

JAVASCRIPT FUNCIONAL



Jesús Javier Doménech Arellano

27 Enero 2016

Este lenguaje posee las siguientes características:

- Es dinamicamente tipado.
- Esta orientado a objetos (prototype).
- Funcional.
- De propósito general.

Este lenguaje posee las siguientes características:

- Es dinámicamente tipado.
- Esta orientado a objetos (prototype).
- Funcional.
- De propósito general.

Además, está presente en:

- Aplicaciones móviles.
- Sitios web.
- Servidores web.
- Aplicaciones de escritorio.
- Bases de datos.

λ expresiones:

```
1 (param1, param2, ...) => expresión;  
2  
3 (param1, param2, ...) => {  
4     predicado1;  
5     predicado2;  
6     ...  
7     return predicadofinal;  
8 }
```

EXPRESIONES

λ expresiones:

```
1 (param1, param2, ...) => expresión;  
2  
3 (param1, param2, ...) => {  
4     predicado1;  
5     predicado2;  
6     ...  
7     return predicadofinal;  
8 }
```

Funciones:

```
1 function (param1, param2, ...){  
2     instrucción1;  
3     instrucción2;  
4     ...  
5     return instrucciónfinal;  
6 }  
7  
8 function Nombre( ... ){ ... }
```

EJEMPLO

```
1   var multiplicar = (a,b) => a * b;  
2  
3   var apply = (opt, a, b) => opt(a,b);  
4   apply(multiplicar, 4, 8); // -> 32
```

```
1   function sumar(a) {  
2       return function (b) {  
3           return a + b;  
4       }  
5   }  
6  
7   sumar(3)(4); //-> 7
```

Erlang:

```
1 condición1 ->
2             instrucciones1;
3 condición2 ->
4             instrucciones2;
5 ...
6 condiciónN ->
7             instruccionesN;
8 true       ->
9             instruccionesDefault;
```

CONDICIONALES

Erlang:

```
1 condición1 ->  
2           instrucciones1;  
3 condición2 ->  
4           instrucciones2;  
5 ...  
6 condiciónN ->  
7           instruccionesN;  
8 true      ->  
9           instruccionesDefault;
```

Operador Ternario js:

```
1 condición ? instruccionesCasoCierto : instruccionesCasoFalso;
```


CONDICIONALES

Erlang:

```
1 condición1 ->
2             instrucciones1;
3 condición2 ->
4             instrucciones2;
5 ...
6 condiciónN ->
7             instruccionesN;
8 true        ->
9             instruccionesDefault;
```

Operador Ternario js:

```
1 condición ? instruccionesCasoCierto : instruccionesCasoFalso;

1 condición1 ? instrucciones1 :
2 condición2 ? instrucciones2 :
3 ...
4 condiciónN ? instruccionesN :
5             instruccionesDefault;
```

UnderScore y Lodash son dos librerías que definen un entorno contenido en la variable “_”. Nos proporciona métodos para realizar bucles.

- *forEach*: recibe una lista y una función, se ejecuta la función una vez por cada elemento de la lista.
- *map*: recibe una lista y una función, devuelve el resultado de ejecutar la función a cada elemento de la lista.
- *filter*: recibe una lista y una función, devuelve los valores de la lista que dan cierto en la función pasada.
- *reduce*: recibe una lista, una función y un elemento base. La función recibe un acumulador y un elemento de la lista. Devuelve el valor final de la acumulación.
- *curry*: recibe una función con N argumentos y la currifica.

- *Wrappers*
- *Funtores*
- *IO Monads*
- *Acceso al DOM*

EJEMPLO

1. <https://dzone.com/storage/assets/379008-rc217-functionalprogramming.pdf>