Aplicando lo aprendido 1

Alumna: Angelina Garavaglia.

Java Script

PARTE A

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

Las reglas escritas del lenguaje son:

Sintaxis y gramática:

* Definidas por la *ECMAScript Language Specification* (estándar ECMAScript de ECMA International).
* Incluye:
* Uso de palabras reservadas (if, while, for, return, etc.).
* Estructura de declaraciones (let x = 5;).
* Expresiones aritméticas y lógicas (a + b, a && b, !c).
* Definición de bloques de código con llaves { ... }.
* Reglas de alcance de variables (scope léxico).
* **Estructuras de control (núcleo de lo estructurado)**
* Secuencia: ejecución línea por línea.
* Selección: condicionales (if, if...else, switch).
* Iteración: bucles (for, while, do...while).
* **Reglas de tipado y valores**
* Tipado dinámico: variables no tienen tipo fijo, pero los valores sí (Number, String, Boolean, Object, etc.).
* Conversión implícita de tipos (coerción).
* **Reglas de declaración y ámbito**
* Declaración de variables con let, const y var (este último heredado).
* Ámbito de bloque vs. ámbito de función.
* **Reglas de funciones**
* Definición de funciones con function o como expresiones (const f = function(...) {...};).
* Parámetros formales y reales, valores por defecto, paso por valor/referencia.

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

Se cree que JavaScript es fácil de aprender porque su sintaxis básica es cercana al pseudocódigo y permite a principiantes empezar rápido.

Con unas pocas líneas ya se pueden ver resultados”

Profesionales destacan la **libertad** que da no declarar tipos de variables estrictamente.

Se considera más expresivo que lenguajes estructurados clásicos como C.

Existe la percepción de que cualquier problema ya tiene solución en foros, librerías o documentación.

JavaScript es accesible, ubicuo, flexible y práctico, incluso en su uso estructurado. Se valora sobre todo su rapidez para construir y probar ideas, su portabilidad universal en navegadores y la fuerza de su comunidad.

PARTE B

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

Sí, JavaScript tiene una sintaxis y una semántica bien definida, estandarizada en **ECMAScript (ECMA-262)**.

la documentación oficial del lenguaje es la **especificación ECMAScript**, publicada por **ECMA International** (documento técnico ECMA-262).

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

En **JavaScript** puro (estructurado o no), la comprobación estricta de**l código** es limitada: solo se asegura que la sintaxis sea correcta, pero no se previenen errores lógicos o de tipos antes de ejecutar.

1. ¿Es confiable?

JavaScript es confiable en cuanto a estandarización y portabilidad, pero limitado en cuanto a comprobación de errores por su tipado dinámico.

**JavaScript con buenas prácticas y herramientas modernas**: puede ser altamente confiable para desarrollar aplicaciones críticas y escalables.

1. ¿Es ortogonal?

Java script no es ortogonal, aunque en su uso estructurado básico mantiene cierta regularidad en la estructura de control y funciones. Por eso se considera un lenguaje **pragmático y flexible.**

1. ¿Cuáles son sus características de consistencia y uniformidad?

* Consistencias: Las estructuras de control (if, while, for, switch) funcionan de manera uniforme y predecible.
* Las funciones se pueden declarar y llamar en cualquier parte (respetando el ámbito), con sintaxis clara.
* La evaluación condicional trata a cualquier valor como *truthy* o *falsy*, lo cual es consistente, aunque pueda sorprender al principio.

Uniformidad

* Solo existe un tipo de estructura de función básica (function), aunque hoy también tenemos funciones flecha.
* Todas las estructuras de control usan llaves {} para delimitar bloques.
* Existe un único mecanismo para agrupar valores compuestos: los **objetos**, y a partir de ahí se construyen arrays y otras estructuras.

1. ¿Es extensible? ¿Hay subconjuntos de ese lenguaje?

sí es extensible, principalmente a través de librerías, prototipos y supersets **como TypeScript**, aunque no es extensible de forma formal en el núcleo de su especificación (ECMAScript se mantiene estándar).

Si existen subconjuntos de java script y varios, subconjunto estructurado, stric mode ("use strict"), safe subset (caja segura), subconjuntos de proposito educativo o estandarizado, ASM.js.

7) El código producido, ¿es transportable?

Si el código producido es transportable en gran medida gracias a la estandarización y la disponibilidad de motores en prácticamente todas las plataformas.

No es 100% transportable si se usan APls especificas del entorno (ejemplo document.getElementById no funciona en Node.js).