# Como estudiar Javascript

Este documento no trata de Javascript, no explica nada de Javascript (aunque usa algunos ejemplos). Habla de cómo estudiar este tema, cómo lograr entenderlo y tips en ese sentido.

Con Javascript, como con cualquier cosa de esta carrera, **no hay que preocuparse, pero si ocuparse**. Es lo difícil de la materia porque es cuando empezamos a programar. Sin programar solo podemos hacer una página estática, sin interactividad, y la idea en esta carrera es llegar a programar aplicaciones. Las aplicaciones son interactivas, o sea, interactúan con el usuario.

El cambio de chip viene porque ahora ustedes no son los que usan la app, son los que hacen la app. No son ni el usuario, ni la app, son el programador. **El programador hace la app, diciéndole a la computadora qué hacer**. Para eso tienen que poder imaginarse que haría la computadora (si ustedes fueran la computadora), y decírselo con órdenes que entienda la computadora.

Para decirselo tienen que **usar un lenguaje que la computadora entienda**. En este caso Javascript. El lenguaje **tiene sus reglas**, o sea, para decir algo se escribe de una forma y no de otra. Todo eso tiene una estructura (una sintaxis, gramática, etc), que hay que respetar. No hacemos hincapié en los detalles esos porque se aprenden naturalmente viendo ejemplos, pero para que entiendan a qué nos referimos va un ejemplo.

| | |
|---|---|
| `function funcion() {`<br>`  console.log("hola");`<br>`}`<br><br>`//CORRECTO` | `function funcion() {`<br>`{`<br>`console.log("hola");`<br>`}`<br><br>`//INCORRECTO: abre una { de más, no respeta la sintaxis de Javascript` |

Las slides y ejemplos están organizados para mostrar una o dos cositas en cada ejemplo (esas que se marcan en las slides de tildes). Pero esas cosas terminan siendo unas varias, ponele 7 que se usan todas juntas. **Parece difícil porque hasta que no entendes todas esas cosas básicas no podes hacer casi nada. Una vez que las entendes podes hacer todo.**

Justamente por eso los ejemplos estos los hemos cambiado tantas veces a lo largo de los años. Buscando una serie de “paso a paso” que nos permita mostrar cada partecita sin las otras. Por eso los primeros dos ejemplos son tan feos (con un alert).

**Lo mejor es cada una de estas cosas aprenderlas de a una.** Copien de 0 (**escriban el programa que está en las slides desde cero, sin copiar y pegar**) cada uno de los ejemplos al navegador y haganlos andar. La idea de escribirlo (sin copy&paste) es ejercitar la memoria. Así memorizamos las reglas de JS, no queda otra que por repetición. Después de copiarlo **hagan una pequeña modificación** que se les ocurra. Por ejemplo: qué pasaría si en lugar de “hola” le pongo “chau”? Qué debería hacer? **Y recién después de suponer lo verificas**. Y asi vas probando, pero no probando a ver que hace, es importante

primero suponer que hace, probando a ver si acertaste. **En ese proceso vas a entender cómo funciona**. Eso con cada ejemplo. Tantas modificaciones como necesites para estar seguro que lo que está en la slide del tilde lo entendes y lo podes volver a hacer casi de memoria. En el proceso de aprendizaje **SIEMPRE** es recomendable ir de lo **simple** a lo **difícil**

Luego vas a la guía de prácticos y te fijas cada uno de esos. En general se parecen a algun ejemplo o ejercicio anterior. A cada uno lo agarras de a uno y los vas intentando resolver. **Los pasos son similares a la secuencia de ejemplos que vimos**: agregar un JS, configurar un evento, leer algo del DOM, HACER ALGO, escribir el DOM. El HACER ALGO es lo que más cambia en cada ejercicio según lo que te está pidiendo. **Por eso es importante haber reproducido los ejemplos y haberlos entendido bien** antes de hacer los ejercicios (también se pueden intercalar a medida que los necesitas, pero hay que poder diferenciar cual necesitas). En esta altura de la materia/carrera son cosas chicas que hay que hacer, en el ejemplo de los dados el HACER ALGO es “generar un número aleatorio y contar si salio 8”. A lo largo que avancemos vamos a poder hacer cosas más complicadas ahí (ver el tiro que más sumó, calcular el promedio y subirlo a un servidor por ejemplo).

**Siempre la forma de encarar el problema es por etapas.** Primero ves que pide el ejercicio, después qué solución se les ocurre "contando que harían" en español mismo, después viendo qué y dónde hacer conexiones (que botones, que haría cada función, identificando las partes que se nombran en el párrafo anterior), finalmente haciendo el código y verificando si funciona de acuerdo con lo que pensamos con la consola abierta. **Muchas veces (casi siempre) descubrimos cómo funciona el código con errores o porque hace algo distinto a lo que habíamos pensado que iba a hacer**. En general, es mejor ir aproximándose a la solución en pequeños pasos pero firmes donde entendamos bien cada uno de esos pasos. Resolver de manera simple suele ser un desafío, y si bien es bueno ser curioso y ver otros contenidos o información hay que evitar buscar soluciones complicadas donde el código difiera mucho del contenido de las filminas o de los ejemplos. Copiar y pegar una solución más avanzada es una buena receta para no entender nada y que nos juegue en contra a la larga. Al avanzar la carrera van a poder buscar cosas más complejas en la web y entenderlas fácilmente.

Cuando la materia es presencial esto lo evaluamos en papel, no nos interesa el detalle del “;” o la “{“ de JS para que ande. Sino la lógica. La mejor técnica que conozco para eso, me la recomendó una alumna que le sirvió para aprobar (muy bien) el recuperatorio. Consiste en **agarrar los prácticos nuevamente, intentar hacerlos en papel, luego pasarlos a la PC. Hacer los ajustes necesarios para que ande. Pasar esos ajustes al papel como si te estuvieras corrigiendo.** Y así con cada ejercicio. La memoria se ejercita con repetición, al igual que los músculos, solo que el tipo de “pesas” es otro. Este año de cuarentena seguramente no aplique demasiado, pero no deja de ser un excelente ejercicio para integrar más el lenguaje a la cabeza. Lo sugerimos hacer una vez que ya en computadora se sienten más cómodos.

Al saltar un error también tenemos que aprender a leer los errores. A veces no son en la línea que nos dice, sino en una línea anterior. A medida que vamos comprendiendo el lenguaje los detectamos cada vez más fácil.

Y a todo este va a pasar muchas veces “creí que entendía, pero no anda!”. Los errores pueden venir de muchos lados, muchas veces la idea es correcta pero le erramos al escribirlo, otras veces no tuvimos en cuenta un paso. Hay que distanciarse de uno mismo, poner en jaque lo que suponemos nosotros mismos, ser un poco abogado del diablo como dice el dicho. Porque las órdenes a la computadora se las dimos nosotros, y es muy obediente, el problema está en que nosotros creemos en que las órdenes son

correctas, no en que haga algo mal la computadora. **Tenemos que usar herramientas como el "console.log" para entender lo que realmente está pasando. En ese proceso es cuando realmente se aprende.** Se usa como explicamos, en tu programa las cosas deberían ser de una manera y no de otra según lo que vos supones. Por ejemplo, si el usuario puso web la variable nombre debiera ser "web", agregá un console.log para verificarlo. Esta función debería ejecutarse en X momento, agrega un console.log para ver si está llamándose. Y así con cada suposición. En un momento llegas a cuál es la diferencia. Y ahí podes analizar la causa con otros ojos.

En esta clase de ejercicios, es muy importante desarrollar **tolerancia a la frustración**. Es dedicarle mucho a práctica, que las cosas no te salgan y seguir intentando. La satisfacción que se tiene cuando anda es la parte linda. También es importante **administrar el tiempo**, estar 3hs con el mismo ejercicio no tiene sentido. Pedí ayuda antes y mientras te responden seguí con otro. Media hora bloqueado es suficiente, dejalo de lado y seguí con otra cosa. Pregunta por algun lado, o volver al dia siguiente. Muchas veces volver a ver el problema más tarde luego de ver otro ejercicio o de descansar/distraerse también ayuda a no quedarse enfrascado en un ejercicio donde ya estamos sugestionados y no veíamos los errores. Recuerden el concepto de "time-boxing", los ayudará mucho a administrar el tiempo y avanzar de manera más eficiente.

**La catedra de Web 1**