

Trabajo práctico 2

1A.1

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

JavaScript es un lenguaje de programación de scripting, interpretado — más precisamente, de compilación just-in-time—, multiparadigma, con funciones de primera clase, de un sólo hilo.

Como *lenguaje de scripting*, JavaScript está diseñado para ejecutarse en un entorno hospedado, por lo que no posee procesos de entrada o salida y depende del entorno para comunicarse con el mundo exterior.

JavaScript nos permite escribir usando varios estilos o paradigmas. Está orientado principalmente a Programación Orientada a Objetos basada en prototipos, pero soporta el uso de clases. En JavaScript, las funciones son de primer nivel, lo que significa que son tratadas como objetos. Esto nos permite que no haya diferencia práctica entre una variable y una función, lo cual es una característica de los lenguajes funcionales.

JavaScript sólo puede realizar una tarea por vez. Esto puede parecer negativo, pero en realidad abre la puerta a varias ventajas.

La sintaxis de Javascript es similar a C y Java: distingue entre mayúsculas y minúsculas, las declaraciones terminan con punto y coma y las llaves se utilizan para marcar bloques de código. Los nombres de variables y funciones utilizan símbolos alfanuméricos más guiones bajos, y los operadores aritméticos, de comparación y lógicos son los mismos que en C y Java.

En resumen:

- 1) Es un lenguaje multiplataformas.
- 2) Es un lenguaje interpretado y orientado a objetos.
- 3) Es un lenguaje de tipado débil y alto nivel.
- 4) Cuenta con una gran cantidad de librerías y frameworks.
- 5) Se puede ejecutar con asincronía.
- 6) Permite trabajar con JSON y conectarse a bases de datos.

1A.2

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

Java es un lenguaje de programación orientado a objetos. Javascript es un lenguaje que permite programar sin necesidad de crear clases y a diferencia de Java no requiere hacer la programación orientándose a objetos, aunque sí lo permite.

Una de las razones de su popularidad y su amplia adopción es su versatilidad. Admite desarrollo multiplataforma y puede ser usado tanto para aplicaciones del lado del servidor como del lado del cliente.

JavaScript es un lenguaje de programación ligero con un sistema de tipado dinámico, lo que facilita la ejecución de scripts durante el desarrollo de aplicaciones web. Además, su compatibilidad con diferentes navegadores hace que el desarrollo sea más manejable, ya que los desarrolladores no tienen que preocuparse por cómo se ejecutará su código en otros navegadores.

Además:

Interactividad en tiempo real.

Detección y respuesta a eventos.

Manipulación sencilla del DOM.

Creación de formularios interactivos.

Innovación en la carga de contenido dinámico.

Animación y efectos visuales para los sitios web.

Desarrollo de juegos y aplicaciones interactivas.

1b.1

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

Sí. JavaScript tiene una sintaxis y una semántica bien definidas, regidas por el estándar ECMAScript, mantenido por ECMA International. La versión más reciente del estándar especifica la sintaxis, semántica, y comportamiento del lenguaje.

1b.2

¿Es posible comprobar el código producido en ese lenguaje?

Sí. Es posible comprobar el código JavaScript de varias maneras:

Linting: Herramientas como ESLint permiten analizar el código fuente en busca de errores de sintaxis y estilo.

Testing: Existen frameworks como Jest, Mocha o Jasmine que permiten escribir y ejecutar pruebas automatizadas.

Depuración: Los navegadores modernos y entornos como Node.js ofrecen herramientas de depuración (debugging) que permiten verificar el comportamiento del código.

1b.3

¿Es confiable?

Sí, pero con reservas. JavaScript es confiable en el sentido de que es utilizado en millones de aplicaciones web y es el lenguaje principal del desarrollo web front-end. Sin embargo, la confiabilidad del código depende de cómo se use el lenguaje.

1b.4

¿Es ortogonal?

No completamente. La ortogonalidad en un lenguaje de programación se refiere a que las operaciones pueden combinarse libremente con

independencia de su contexto. JavaScript no es completamente ortogonal porque ciertas construcciones y características pueden comportarse de manera diferente según el contexto en el que se utilicen.

1b.5

¿Cuáles son sus características de consistencia y uniformidad?

Moderadas. JavaScript es bastante consistente en cuanto a su sintaxis, pero existen algunas inconsistencias y peculiaridades históricas (por ejemplo, el uso de `==` versus `===`). A lo largo del tiempo, ECMAScript ha trabajado para mejorar la consistencia, pero algunas de estas peculiaridades persisten por razones de compatibilidad hacia atrás.

1b.6

¿Es extensible? ¿Hay subconjuntos de ese lenguaje?

Sí. JavaScript es extensible en varios sentidos:

Bibliotecas y frameworks: Los desarrolladores pueden crear y usar bibliotecas y frameworks para extender la funcionalidad de JavaScript.

Subconjuntos: Existen subconjuntos de JavaScript, como TypeScript, que añade tipado estático, o JSX, una extensión utilizada en React.

Polyfills: Se pueden usar polyfills para añadir nuevas funcionalidades en navegadores que no las soporten nativamente.

1b.7

El código producido, ¿es transportable?

Sí. JavaScript es altamente transportable, ya que el mismo código puede ejecutarse en diferentes navegadores y plataformas (Windows, macOS, Linux, dispositivos móviles, etc.). Sin embargo, en entornos fuera del navegador (como Node.js) o en versiones más antiguas de navegadores, algunas características podrían no estar disponibles, lo que podría requerir polyfills o modificaciones menores.