

FLIPPED CLASS INTRODUCTION

El objetivo de la flipped classroom es que la teoría se vea en casa, y los ejercicios se hagan en el aula. Siguiendo este procedimiento, las lecturas/actividades propuestas son:

Nº	Descripción	Enlace	Tiempo aproximado (min)
1	Por qué Javascript e intro	Este documento sección 1-2-3	20-30 si no se tienen las cosas instaladas
2	Introducción a la IA en programación	Este documento sección 4	10
3	Ejercicios de iniciación 1	https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_variables1	5
4	Ejercicios de iniciación 2	https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_variables2	5
5	Ejercicios de iniciación 3	https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_variables3	5
6	Ejercicios de iniciación 4	https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_variables4	5
7	Ejercicios de iniciación 5	https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_variables5	5
8	Ejercicios de iniciación 6	https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_operators1	5
9	Ejercicios de iniciación 7	https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_operators2	5
10	Ejercicios de iniciación 8	https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_operators3	5
11	Ejercicios de iniciación 9	https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_operators4	5

12	Ejercicios de iniciación 10	https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_operators5	5
----	-----------------------------	---	---

1. ¿Por qué JavaScript?

Esta pregunta es muy importante, debemos entender que hay muchísimos **lenguajes de programación, y todos son igualmente válidos**. Algunos son más complejos, otros más sencillos de aprender, otros más eficientes, otros más encaminados a aplicaciones concretas. Todos los lenguajes son buenos y todos nos abren la puerta al mundo de la tecnología. Generalmente, si sabemos un lenguaje, casi podemos descifrar, con un poco de esfuerzo y ganas, cualquier otro. (Cuidado con C!)

Dicho esto, existen estudios (a día de hoy) que hablan de la **escasez de programadores en el ámbito de la web** (oh, amigo, internet, claro). Y uno de los lenguajes con los que se comunican los ordenadores para poder sacar por los navegadores información de manera bonita y útil es JavaScript. Antes se usaba más PHP, pero ha caído en desgracia y ahora todo se basa en JavaScript.

JavaScript es un nombre que puede parecer confuso, ya que el lenguaje, aunque basa sus raíces “sintácticas” en Java, no tiene mucho que ver con el propio lenguaje de Java. Además, ahora es muchísimo más utilizado que Java, sin duda. Originalmente, JavaScript fue un lenguaje creado por el actual Oracle para usar Netscape, ahora conocido como Mozilla. **El estándar de JavaScript se denomina como ECMAScript.**

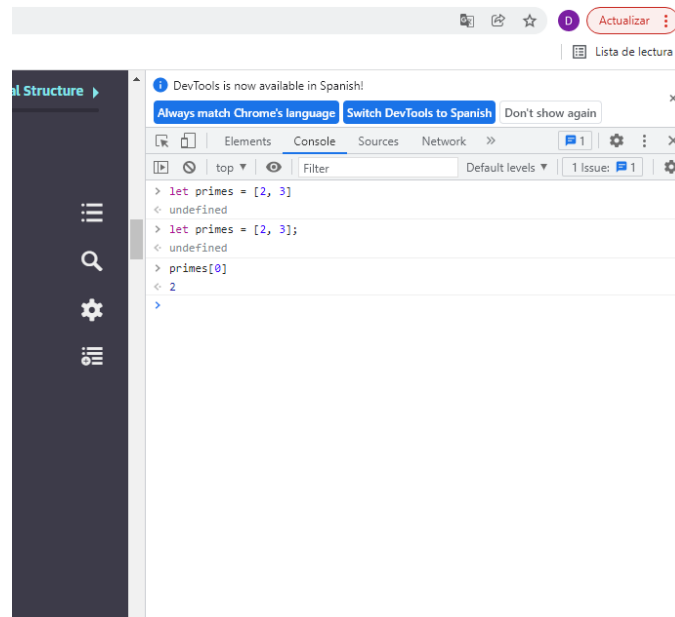
Originalmente, JavaScript se ejecutaba únicamente en los navegadores, al contrario que PHP, que lo hacía en los servidores. Por ello, ésta sigue siendo la ejecución más normal del código. El lenguaje se encuentra en el navegador y por medio de peticiones HTTP obtiene datos y los expone en el navegador haciendo uso de CSS y HTML.

Hoy en día, viendo la importancia que cobraba el lenguaje se creó otra manera de trabajar con JavaScript en el lado de los servidores, y se llamó Node. Node ahora es una elección típica para trabajar con servidores web y eso os tocará aprender más adelante.

2. ¿Dónde vamos a trabajar?

Aquí, cada maestrillo con su librillo, podéis usar lo que consideréis más importante. Sin embargo, nosotros utilizaremos **VisualStudio code**, para prepararnos para nuestras futuras clases, y por ello nos suscribiremos a eso. Si alguna persona prefiere trabajar con otro IDE adelante, pero los problemas que deriven de esa razón deberán ser solventados por el propio usuario. (aunque intentaremos ayudar)

- También se puede usar la consola del propio navegador (en este caso, utilizaremos Google Chrome)

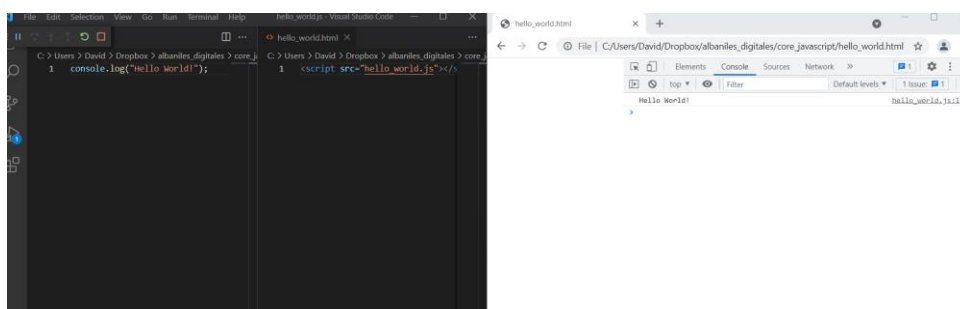


- Nos descargaremos Node.js para tenerlo a punto: <https://nodejs.org/en/>. Esto nos puede ser útil a veces cuando necesitemos probar cosas de manera interactiva.
- Y, por último, Visual Studio Code: nos lo descargaremos en: <https://code.visualstudio.com/>

3. Hello World

Vamos a hacer una prueba sencilla:

- Crea un archivo en vscode llamado `hello_world.js`
- Escribe un log de consola con: `console.log("hello world")`
- Crea un archivo en vscode llamado `hello_world.html`
- Escribe la detección del archivo de js con: `<script src="hello.js"></script>`
- Ahora ejecuta el código html y saca la pantalla de herramientas para desarrolladores. Deberías de tener algo así:



4. IA

a. Introducción

Hoy en día, la inteligencia artificial ha llegado también a los programadores, de momento con la incertidumbre de saber si acabará supliéndolos o no por completo. Lo que está claro es que hoy en día es una buena herramienta para mejorar los procesos y la eficiencia de los programadores, siempre y cuando se utilice con comprensión y supervisión.

Cuando hablamos de programar con IA, se trata de conseguir que la inteligencia artificial nos cree código para lo que queremos hacer, y se me ocurren hoy en día dos maneras más típicas de hacerlo:

- Preguntar a un Bot tipo chatgpt para que te resuelva el código
- Instalarte un asistente de IA tipo Copilot para que te vaya sugiriendo código conforme lo vas haciendo.

b. Precauciones

El código generado por IA, a pesar de poder mejorar nuestros procesos y nuestros tiempos, no está exento de problemas, por ello, a la hora de usarlo hay que tener cuidado y tener en cuenta una serie de precauciones:

- Investigaciones sobre copilot, por ejemplo, aducen que el 40% de las veces el código tiene fallas de seguridad
- Siempre se debería revisar lo que se ha escrito en código con diferentes herramientas: pylint, comprobadores de dependencias, comprobadores de seguridad...
- A veces pueden generar estructuras demasiado complejas para lo que se está llevando a cabo. Las nomenclaturas también deben ser coherentes y revisadas.
- Siempre hay que entender el código creado por IA, si no, puede dar lugar a errores o a no entender el propio programa de uno.
- Tener en cuenta los problemas con los copyrights, ¿se está usando código de otras personas con la IA utilizada?
- Lectura extra: <https://madappgang.com/blog/chat-gpt-code-errors/>

c. Github copilot

Es una herramienta que se puede instalar en vscode, y por eso nos resulta interesante en nuestras prácticas. Sus características principales son:

- Proporciona sugerencias de código en tiempo real

- Se integra fácilmente con vscode y con otros
- Sus sugerencias están basadas en el propio contexto del proyecto, de tal manera que crea estructuras y nombres coherentes
- Mejora productividad ya que en muchos casos (testing) ahorra tiempo.
- Cierta privacidad y seguridad, según otros revisores

BIBLIOGRAFÍA

Castles, R. (2022, 4 noviembre). *Why You Should Apply Caution When Using AI in Code Development*. Spiceworks Inc. <https://www.spiceworks.com/tech/artificial-intelligence/guest-article/why-you-should-apply-caution-when-using-ai-in-code-development/>

Flanagan, D. (2020). *JavaScript: The Definitive Guide : Master the World's Most-used Programming Language*. O'Reilly Media.

Polo, J. D., & Polo, J. D. (2023, 3 agosto). *Inteligencia Artificial para desarrolladores, recursos que ayudan a programar*. WWWhat's New. <https://www.whatsnew.com/2023/06/01/inteligencia-artificial-para-desarrolladores-recursos-que-ayudan-a-programar/#:~:text=En%20el%20mundo%20de%20la,la%20vida%20de%20los%20desarrolladores>