

## Practica 02-05 Operadores de tipos avanzados y Objetos en JS

Nombre		Curso	
Apellidos		Fecha	

### Operador . (punto)

Se usa para **acceder a propiedades o métodos de un objeto**.

```
let persona = {  
  nombre: "Lucía",  
  edad: 28,  
  saludar: function() {  
    console.log("Hola");  
  }  
};
```

```
console.log(persona.nombre); // "Lucía"  
console.log(persona.edad);   // 28  
persona.saludar();           // "Hola"
```

- Solo se puede usar si conoces el **nombre exacto** de la propiedad.
- No sirve para claves dinámicas (para eso se usa [ ]).

### Operador , (coma)

Se usa para:

1. Separar **elementos** en listas: arrays, objetos, argumentos de funciones.
2. En expresiones, **evalúa múltiples expresiones** y retorna la última.

#### Ejemplo 1: Separación en listas

```
let frutas = ["manzana", "pera", "uva"];  
let persona = { nombre: "Ana", edad: 30 };  
function saludar(nombre, edad) {  
  console.log(`Hola ${nombre}, tienes ${edad} años`);  
}  
saludar("Ana", 30);
```

#### Ejemplo 2: En expresiones

```
let x = (1 + 2, 3 + 4, 5 + 6);  
console.log(x); // 11 → se evalúan todas, pero se retorna la última (5 + 6)
```

### Operador : (dos puntos)

Se usa principalmente en:



1. **Objetos literales** → para asignar valores a propiedades.
2. **En estructuras como switch.**

### Ejemplo 1: En objetos

```
let persona = {  
  nombre: "Juan", // clave: valor  
  edad: 35  
};  
  
console.log(persona.nombre); // "Juan"
```

### Ejemplo 2: En switch

```
let dia = "lunes";  
  
switch (dia) {  
  case "lunes":  
    console.log("Es lunes"); // ← los dos puntos van después del `case`  
    break;  
  default:  
    console.log("Otro día");  
}
```

## Operador instanceof

Ya te lo había explicado antes, pero te lo resumo con otro ejemplo para reforzar:

Se usa para verificar si un objeto fue creado por una **clase o constructor específico**.

```
function Animal(nombre) {  
  this.nombre = nombre;  
}  
  
let perro = new Animal("Firulais");  
  
console.log(perro instanceof Animal); // true  
console.log(perro instanceof Object); // true  
console.log(perro instanceof Array); // false
```

## Operador typeof

Sirve para conocer el **tipo de dato primitivo** de una variable.

```
let nombre = "Juan";  
let edad = 30;  
let esActivo = true;  
  
console.log(typeof nombre); // "string"  
console.log(typeof edad); // "number"  
console.log(typeof esActivo); // "boolean"
```



```
console.log(typeof undefined); // "undefined"  
console.log(typeof null); // "object" (quirk de JavaScript)
```

## Operador instanceof

Sirve para comprobar si un objeto es **instancia de una clase** o de un constructor.

```
class Persona {}  
let p = new Persona();  
  
console.log(p instanceof Persona); // true  
console.log(p instanceof Object); // true  
  
let fecha = new Date();  
console.log(fecha instanceof Date); // true  
console.log(fecha instanceof Object); // true  
console.log(fecha instanceof Array); // false
```

## Operador [ ]

Se usa para:

- Acceder a elementos de un **array**
- Acceder a propiedades de un **objeto** mediante **notación por corchetes**
- Crear arreglos o claves dinámicamente

```
// Array  
let frutas = ["manzana", "pera", "naranja"];  
console.log(frutas[0]); // "manzana"  
  
// Objeto con clave dinámica  
let persona = {  
  nombre: "Ana",  
  edad: 25  
};  
  
let clave = "nombre";  
console.log(persona[clave]); // "Ana"  
  
// También se puede asignar propiedades así  
persona["apellido"] = "Gómez";  
console.log(persona.apellido); // "Gómez"
```

## Operador { }

Se usa para:

- Definir **objetos literales**
- Crear **bloques de código**



```
// Objeto literal
let coche = {
  marca: "Toyota",
  modelo: "Corolla"
};

console.log(coche.marca); // "Toyota"

// Bloque de código
if (true) {
  let mensaje = "Hola";
  console.log(mensaje); // "Hola"
}
```

## Operador delete

Se usa para **eliminar propiedades de un objeto**.

```
let usuario = {
  nombre: "Carlos",
  edad: 40
};

delete usuario.edad;

console.log(usuario); // { nombre: "Carlos" }
console.log(usuario.edad); // undefined
```

No funciona para eliminar variables declaradas con `let`, `const` o `var`.

```
let x = 10;
delete x; // false (no se puede eliminar una variable local)
```

## Preguntas

1. Que hace el operador `.`?
2. Ejecuta este código y explícalo
 

```
var s = new String('rafa');document.write(s);
var longitud = s.length;document.write(longitud);
var pos = s.indexOf("fa"); document.write(pos);
```
3. que hace el operador `[]`?
4. Ejecuta estos códigos y explica que hacen
 

```
var arraya=["Santiago","Coruña", "Lugo"];
// var arrayb={ elem1:"Santiago", elem2:"Coruña", elem3:"Lugo" };
document.write(arraya[1]);
// document.write(arrayb["elem3"]);
```
5. que hace el operador `{ }`?
6. Ejecuta estos códigos y explica que hacen
 

```
var arrayb={ elem1:"Santiago", elem2:"Coruña", elem3:"Lugo" };
document.write(arrayb["elem3"]);
```
7. Que hace el operador `Delete`?
8. crea un ejemplo de delete para eliminar la propiedad `elem3`
9. que hace el operador de escape `\` ?
10. crea un ejemplo para escribir la cadena literal “<el pajarito “pedro” comia “grillos”>”  
Ojo a lo mejor debes usar el las entidades html
11. que hace el operador *instanceof*?
12. Ejecuta el código
 

```
a = new Array(1,2,3);
a instanceof Array; // devolverá true.
```
13. que hace el operador *coma*? Operador . *Coma operador : instanceof*
14. Ejecuta este código
 

```
var nombre="juan", direccion="C\ pepep,1", apellidos="lopez Abad", edad="30";
Document.write( nombre + "," + apellidos + "," + direccion + "," + edad );
```
15. que hace el operador `:`?
16. Ejecuta este código
 

```
var a =10;
var b = 15;
var opcion= b>a ? "B es mayor que A": "A es mayor que B";
document.write( "opcion "+opcion);
```
17. Ejecuta este código
 

```
var a =confirm("Estas de Acuerdo");
var opcion= a ? "Acuerdo": "En Desacuerdo";
document.write( "opcion "+opcion);
```
18. que hace el operador *typeof*?



19. Ejecuta este código

```
var a="hola" , b=34, c=3.1e3 d = 34;  
var at =typeof(a) ;  
var bt =typeof(b) ;  
var ct =typeof(c) ;  
document.write( " tipo de "+ a: + " es "+at);  
document.write( " tipo de "+b: + " es "+bt);  
document.write( " tipo de "+ c: + " es "+ct);
```