

El concepto de JavaScript High Order Functions , es uno de los más importantes que tenemos que conocer de la programación funcional. Sin embargo también en muchos casos es uno de los más difíciles de entender cuando lo abordamos. ¿Qué es y para que sirven las JavaScript High Order Functions?. Son funciones que reciben como parámetros funciones o devuelven funciones. Esto NO es algo sencillo de entender, pero sobre todo es algo difícil de ver su utilidad. Vamos a construir un ejemplo muy elemental que nos permita entenderlo. Supongamos que tenemos una serie de funciones que nos imprimen operaciones de suma ,resta y multiplicación por pantalla.

```
function sumar(a,b) {  
    console.log("*****")  
    console.log(a+b);  
    console.log("*****");  
}
```

```
function resta(a,b) {  
    console.log("*****")  
    console.log(a-b);  
    console.log("*****");  
}
```

```
function producto(a,b) {  
    console.log("*****")  
    console.log(a*b);  
    console.log("*****");  
}
```

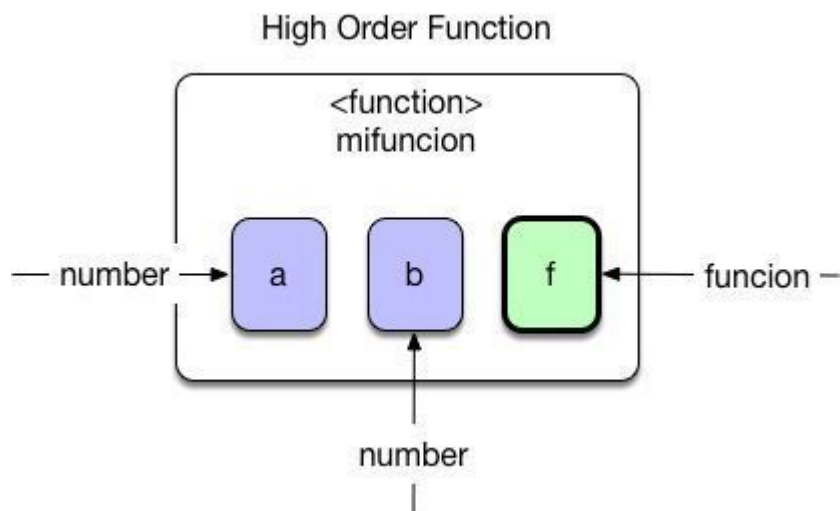
```
sumar(2,2);
```

```
resta(2,2);  
producto(2,2);
```

Como vemos son tres funciones muy similares que nos imprimen el resultado por la consola si las ejecutamos con node.js.

```
*****  
4  
*****  
*****  
0  
*****  
*****  
4  
*****
```

El código se ejecuta correctamente y cada función es completamente independiente. Ahora bien no nos gusta mucho ya que parece que hay mucho código repetido. ¿Como podemos solventar nuestros problemas?. Podemos crear un High order function y pasar una función como parámetro a otra función que será la encargada de imprimirlo.



Vamos con ello ,el primer paso es redeclarar las funciones de forma independiente:

```
var suma=function(a,b) {
```

```
    return a+b;
}
```

```
var restar=function(a,b) {
```

```
    return a-b;
}
```

```
var multiplicar=function(a,b) {
```

```
    return a*b;
}
```

JavaScript High Order Functions

Son nuestras funciones de siempre solamente que en este caso simplemente realizan la operación matemática y punto. Es momento de diseñar nuestra otra función que es la función de orden superior e importante.

```
function imprimir(a,b,operacion) {  
    console.log("*****");  
    console.log(operacion(a,b));  
    console.log("*****");  
  
}
```

Esta función tiene la peculiaridad de que recibe una función como parámetro y nos permite simplificar sobre manera el código anterior . Únicamente nos queda invocarla y que se ejecute:

```
imprimir(2,3,suma);  
imprimir(2,3,restar);  
imprimir(2,3,multiplicar);
```

El resultado es idéntico.



Acabamos de realizar nuestro primer ejemplo con JavaScript High Order Functions . Hemos eliminado por completo la repetición del uso de asteriscos (*).

Otros artículos relacionados :

1. [JavaScript Deferred Objects y peticiones asíncronas](#)
2. [JavaScript currying y funciones parciales.](#)
3. [El uso de JavaScript for in vs for of](#)
4. [JavaScript Functions](#)