Auxiliar 2

Frontend 2: Electric bogaloo

START

Javascript

Lenguaje de programación que se utiliza (principalmente) del lado del cliente

Permite mejoras de interfaz de usuario y dinamismo en sus páginas web :)

Vamos a usar Javascript con cositas nuevas (Desde el estándar **ES6** en adelante)

Correr Javascript

Todos los navegadores soportan correr Javascript.

Hay funciones que son **experimentales** o que no son **compatibles** con algunos navegadores.

También pueden correr código desde la consola de sus computadores usando **node** o **deno**

Se busca alguien que sepa JS!

Requisitos

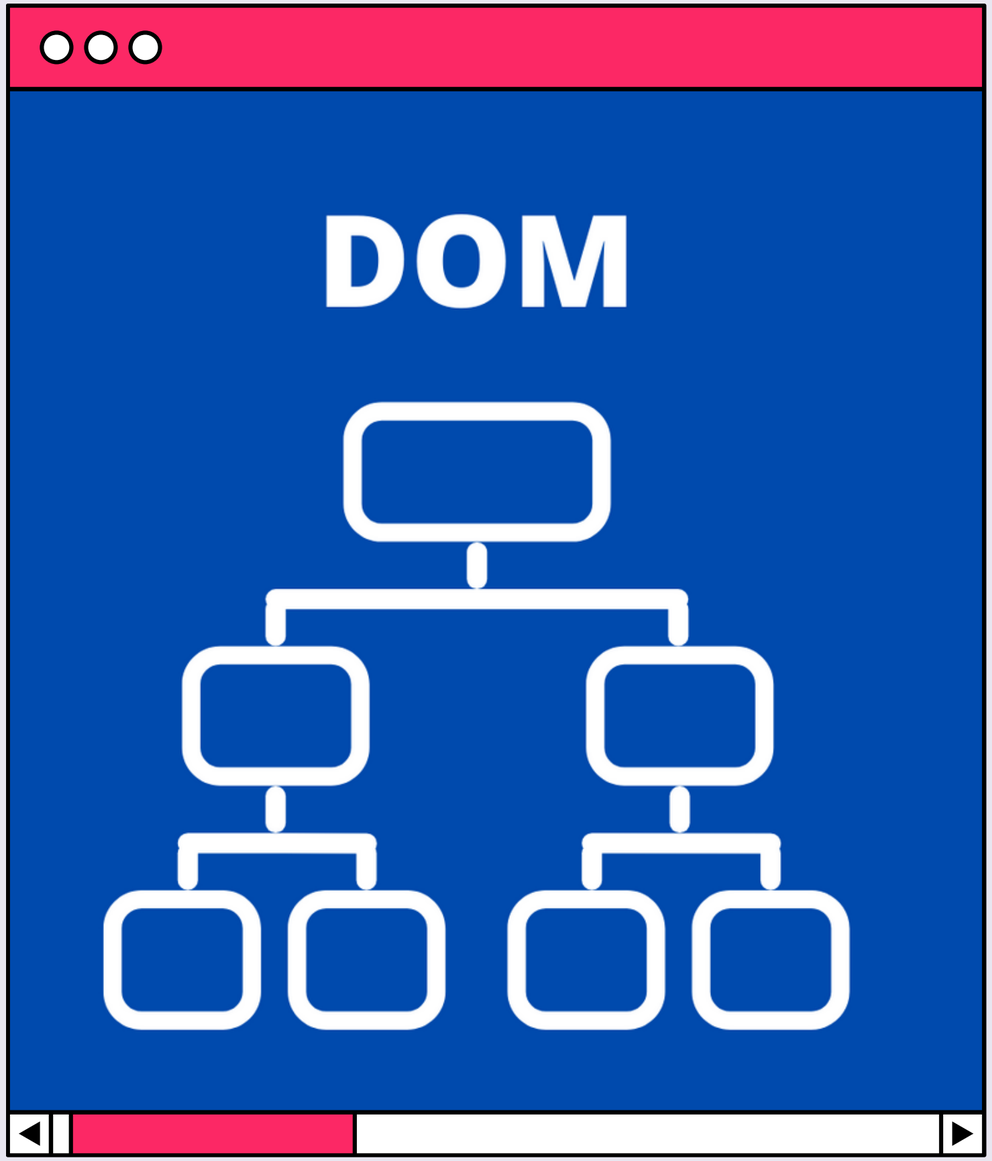
- No hay requisitos
- Solo quiero que mi página funcione
- Por favor
- Ayuda

DOM

Javascript puede interactuar con su HTML

Tiene acceso a un objeto llamado **document** que representa todo el documento HTML

Pueden referenciar elementos HTML usando **id** o **clases** para modificarlos



Variables

Const: Declara una variable constante. No se puede cambiar

```
> const PI = 3.145
> PI = 3 ❌
> PI
3.145
```

```
> const CONSTANTES = ['A', 'B']
> CONSTANTES.push('C')
> CONSTANTES = {'hola': 'wenas'} ❌
```

Variables

let: Declara una variable que si puede ir cambiando

```
> let val = 13
> val
13
> val = 'hola'
> val
hola
```

Funciones

$f(x)$

Nombre

Inputs

```
function f(x, y, z) {
  let a = x + y + z
  return a
}
```

Cuerpo

Pre ES6

Nombre

Inputs

```
const f = (x, y, z) =>
{
  let a = x + y + z
  return a
}
```

Cuerpo

Post ES6

Funciones

Las funciones también pueden ser inputs. Se usan harto en funciones de Javascript.

```
> const funcion = (a, b) => a + b
```

=

```
> const funcion = (a, b) => {
 return a + b
}
```

```
> const funcion = (a, b) => a + b
> let a = [1, 2, 3]
> a.reduce(funcion, 0)
6
```

=

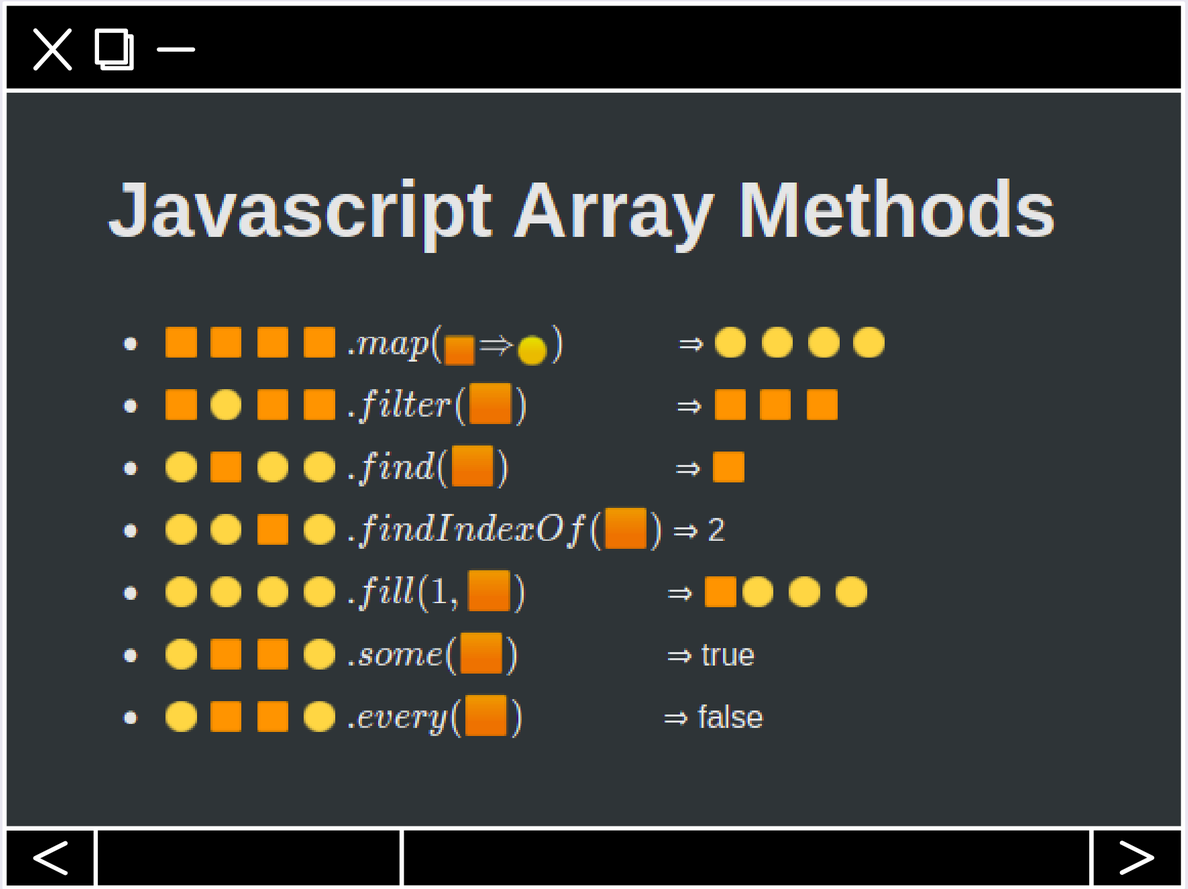
```
> let a = [1, 2, 3]
> a.reduce((x, y) => x + y, 0)
6
```

Objetos

- Almacenan una colección de valores
- Se declaran usando llaves (**{}**)
- Tienen una **llave** y un **valor**
- Tienen muchas **funciones asociadas**

```
> const o = {
 foo: 1,
 baz: (x, y) => x * y
}
> o['foo']
1
>o.foo
1
```

```
> let r = o.baz(2, 3)
> r
6
> o.boquitaLoMasGrande = true
> console.log(o)
Object { foo: 1, baz: function
boquitaLoMasGrande: true}
```



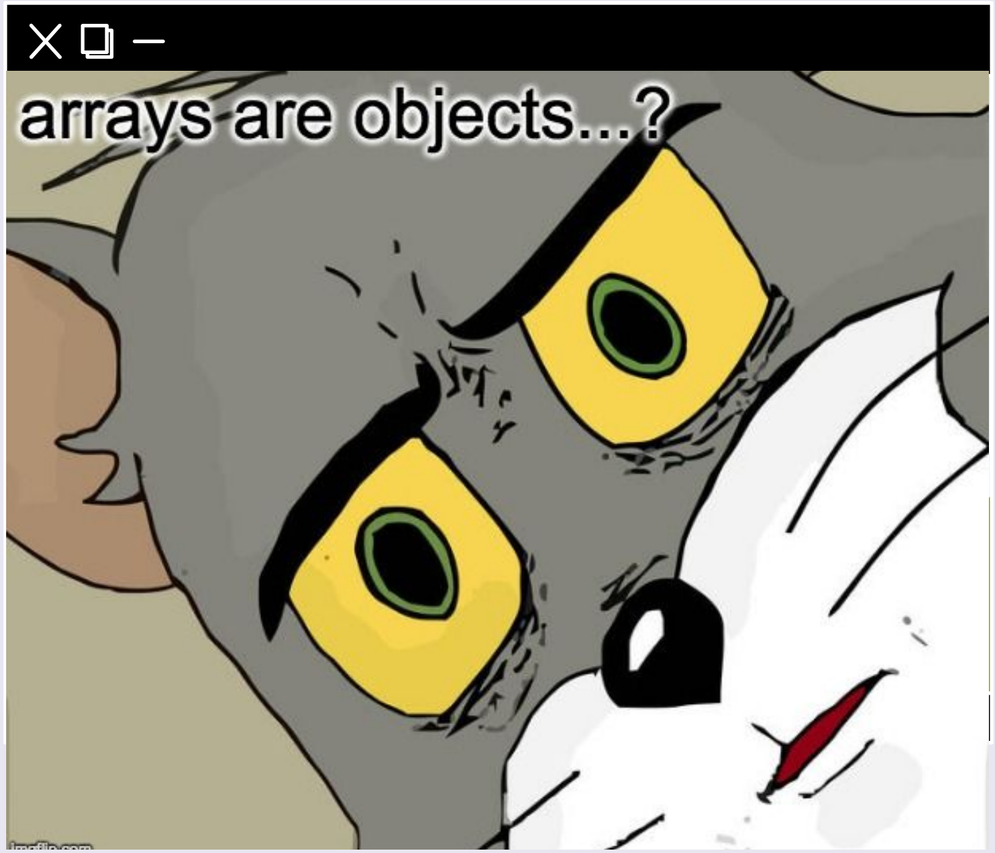
Arreglos

Se declaran usando corchetes (**[]**)

Pueden almacenar distintos tipos

Muchas **funciones asociadas**

```
> const a = [1, 'hola', [3, 4]]
> a[1]
hola
> a[0] = 2
> a
[2, 'hola', [3, 4]]
```



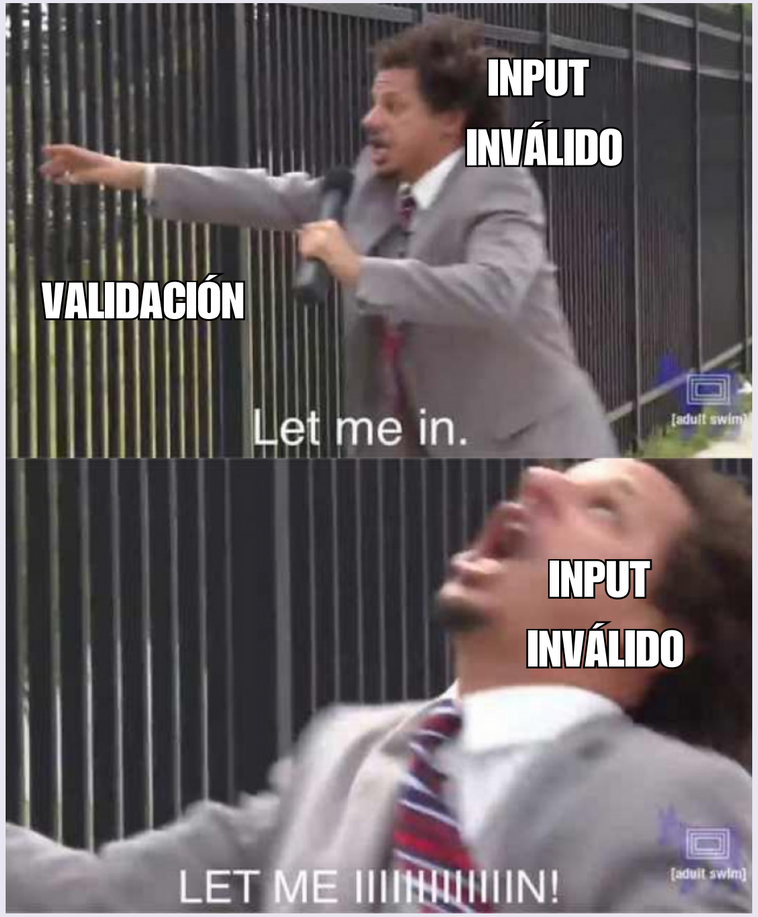
Arreglos

push: agregar un elemento al final

forEach: le pasas una funcion y usa esa función con cada elemento del arreglo

```
> const a = [1, 'hola', [3, 4]]
> a.push('x')
[1, 'hola', [3, 4], 'x']
> a.forEach((elem) =>
console.log(elem))
1 hola [3, 4] x
```

Validación

Siempre es bueno validar los inputs de formularios usando Javascript

1. Tomar los inputs **let elem = document.getElementById()**
2. Tomar su valor **let val = elem.value**
3. Validar que el valor que obtuvieron sea correcto
   a. Para esto les recomiendo crear funciones ya que muchas validaciones van a ser iguales (reutilicen código)
   b. Puede que necesiten validar con **expresiones regulares**
4. Si está bueno todo OK 😊
5. Si está mal sacrifiquen a su usuario 😈 (o solo avísenle que esta mal cualquiera sirve)