



Argentina  
programa  
4.0

**.UBA**fiuba   
FACULTAD DE INGENIERÍA

# **JavaScript: Clase 5 – Objetos**

**Bianchi – Zulaica – Gonzalez**

Febrero 2023

# Contenido

---

- Qué son los objetos
- Cómo se declaran y cómo se crean
- Atributos
- Declaración de métodos
- Usar funciones para crear objetos
- Desarrollo de un ejemplo simple

# Qué son los objetos

Los objetos son **colecciones** de datos o propiedades relacionadas, que pueden tener un **comportamiento** específico

- Llamamos **atributos** a los datos de un objeto
- Llamamos **métodos** a los atributos que son funciones, y a estas funcionalidades ligadas al objeto las llamamos **comportamiento**
- Los utilizamos para representar elementos complejos, como por ejemplo personas (que pueden tener nombre, apellido, edad, etc.)

# Cómo se declaran y cómo se crean

**Sintaxis** → `let objeto = {  
 atributo: valor,  
 atributo: valor,  
 ...;  
}`

**Ejemplo** → `let persona = {  
 nombre: "Juan",  
 edad: 20,  
 dni: 40400222,  
}`

- Es un tipo de variable, declaramos pares clave-valor dentro de **llaves** ( { } )  
Similar a CSS salvo que separados por **comas** ( , )

# Atributos

- Los atributos pueden ser de **cualquier tipo**, incluso otros objetos
- Para acceder a un atributo utilizamos el **punto ( . )**
- Al acceder a un atributo funciona **como una variable**, podemos usar o modificar su valor

## Ejemplo

```
let persona = { nombre: "Juan", edad: 20, dni: 40400222 };
```

```
console.log( persona.nombre ); ➡ Muestra Juan
```

```
persona.edad += 1; ➡ Aumenta la edad en 1
```

```
persona.apellido = "Perez"; ➡ Podemos crear un atributo que antes no existía
```

```
console.log( persona ); ➡ Muestra todos sus datos
```

# Declaracion de metodos

- Al asignar una función a un atributo, la llamaremos **método**
- Podemos usar una sintaxis similar a **funciones sin la palabra function**
- Utilizamos la palabra **this** para acceder al objeto dentro del método
- Recordemos: Hay que invocar la función para que se ejecute

## Ejemplo

```
let persona = {  
  edad: 20, es_mayor(){ return this.edad >= 18 }  
}  
console.log( persona.es_mayor() ) ➡ Muestra True
```

# Usar funciones para crear objetos

- Si tenemos lógica repetida, por ejemplo objetos con la misma estructura y métodos, podemos hacer una función que la envuelva

## Ejemplo

```
function crear_persona (nombre, edad) {  
  return {  
    nombre: nombre,  
    Edad, → Es una sintaxis abreviada equivalente a “edad:edad”  
    es_mayor(){ return this.edad >= 18 }  
  }  
}
```



# Desarrollo de un ejemplo simple

Escribí este  
ejemplo en un  
archivo .html  
y probalo

```
<script>
  let persona = {
    nombre: "Juan",
    apellido: "Perez",
    saludar() {
      document.write(`Hola, soy ${this.nombre} ${this.apellido}`);
    },
  };

  persona.saludar();
</script>
```

# Resumen

---

- ✓ Un **objeto** está compuesto por
  - **Atributos:** que definen sus propiedades
  - **Métodos:** que definen su comportamiento
- ✓ Declaramos los objetos como pares **clave-valor**, separados por **comas (,)** que a su vez encerramos con llaves ( **{}** )

`{ atributo:valor, atributo:valor, ... }`

# Ejercicios

---

- 1) Declarar un objeto `Rectangulo` que tenga como atributos ancho y largo. Además que tenga dos métodos: `perimetro()` y `area()` que calculen y devuelvan las magnitudes correspondientes.
- 2) Declarar un objeto `CuentaDeBanco` que tenga como atributos número de cuenta y balance. Implementar los siguientes métodos:
  - `depositar(cantidad)`: Agrega `cantidad` al balance de la cuenta
  - `extraer(cantidad)`: Saca `cantidad` del balance de la cuenta
  - `transferir(cantidad, cuenta)`: Transfiere `cantidad` a la cuenta recibida.

**[www.ingenieria.uba.ar](http://www.ingenieria.uba.ar)**

**f**    /ingenieriauba

 /FIUBAoficial