

Indicaciones previas

- De todas las habilidades necesarias para programar, cada un@ de [vosotr@s](#) tenéis un nivel en cada habilidad y una forma de aprender distinta. Aquí os proporciono alternativas para que seleccionéis vuestra forma de estudio particular.
- Os animo a que os leáis este artículo donde se dan pautas y requisitos para aprender JS:
 - <http://wmedia.es/como-aprender-javascript-de-una-vez-por-todas/>
- Tenemos que ser realistas: Para aprender Angular se necesario unos conocimientos mínimos de JavaScript. NO HAY ATAÑOS, programar y JavaScript se aprende practicando.
- En el PDF de [aprende-web.net](#) se resumen muy bien estos conocimientos mínimos:
 - <https://aprende-web.net/javascript/objetos/resumenJavascript.pdf>
- Para asimilar bien todo, lo ideal sería que hiciérais todos los ejercicios de JavaScript Ya, y los temas que no hemos visto en clase, leer la parte teórica y los ejercicios resueltos
 - <https://www.tutorialesprogramacionya.com/javascriptya/>
- Como algunos no podéis dedicar tanto tiempo, en la siguiente página os propongo videotutoriales, temas y ejercicios específicos con los que trabajar. Os animo a ver los que no hayáis comprendido bien, y a practicarlos también
- En la última página, refresco esa “guía de aprendizaje” de programar, los pasos normales para aprender algo nuevo.

VIDEOTUTORIALES

Si os gustan los videotutoriales, la web www.pildorasinformaticas.es tiene tutoriales geniales para JavaScript, HTML y CSS. Concretamente en los de JS, en todos ellos hablan de cosas que ya hemos visto en clase. Según el nivel, os propongo empezar por:

REPASOS

- Nivel 0 ¿Qué es JS?:
 - Video 1: <https://youtu.be/m2nscBtQEIs>
- NIVEL 1. Sintaxis básica y variables.
 - Video 3 al 5: <https://youtu.be/SVmKhuAWvY4>
 - Video 6: Jugar con variables, operadores: <https://youtu.be/f719B0LDpiE>
- NIVEL 2: Arrays
 - Video 7: <https://youtu.be/hTeFMke6F6Q>
 - Video 8: <https://youtu.be/yn-o0rxXW0o>
- NIVEL 4: POO (Introducción Programación orientada a objetos)
 - <https://youtu.be/6wB-0QSkk3Y>
 - https://youtu.be/CfKVU0i7w_w
- NIVEL 5: Estructuras de control de flujo:
 - El IF (Videos 11 al 14): <https://youtu.be/jl2pp0ILs40>
 - Los bucles (videos 15 al 20): <https://youtu.be/-s1c-4mbNSQ>
- NIVEL 6: Funciones
 - Videos 21 AL 23 <https://youtu.be/uUfYLGVLHfI>
 - Funciones y eventos: Video 24: <https://youtu.be/yfWcTBufkcM>

PRACTICA GUIADA (videos 25 al 28):

- https://youtu.be/mQajJT80_c4













NIVEL 7: El DOM, HTML y (con JS e introducción a jQuery):

- https://youtu.be/J8_LGy1gGqoc

MODO DE EMPLEO JAVASCRIPT YA

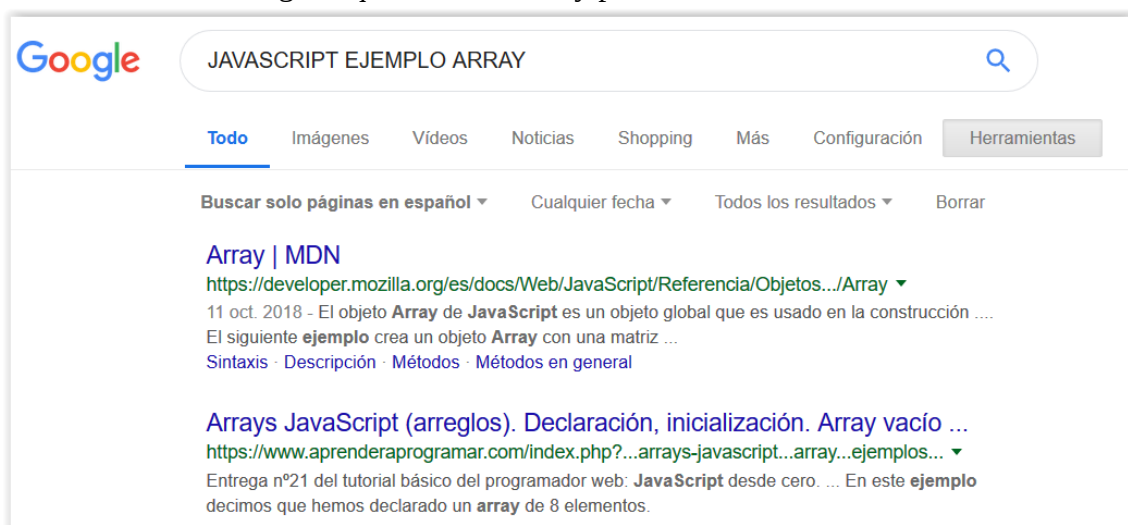
Para no perder mucho tiempo y avanzar rápido, la forma de trabajar debe ser la siguiente:

1. Leer el **concepto**.
2. Si no lo entendiste en clase, ir al **Detalle del concepto**
3. Si más o menos entiendes el concepto pero no bien cómo usarlo, ir al **Problema resuelto**
4. Si más o menos crees que lo tienes controlado, ir al **Problema a resolver** e intenta resolver los ejercicios.

Orden del concepto	Concepto	Detalle del concepto	Problema resuelto	Problema a resolver
21	Clase Array			
22	Clase Math			
23	Clase String			
24	Formularios y Eventos.			
25	Controles FORM, BUTTON y TEXT.			
26	Control PASSWORD			
27	Control SELECT			

5. Si después de un buen rato de escribir, probar, DEPURAR, comprobar errores, etc. no te sale el ejercicio, vuelve a los puntos anterior 1 vez o BUSCA EN GOOGLE otra web donde lo explique u otros ejemplos, así:

1. Primero la palabra JavaScript
2. En herramientas, seleccionar “Buscar sólo en español”
3. Encontrar alguno que lo entiendas, y probarlo con Visual Studio Code



GUÍA DE EJERCICIOS JAVASCRIPT YA

- Fundamentos JavaScript (variables, condicionales y operadores lógicos):
 - <https://www.tutorialesprogramacionya.com/javascriptya/index.php?inicio=0>
- Estructuras de control Switch, bucles, concepto de contador y funciones:
 - <https://www.tutorialesprogramacionya.com/javascriptya/index.php?inicio=10>
- Arrays y formularos:
 - <https://www.tutorialesprogramacionya.com/javascriptya/index.php?inicio=20>
- Eventos y comienzo de programación orientada a objetos:
 - <https://www.tutorialesprogramacionya.com/javascriptya/index.php?inicio=30>
- Más sobre Arrays:
 - <https://www.tutorialesprogramacionya.com/javascriptya/index.php?inicio=40>
- Más sobre arrays. Funciones:
 - <https://www.tutorialesprogramacionya.com/javascriptya/index.php?inicio=50>
- Funciones y eventos:
 - <https://www.tutorialesprogramacionya.com/javascriptya/index.php?inicio=60>
- Otros operadores y expresiones regulares:
 - <https://www.tutorialesprogramacionya.com/javascriptya/index.php?inicio=70>
- Más sobre RegExp y comienzo EcmaScript6:
 - <https://www.tutorialesprogramacionya.com/javascriptya/index.php?inicio=80>
- JavaScript avanzado. Más sobre ES 6:
 - <https://www.tutorialesprogramacionya.com/javascriptya/index.php?inicio=90>

¿Cómo aprender algo nuevo en programación hasta asimilarlo y poder aplicarlo?

- 1 — Ver (escucharlo o leerlo) de que trata, **el concepto**
- 2 — Cuando es algo nuevo (es decir, casi todo), no entenderlo o entenderlo mal (como humanos en nuestra lógica)
- 3 — Ver un **ejemplo práctico ya resuelto**
- 4 — Creer entenderlo pero entenderlo mal
- 5 — Ver otro ejemplo práctico
- 6 — Volver a creer entenderlo, pero esta vez lo que crees se acerca un poco a lo que **de verdad va a hacer la máquina**
- 7 — Los pasos 5 y 6 varias veces, en bucle mientras no te aburras de ver ejemplos
- 8 — Cuando ya no puedes ver más ejemplos porque es un rollo, intentas **ponerlo en práctica para resolver un problema**
- 9 — Fallas porque no funciona como esperabas, ya que **la máquina siempre «piensa» diferente**. Aquí es necesario superar cierto sentimiento de derrota, y motivarse sea como sea.
- 10 — **Pruebas a resolverlo de otra manera**, y vuelves al punto 9, varias veces, hasta que te das cuenta de que algo se te escapa, así que **usas el depurador para ver “qué se te escapa”**.
- 11 — Vuelves al punto 5, esta vez con cierta rabia, y al repasar y ver otros ejemplos, por fin entiendes como se comporta la máquina.
- 11 — Ahora que ya entiendes esa nueva característica del lenguaje, te das cuenta de que lo que has resuelto no soluciona el problema, es decir, que no habías entendido el problema bien.
- 12 — Por lo tanto, intentas **ahora entender el problema, pero con las armas para solucionarlo**. Con cierto sentimiento de dominio, reformulas el problema, vuelves al punto 8, peeeeero pasas el resto de fases más rápido.
- 13 — Casi casi **resuelves el algoritmo (lógica y sintaxis)**, a falta de un par de casos que no habías tenido en cuenta. La sensación es que la victoria está próxima...
- 14 — Y de repente.... !Bingo! ¡EUREKA! ¡Aleluya hermanas y hermanos! El programa funciona, con las pruebas y casos que se te ocurren
- 15 — O bien tú mism@, o un usuario, o bien un compañero, se da cuenta de que **en realidad casi funciona** salvo un caso en el que no funciona. Uyyyyyyyy casi...pero no pasa nada, se corrige por fin, cantamos otra vez victoria esta vez con cautela, nos limpiamos la sangre el sudor y las lágrimas, y con valor, seguimos adelante, sabiendo que ahora somos un poquito más expertos.