

# Javascript – algunos comentarios

Carlos Lombardi

Universidad Nacional de Quilmes – Argentina

Sesiones sobre microservicios

# Javascript – cómo surge

## Dinamismo a páginas Web

- ▶ Corre en browsers, una página puede incluir **scripts**.
- ▶ **DOM**: modelo de la página.
- ▶ Un script **JS** puede acceder y modificar el DOM “en vivo”.
- ▶ La página cambia **sin necesidad de “ir” al server**.

```
document.getElementById("demo").innerHTML = "Todo cambia";  
document.getElementById("demo").style.fontSize = "25px";
```

# Javascript – primeros elementos

Funciones y variables.

```
function triple(n) { return n * 3 }  
let a = triple(9)           // this is a comment
```

“No tipado”

No hace chequeo de tipos estático (antes de ejecutar).

Memoria dinámica

Corre sobre máquina virtual.

# Funciones – son ciudadanos de primer orden

Se pueden pasar por parámetro, se pueden asignar a variables.

```
function applyToInc(fn, n) { return fn(n+1) }  
let tripleFn = function(n) { return n * 3 }  
applyToInc(niceFn, 4)
```

Arrow function

**Nuevo** desde ES6 = JS 2015

```
applyToInc((n) => n * 2, 9)
```

# Object literal notation

Para registros ...

```
let a = {  
  name: 'Carlos',  
  age: 48,  
  country: 'Argentina'  
}  
a.name
```

... y para objetos

```
let a = { ...  
  halfTheAge: function() { return this.age / 2 }  
... }  
a.halfTheAge()
```

## Arrays y strings – son objetos

Se usan accediendo a **propiedades** y **métodos**.

También se puede acceder a un elemento: `arr[0]`.

```
let arr = [2,4,6,8,10]
arr[0]
arr.length
arr.indexOf(8)

"hola".length
"hola".toUpperCase()
```

## Falsy y truthy

Todos los valores se consideran, o bien false, o bien true.

P.ej., 0, '', null se consideran false, son **falsy values**.

Por otro lado, 20, 'hola', {a:0, b:2} son **truthy values**.

Esto permite jugar con expresiones usando and y or.

```
0 || 500           // esto es 500
20 || 500          // esto es 20

0 && 500           // esto es 0
20 && 500          // esto es 500
```

# Undefined

## ¡¡ OJO !!

Las variables o propiedades no definidas **no** generan error.

Valor especial `undefined` .

Es *falsy*.

```
let a = {  
  name: 'Carlos',  
  age: 48,  
  country: 'Argentina',  
}  
a.city           // esto es undefined, no da error  
a.city || 'Posadas'  // esto es 'Posadas'
```



# Libraries

Hay **muchísimas**, por lo general son fáciles de usar.

Busquen, lean, aprendan.

Una muy popular: `lodash`

```
var _ = require('lodash');  
var rainRecords = [  
  { city: 'BsAs', year: 1901, rain: 984 },  
  { city: 'BsAs', year: 1902, rain: 846 },  
  { city: 'Mendoza', year: 1901, rain: 238 }  
];  
_.find(rainRecords, { city: 'BsAs', year: 1901 });
```

# Javascript – conclusión

## Surge para

Dar dinamismo y poner chiches en páginas Web.

## Ahora

Es un lenguaje de programación de propósito general. Es un señor lenguaje de programación de propósito general.

## Por lo tanto

Salió del browser a conquistar el mundo.