

PDP TP1

NOMBRE: JENNIFER AGUILAR

Ejercicio 1

Parte A

1. Generalización simbólica: ¿Cuáles son las reglas escritas del lenguaje?

La sintaxis de JavaScript es un conjunto de reglas que definen la estructura de un programa, definidas por el estándar **ECMAScript**, que determinan cómo escribir declaraciones, funciones, clases, expresiones y estructuras de control (**if**, **for**, **while**, **switch**).

Incluye tipos de datos como **Number**, **String**, **Boolean**, **Object**, **Array**, y reglas para manipularlos dinámicamente.

Permite definir variables con **var**, **let** y **const**.

2. Creencias de los profesionales: ¿Qué características particulares del lenguaje se cree que sean "mejores" que en otros lenguajes?

Programadores valoran a JavaScript por su **versatilidad**: se puede usar tanto en el navegador como en el servidor gracias a Node.js. Otra cosa muy mencionada es la gran cantidad de **frameworks y librerías** (React, Angular, Vue) que lo hacen muy práctico para proyectos modernos. Aunque a veces se critica que su tipado dinámico puede causar errores, la mayoría lo ve como una ventaja para programar rápido y con menos restricciones.

Parte B

1. ¿Tiene una sintaxis y una semántica bien definida? ¿Existe documentación oficial?

Sí, están bien definidas en **ECMAScript**. Esta Documentación oficial existe en **ECMA** y **MDN Web Docs**.

2. ¿Es posible comprobar el código producido en ese lenguaje?

Se puede probar directamente en un navegador o con Node.js, lo que permite ver rápido si funciona.

3. ¿Es confiable?

Es confiable porque se ejecuta en cualquier navegador o servidor, pero puede dar errores por un **tipado dinámico**, si no se controla bien.

4. **¿Es ortogonal?**

No es ortogonal porque algunas instrucciones no siempre se comportan igual.
Por ej: la diferencia entre `==` y `===`, o el uso de **this** en distintos contextos.

5. **¿Cuáles son sus características de consistencia y uniformidad?**

Es bastante consistente en el uso de condicionales y ciclos, pero presenta detalles confusos, como la coerción de tipos o el uso distinto de **var**, **let** y **const**.

6. **¿Es extensible? ¿Hay subconjuntos de ese lenguaje?**

Si es extensible gracias a la gran cantidad de librerías y frameworks (**React**, **Angular**, etc...) Existen versiones y subconjuntos como **ES5** y **ES6** y también superconjuntos como **TypeScript**.

7. **El código producido, ¿es transportable?**

Totalmente, porque el mismo código se puede correr en navegadores, distintos sistemas operativos e incluso en servidores con Node.js.