

UD5 RESUMEN ULTRACONDENSADO



POO en JavaScript – Resumen en 1 Página

- ◆ JS es un lenguaje basado en objetos y prototipos.

Todo en JS es un objeto o se comporta como tal (primitivos aparte).

◆ JSON

- Formato ligero de intercambio de datos.
 - Usa `{}` para objetos, `[]` para arrays.
 - Strings y claves → comillas dobles.
 - Conversión:
 - `JSON.parse(cadena)`
 - `JSON.stringify(objeto)`
-

◆ Objetos en JS

- Acceso: `obj.prop` o `obj["prop"]`
 - Comprobar tipo: `typeof`
 - Comprobar prototipo: `instanceof`
 - Crear objetos:
 - Literal: `{prop: valor}`
 - `new Object()`
 - `this` referencia al objeto actual dentro de métodos.
-

◆ Constructores

- Se usan con `new`.
 - Inicializan propiedades.
 - En clases modernas → método `constructor()`
-

◆ Herencia

- `class Hija extends Madre`
 - `super()` llama al constructor padre.
 - Objetos heredados comparten métodos y propiedades de la superclase.
-

◆ Polimorfismo

- Mismo método en clases diferentes → ejecuta comportamientos distintos.
 - Permite abstracción de acciones.
-

◆ Recorrer objetos

- `for...in` → recorre propiedades (incluye prototipos).
 - `Object.getOwnPropertyNames()` para obtener solo propiedades propias.
-

◆ Eliminar propiedades

```
delete objeto.prop;
```

◆ Prototipos

- JS usa herencia prototípica: todos los objetos tienen un `prototype`.
 - Permite añadir métodos dinámicamente:
 - `Constructor.prototype.metodo = function(){}`
-

◆ Objetos predefinidos

String

- Métodos comunes: `slice`, `replace`, `indexOf`, etc.

Date

- Fechas y tiempos.
- Métodos para extraer año, mes, día, etc.

Math

- Constantes matemáticas.
- Métodos: `sqrt`, `pow`, `random`, etc.

Boolean

- No usar `new Boolean()` → resultados inesperados en condicionales.

Expresiones Regulares

- `/patron flags`
- Métodos principales:
 - `test()`
 - `exec()`
- Modificadores y patrones: `.`, `[]`, `()`, `|`, `+`, `,`, `{n}`, `\d`, etc.