

Jesús Javier Doménech Arellano

27 Enero 2016

JAVASCRIPT

Este lenguaje posee las siguientes características:

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

JAVASCRIPT

Este lenguaje posee las siguientes características:

- Es dinamicamente 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.

EXPRESIONES

λ expresiones:

EXPRESIONES

λ expresiones:

Funciones:

```
function (param1, param2, ...){
  instrucción1;
  instrucción2;
  ...
  return instrucciónfinal;
}
function Nombre( ... ) { ... }
```

EJEMPLO

```
var multiplicar = (a,b) => a * b;

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

```
function sumar(a) {
    return function (b) {
        return a + b;
}

sumar(3)(4); //-> 7
```

CONDICIONALES

Erlang:

```
condición1 ->
instrucciones1;
condición2 ->
instrucciones2;
...
condiciónN ->
instruccionesN;
true ->
instruccionesDefault;
```

CONDICIONALES

Erlang:

```
condición1 ->
instrucciones1;
condición2 ->
instrucciones2;
...
condiciónN ->
instruccionesN;
true ->
instruccionesDefault;
```

Operador Ternario js:

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

CONDICIONALES

Erlang:

```
condición1 ->
instrucciones1;
condición2 ->
instrucciones2;
...
condiciónN ->
instruccionesN;
true ->
instruccionesDefault;
```

Operador Ternario js:

```
condición ? instruccionesCasoCierto : instruccionesCasoFalso;

condición1 ? instrucciones1 :
condición2 ? instrucciones2 :
...
condiciónN ? instruccionesN :
instruccionesDefault;
```

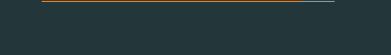
UnderScore

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.

SÀMs?

- Wrappers
- Funtores
- IO Monads
- Acceso al DOM



EJEMPLO

BIBLIOGRAFÍA

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