Curso Intermedio de Programación Orientada a Objetos en JavaScript

JuanDC



<u>Requisitos</u>

- Closures y scope 🔁
- Fundamentos de POO 🎾
- POO Básico en JS 🆠





www.platzi.com/web



¿Qué hay dentro de los objetos en JavaScript?





Static: atributos y métodos estáticos en JavaScript



```
class Patito {
    static sonidito = "cuak!";
console.log(Patito.sonidito);
// cuak!
```

```
class Patito {
    static hacerSonidito() {
      return "cuak!";
console.log(Patito.hacerSonidito());
// cuak!
```

```
const objetito = {
  name: "Juanito",
  email: "juanito@frijolitos.io",
  age: 18,
};
Object.keys(objetito);
// ["name", "email" "age"]
```

```
const objetito = {
  name: "Juanito",
  email: "juanito@frijolitos.io",
  age: 18,
};
Object.getOwnPropertyNames(objetito);
// ["name", "email" "age"]
```

```
const objetito = \{ /* ... */ \};
Object.entries(objetito);
/* [
    0: ["name", "Juanito"]
    1: ["mail", "juanito@frijolitos"]
    2: ["age", 18]
] */
```

Object

.getOwnPropertyDescriptors(objetito);

```
/*
name: {
    value: "Juanito",
    writable: true,
    configurable: true,
    enumerable: true,
```



Métodos estáticos del prototipo Object





Object.defineProperty



let patito → POINTER → { ... }

let patito2



Cómo funciona la memoria en JavaScript



let name = "Juan"

let age = 18

Nombres Valores

let name → "Juan"

let age → 18

STACK

Nombres Valores

let name → "Juan"

let age → 18

STACK

Nombres Valores Objetos

let name → "Juan"

let age → 18

let patito → POINTER → {...}

STACK

HEAP

Nombres Valores

Objetos

let name → "Juan"

let age → 18

let patito → POINTER → { ... }

let patito
POINTER
{ ... }

let patito2



Shallow copy en JavaScript





JSON.parse y JSON.stringify



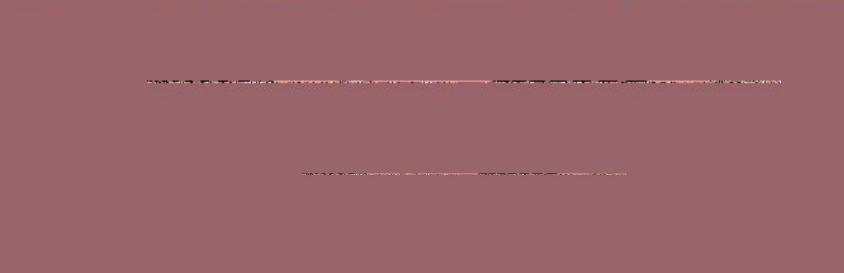


Qué es recursividad



```
let numerito = 0;
while(numerito < 5) {</pre>
  console.log(numerito);
  numerito++;
```

```
function recursiva(numerito) {
  console.log(numerito);
  if (numerito < 5) {</pre>
    return recursiva(numerito + 1);
  } else {
    return 5;
```





Deep copy con recursividad





Abstracción y encapsulamiento sin prototipos





Factory pattern y RORO





Module pattern y namespaces





Atributos y métodos privados en JavaScript





Getters y setters





Duck typing





Duck typing



































Duck typing en JavaScript





Instance Of en JavaScript





Atributos y métodos privados en prototipos





Creando métodos estáticos en JavaScript





¿Quieres más cursos de POO en JavaScript?

