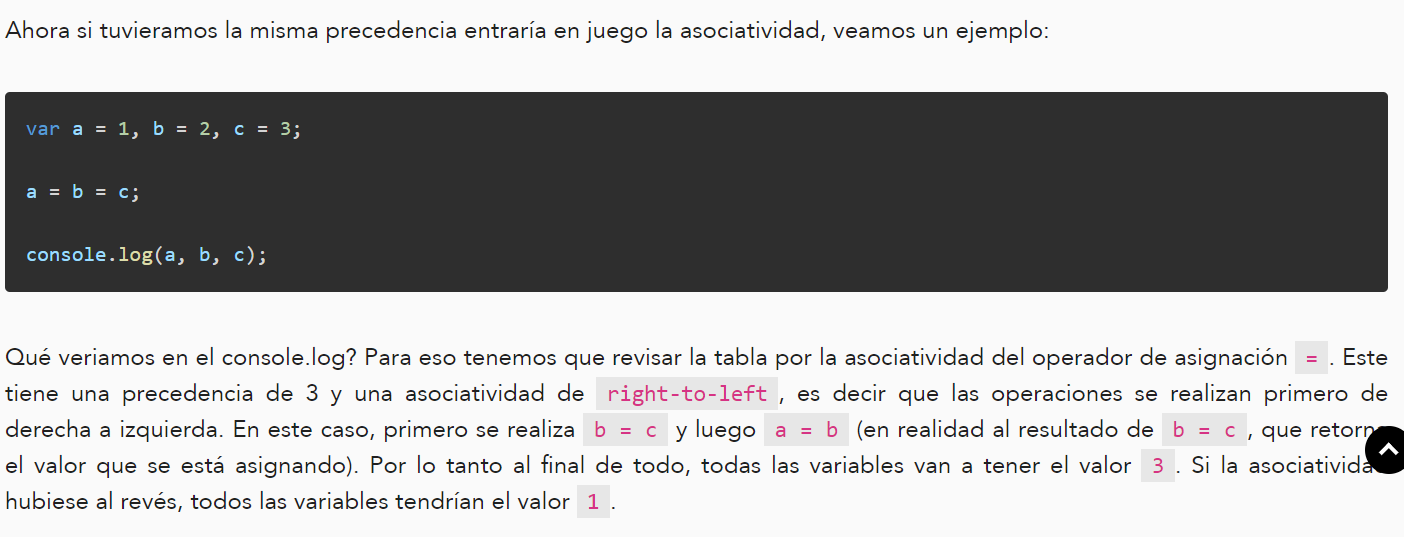
**Precedencia de los operadores y asociatividad**

La precedencia de operadores es básicamente el orden en que se van a llamar las funciones de los operadores. Estas funciones son llamadas en orden de precedencia (Las que tienen mayor precedencia se ejecutan primero). O sea que si tenemos más de un operador, el intérprete va a llamar al operador de mayor precedencia primero y después va a seguir con los demás.

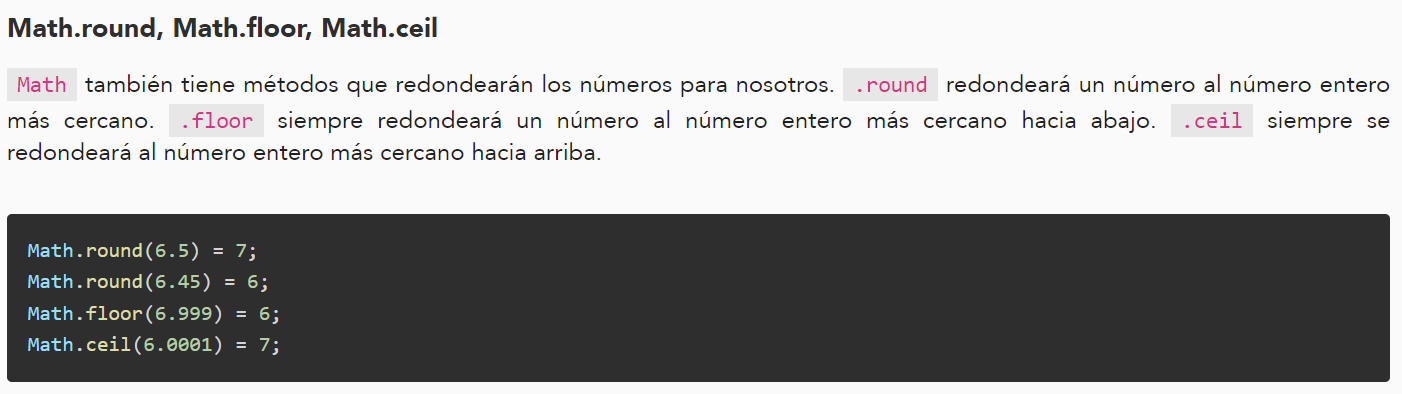
La asociatividad de los operadores es el orden en el que se ejecutan los operadores cuando tienen la misma precedencia, es decir de izquierda a derecha o de derecha a izquierda.

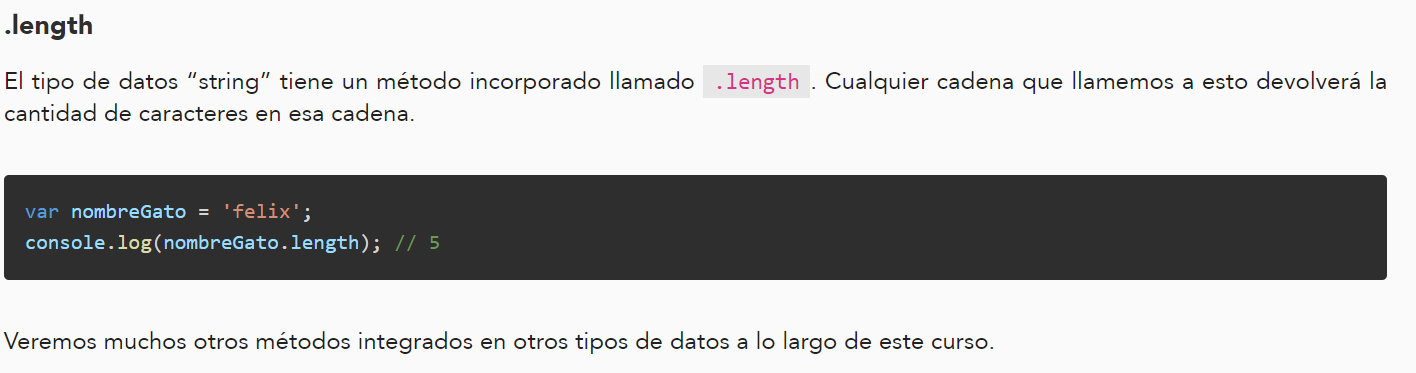
Podemos ver la documentación completa en el siguiente [link](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Operator_Precedence#table)

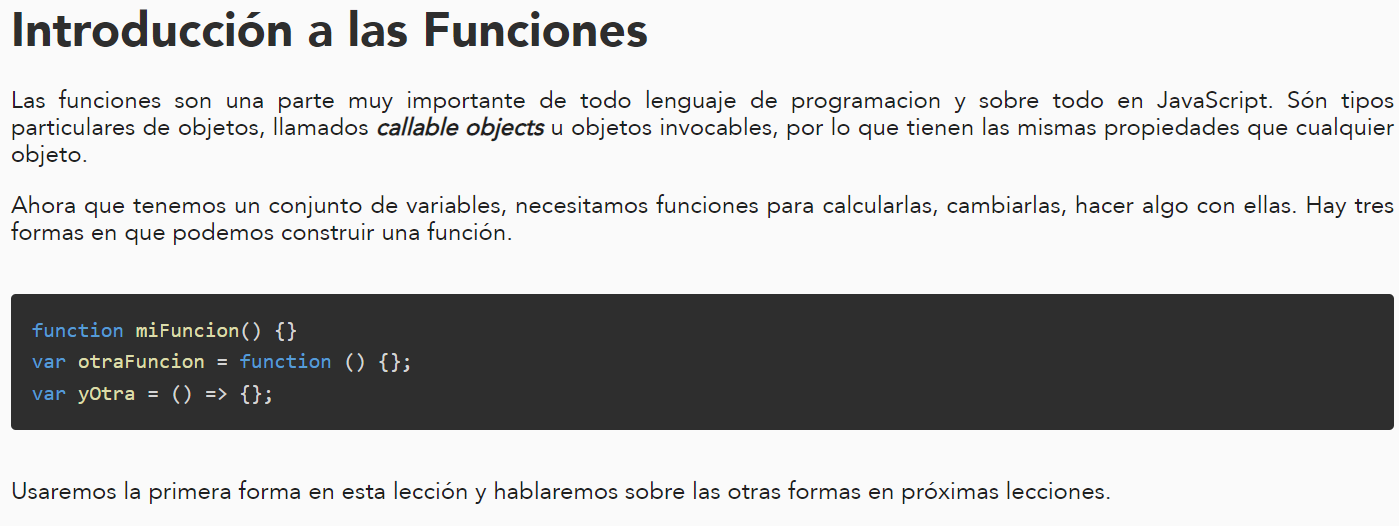
Por ejemplo para resolver las siguiente expresión *console.log(3+4\*5)* y saber qué resultado nos va a mostrar el intérprete deberíamos conocer el orden en el que se ejecutan las operaciones. Al ver la tabla del link de arriba, vemos que la multiplicación tiene una precedencia de 15, y la suma de 14. Por lo tanto el intérprete primero va a ejecutar la multiplicación y luego la suma con el resultado de lo anterior *-> console.log (3 + 20) -> console.log (23).*

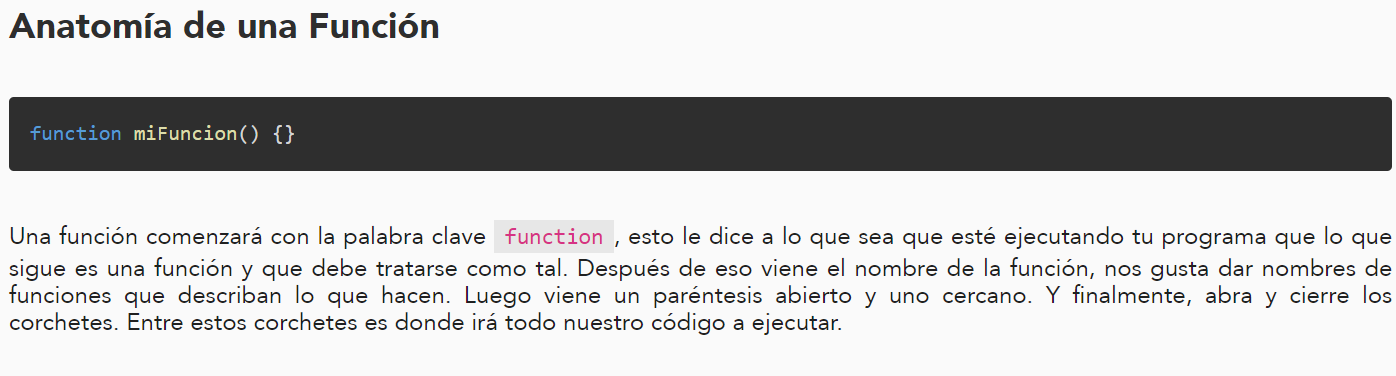


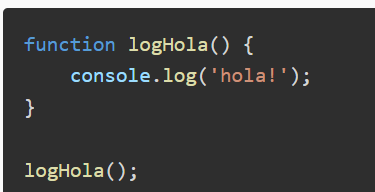


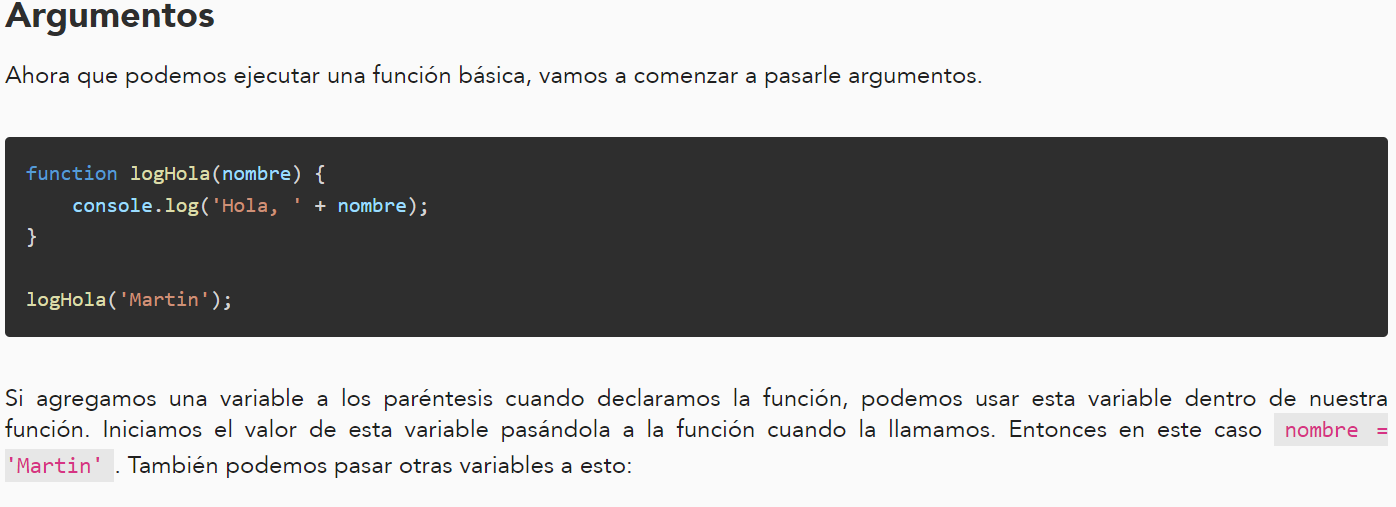














Las funciones pueden ser declarativas

Function miFunction(){

Return 3;

}

Funciones de expresión

Var miFuncion = function () {

Return a+b;

}