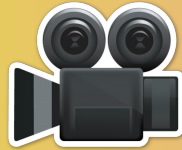




***¡LES DAMOS LA
BIENVENIDA!***

¿Están listos?

***RECUERDA PONER A GRABAR LA
CLASE***

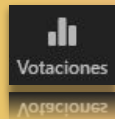


PRESENTACIÓN DE ESTUDIANTES



Por encuestas de Zoom:

1. País
2. Conocimientos previos en HTML, CSS y programación.
3. ¿Por qué elegiste el curso?



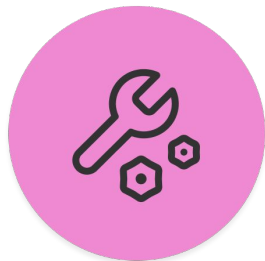


¿DUDAS DEL ON-BOARDING?

MIRALO AQUI

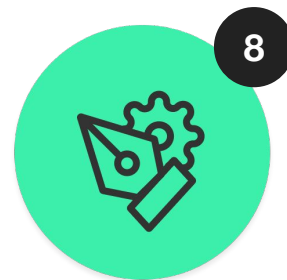
DESAFÍOS Y ENTREGABLES

Son actividades o ejercicios que se realizan durante la cursada, para enfocarse en la práctica.



Desafíos genéricos

Ayudan a poner en práctica los conceptos y la teoría vista en clase. No deben ser subidos a la plataforma.



Desafíos entregables

Relacionados completamente con el **Proyecto Final**. Deben ser subidos obligatoriamente a la plataforma hasta 7 días luego de la clase para que sean corregidos.

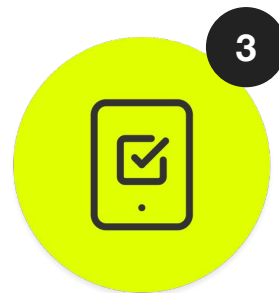
DESAFÍOS Y ENTREGABLES

Son actividades o ejercicios que se realizan durante la cursada, para enfocarse en la práctica.



Desafíos complementarios

Desafíos que complementan a los entregables. Son optativos y, de ser subidos a la plataforma a tiempo y aprobados, suman puntos para el top 10.



Entregas del Proyecto Final

Entregas con el estado de avance de tu **proyecto final** que deberás subir a la plataforma a lo largo del curso y hasta 7 días luego de la clase, para ser corregidas por tu docente o tutor/a.



PROYECTO FINAL

El Proyecto Final se construye a partir de los **desafíos** que se realizan clase a clase. Se va creando a medida que el estudiante sube los desafíos entregables a nuestra plataforma.

El objetivo es que cada estudiante pueda utilizar su Proyecto Final como parte de su portfolio personal.

El **proyecto final** se debe subir a la plataforma la ante-última o última clase del curso. *En caso de no hacerlo tendrás 20 días a partir de la finalización del curso para cargarlo en la plataforma. Pasados esos días el botón de entrega se inhabilitará.*

¿CUÁL ES NUESTRO PROYECTO FINAL?



APLICACIÓN WEB INTERACTIVA

Crearás una página web interactiva en JavaScript que permitirá simular distintos procesos. Un “simulador” es un programa que soluciona ciertas tareas y proporciona al usuario información de valor. Además, utilizarás AJAX y JSON para obtener datos y jQuery para controlar eventos en la interfaz y producir animaciones en respuesta.



PROYECTOS DE NUESTROS ESTUDIANTES

- Tienda de PC: <https://bcoalova.github.io/CODER-js-Tienda/>
- Tienda de Bebidas: <https://vanifederici.github.io/mono-galactico/>
- Conversor de divisas: <https://conversor-de-monedas.000webhostapp.com/index.html>
- Buscador de reservar: <https://antopr.github.io/Javascript-Coder/#>
- Simulador de préstamos: <https://mguidocaruso.github.io/AIPrestamo/>



<i>ENTREGA</i>	<i>REQUISITO</i>	<i>FECHA</i>
1° entrega	Estructura HTML y CSS del proyecto. Variables de JS necesarias. Objetos de JS.	Clase N° 6
2° entrega	Agregar uso de JSON y Storage. DOM y eventos del usuario.	Clase N° 10
3° entrega	Incorporar JQuery para controlar elementos. Sumar efectos y animaciones. Optimizar diseño HTML y CSS	Clase N° 14
Proyecto Final	Simulador final funcionando en un archivo HTML con sus archivos JS complementarios.	Clase N° 16

¡IMPORTANTE!

Los desafíos y entregas de proyecto se deben cargar hasta siete días después de finalizada la clase. Te sugerimos llevarlos al día.

LUNES 16/03 20:30HS

10. Estrategia de contenido para Twitter y LinkedIn

DESAFÍO - EXPIRA EL 23/03/2020

Crear publicaciones para Twitter

👍

● HOY 20:30

📁

Tenes tiempo hasta el
23/03/2020

↑ ENTREGAR



Clase 01. JAVASCRIPT

CONCEPTOS GENERALES: SINTAXIS Y VARIABLES



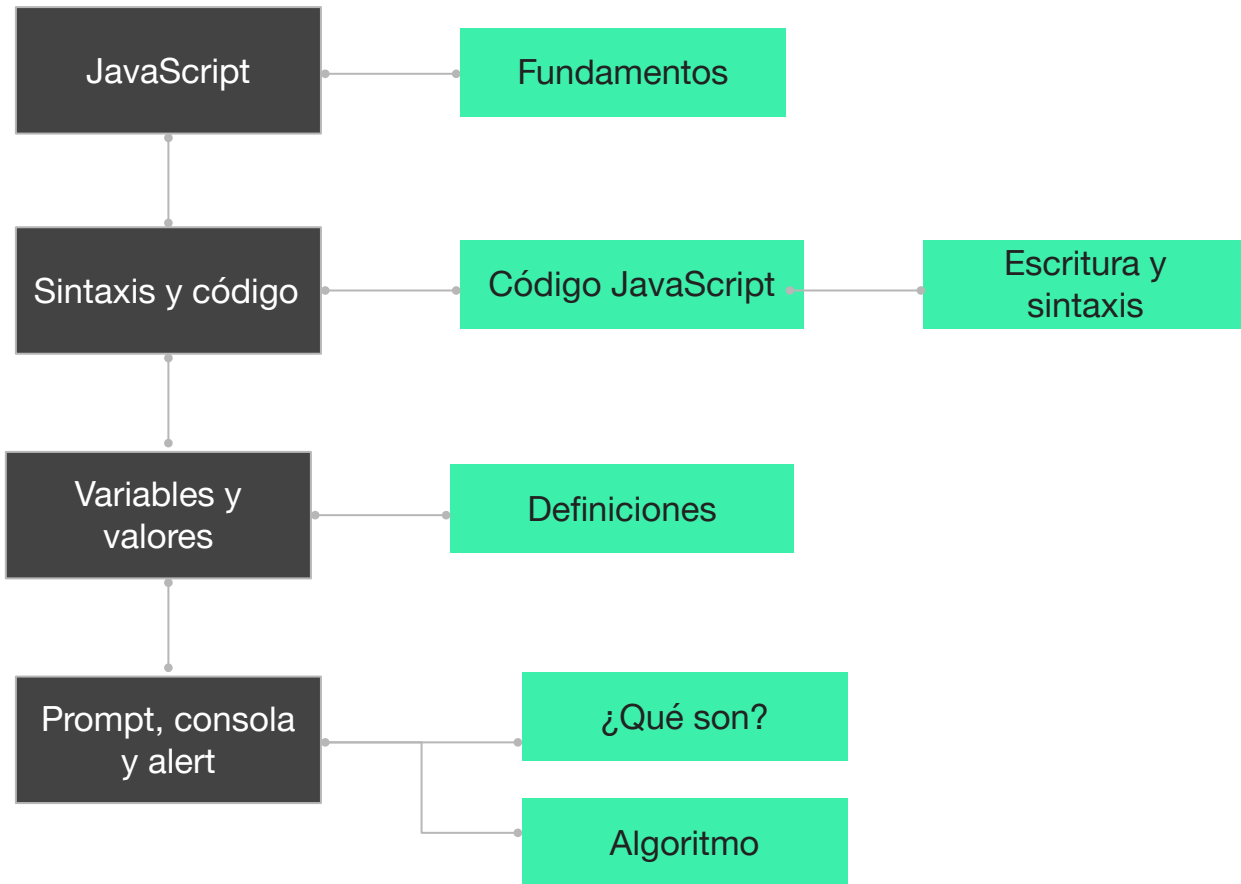
OBJETIVOS DE LA CLASE

- Reconocer qué es un algoritmo, elementos de entrada, salida y proceso.
- Aprender qué es una variable y cómo declararla.
- Indagar cómo asignar y cambiar el valor de una variable.

MAPA DE CONCEPTOS

MAPA DE CONCEPTOS CLASE 1

¡Para
recordar!



CRONOGRAMA DEL CURSO

Clase 1



**Conceptos generales:
sintaxis y variables**



EJEMPLO EN VIVO



CREAR UN ALGORITMO JS
SIMPLE

Clase 2



Control de flujos



EJEMPLOS EN VIVO



CREAR UN ALGORITMO CON
UN CONDICIONAL

Clase 3



**Funciones básicas +
Ciclos/Iteraciones**



EJEMPLOS EN VIVO



CREAR UN ALGORITMO
UTILIZANDO UN CICLO



HERRAMIENTAS DE LA CLASE

Les compartimos algunos recursos para acompañar la clase

- Guión de clase N° 1 [aquí](#).
- Quizz de clase N° 1 [aquí](#)
- Booklet de Javascript [aquí](#)
- FAQs de Javascript [aquí](#)

JAVASCRIPT: FUNDAMENTOS



¿QUÉ ES JAVASCRIPT?

JavaScript es un lenguaje de programación que se utiliza principalmente para aportar dinamismo a *sitios y aplicaciones web*.

Técnicamente, JavaScript es un lenguaje de programación interpretado por lo que el código escrito con JavaScript se puede probar directamente en cualquier navegador sin necesidad de procesos intermedios.

JavaScript funciona en complemento con los lenguajes web HTML Y CSS3.

APLICACIÓN WEB

¡Para
recordar!



Mientras que los sitios web buscan brindar información estática, las web apps permiten a los usuarios realizar múltiples tareas.

Las aplicaciones web son plataformas dinámicas e interactivas y sus funcionalidades están en constante mantenimiento y mejora.



APLICACIÓN WEB

¡Para
recordar!



Plataformas como MercadoLibre, Youtube, Gmail, Facebook, CoderHouse son web apps por la cantidad de funcionalidades que ofrecen



The screenshot shows the CoderHouse website with a dark background. The header includes the CoderHouse logo, navigation links for Cursos, Carreras, Comunidad, and ¿Por Qué Coder?, and a yellow button labeled 'Aplicá a la beca'. The main content area features large yellow and white text: 'Cursos y carreras online en vivo junto a expertos de la industria'. Below this, a paragraph describes the live courses and community access. At the bottom, there are two yellow buttons: 'Aplicá a la beca' and 'Ver cursos', and a small badge stating 'Calidad 100% asegurada'.

CODER HOUSE

Cursos ▾ Carreras ▾ Comunidad ▾ ¿Por Qué Coder?

Aplicá a la beca

Cursos y carreras online en vivo junto a expertos de la industria

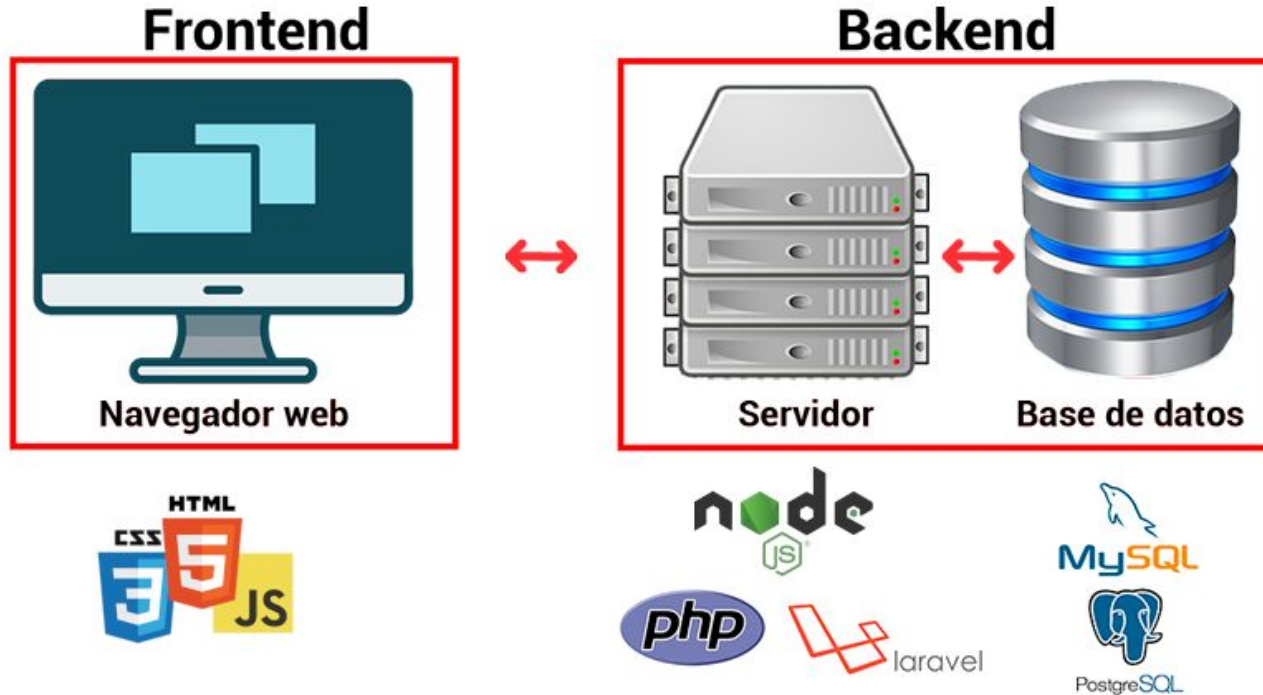
Nuestros cursos online en vivo combinan clases prácticas dictadas por expertos, mentorías personalizadas y acceso a una comunidad de +15,000 alumnos.
Accedé a educación de calidad con becas del 70%.

Aplicá a la beca [Ver cursos](#)

Calidad 100% asegurada

FRONT-END y BACK-END

¡Para
recordar!



En el curso nos focalizaremos en el desarrollo front-end de la aplicación web

CODER HOUSE

HERRAMIENTAS

¡Para
recordar!

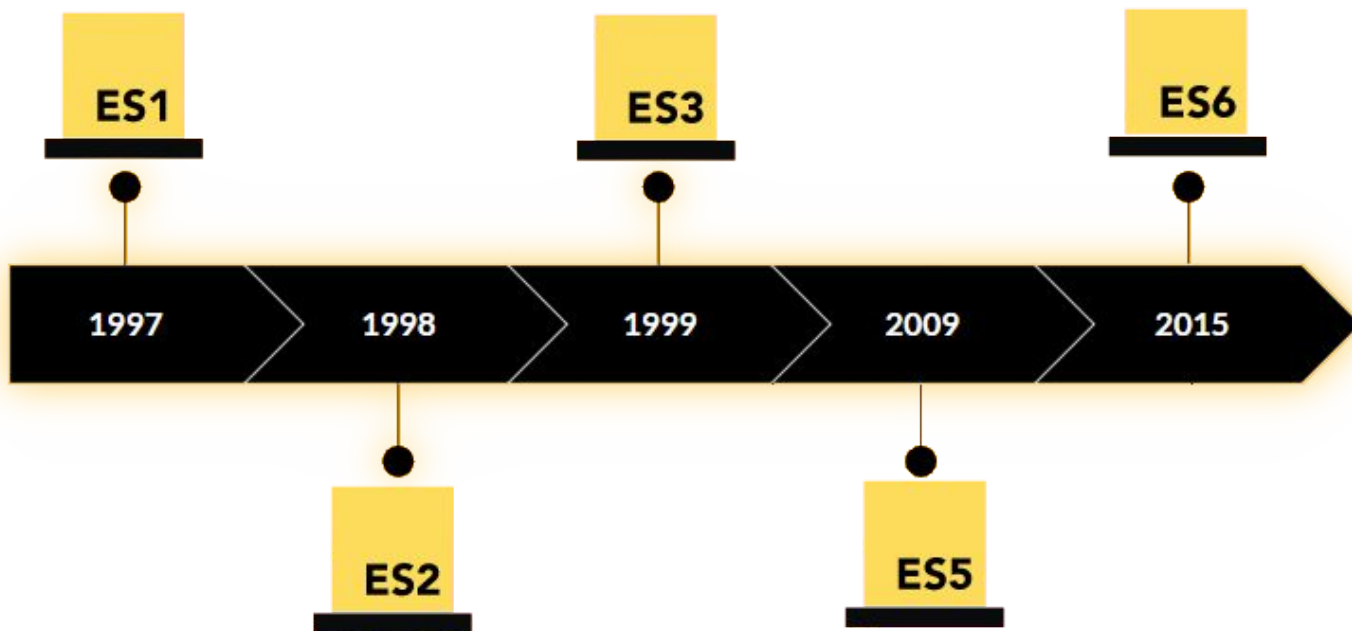


Durante el curso emplearemos algunas de las siguientes herramientas para desarrollar:

- Navegador Web: [Chrome](#), [Firefox](#), [Edge](#) o [Opera](#).
- Editor de Código Fuente: [Visual Studio Code](#), [Sublime Text 3](#) o [Atom](#).
- Framework CSS: [Bootstrap 4](#), [Bootstrap 5](#), [Bulma](#) o [Milligram](#).
- [jQuery](#).
- Servidor Local: [Live Server \(VS Code\)](#), [XAMPP](#) , [WampServer](#).

EVOLUCIÓN DE JAVASCRIPT

¡Para
recordar!



La primera versión de JavaScript ES1 se lanzó en 1997 y el lenguaje fue cambiando con el tiempo. En el curso nos focalizamos en las versiones ES5 y ES6

CODER HOUSE

SINTAXIS Y CÓDIGO

CÓDIGO JAVASCRIPT

JavaScript tiene sus propias reglas para la sintaxis, aunque respeta los estándares de muchos lenguajes de programación lógicos. Existen dos maneras de escribir código en JavaScript.

¿CÓMO ESCRIBIR CÓDIGO JS?

- Dentro de un archivo html, entre medio de las etiquetas **<script>**

Ejemplo:

```
<script>  
    // Aquí se escribe el código JS  
</script>
```

¿CÓMO ESCRIBIR CÓDIGO JS?

- En un archivo individual con **extensión .js**

Ejemplo: mi-archivo.js

Recuerda no utilizar espacios ni mayúsculas en los nombres de archivo.

```
<script src="js/main.js"></script>
```

SINTAXIS: REGLAS BÁSICAS

- No se tienen en cuenta los espacios en blanco y las nuevas líneas (al igual que HTML).
- Se distinguen las mayúsculas y minúsculas.
- Se pueden incluir bloques de comentarios:

```
<script>  
    // Comentario simple: una línea  
    /* Comentario de más de una línea I  
       Comentario de más de una línea II */  
</script>
```

SINTAXIS: PALABRAS RESERVADAS

- Palabras reservadas: son las palabras que se utilizan para construir las sentencias de JavaScript y que por tanto no pueden ser utilizadas libremente. Las palabras actualmente reservadas por JavaScript son:

`break, case, catch, continue, default, let
delete, do, else, finally, for, function, if, in,
instanceof, new, return, switch, this, throw, try,
typeof, var, void, while, with, etc`

VARIABLES Y VALORES

VARIABLES

Una variable es un espacio reservado en la memoria que, como su nombre indica, puede cambiar de contenido a lo largo de la ejecución de un programa. Podemos almacenar un número, un texto, un listado de números, etcétera.

```
<script>
  //Declaración de variable ES5.
  var nombreVariable1;

  //Declaración de variable ES6.
  let nombreVariable2;
  const LENGUAJE = "JAVASCRIPT";
</script>
```

VALORES

¡Para
recordar!



A una variable a la cual se le asigna
un valor al declarar se le dice
variable **inicializada**

TIPOS DE VALORES

En una variable podemos asignar distintos tipos de valores, ya sea un número, un texto, o resultados de operaciones entre ambos.

```
<script>  
  let variableNumerica;  
  var variableTexto;  
  
  variableNumerica = 5;  
  variableTexto = "Mi texto";  
  variableTexto = 'Mi texto';  
</script>
```

OPERACIONES BÁSICAS

Con variables de valores numéricos puedes realizar operaciones matemáticas: sumas, restas, multiplicaciones, etc.

```
<script>
  var    numeroA = 1;
  let    numeroB = 2;
  const  NUMEROC = 3;
  //Suma de dos números (1 + 2 = 3)
  let resultadoSuma = numeroA + numeroB;
  //Resta de dos números (2 - 1 = 1)
  let resultadoResta = numeroB - numeroA;
  //Producto de dos números (2 * 3 = 6)
  let resultadoProducto = numeroB * NUMEROC;
</script>
```

OPERACIONES BÁSICAS

Con variables de tipo string (texto) se puede concatenar los valores, es decir, combinarlas.

```
<script>
  var  textoA = "CODER";
  let  textoB = "HOUSE";
  const BLANCO = " ";
  //Concatenar textoA y textoB ("CODER" + "HOUSE" = "CODERHOUSE")
  let resultadoA = textoA + textoB;
  //Concatenar textoB y 1 ("HOUSE" + 1 = "HOUSE1")
  let resultadoB = textoB + 1;
  //Concatenar textoA, BLANCO y textoB ("CODER" + " " + "HOUSE" = "CODER HOUSE")
  let resultadoC = textoA + BLANCO + textoB;
</script>
```

Ejemplo
en vivo



¡VAMOS A PRACTICAR LO VISTO!



BREAK

¡5/10 MINUTOS Y VOLVEMOS!

PROMPT, CONSOLA Y ALERT

PROMPT

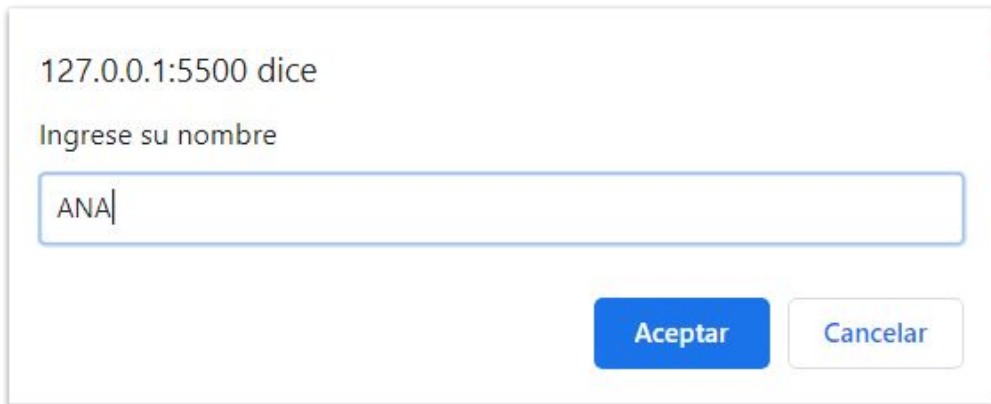
La sentencia `prompt()` mostrará un cuadro de diálogo para que el usuario ingrese un dato. Se puede proporcionar un mensaje que se colocará sobre el campo de texto. El valor que devuelve es una cadena que representa lo que el usuario ingresó.

```
<script>  
  let nombreIngresado = prompt("Ingrese su nombre");  
</script>
```

EJEMPLO DE PROMPT

En la pantalla del navegador, el usuario verá una ventana sobre la web que le solicitará un dato.

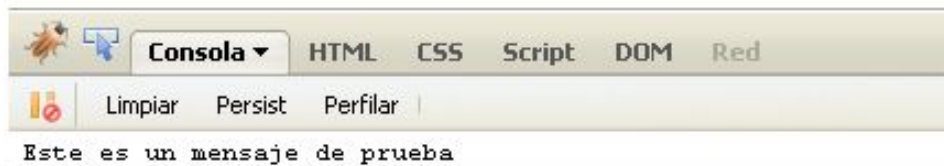
Al valor que el usuario ingresa se lo conoce por el término de ***entrada***.

A screenshot of a web browser window showing a form. At the top, it says "127.0.0.1:5500 dice". Below that is a label "Ingrese su nombre". There is a text input field containing the text "ANA". At the bottom right of the form are two buttons: "Aceptar" (blue) and "Cancelar" (light blue).

CONSOLA

La sentencia `console.log()` muestra el mensaje que pasemos como parámetro a la llamada en la consola JavaScript del Navegador web.

```
<script>  
    console.log("Mensaje de prueba");  
</script>
```



EJEMPLO DE CONSOLE.LOG

En Chrome, la consola del navegador está disponible accediendo mediante:

*Botón derecho sobre alguna
parte de la web > Inspeccionar
> Consola*

ALERT

La sentencia `alert()` mostrará una ventana sobre la página web que estemos accediendo mostrando el mensaje que se pase como parámetro a la llamada.

```
<script>  
  alert("¡Hola Coder!");  
</script>
```

EJEMPLO DE ALERT

En la pantalla del navegador, el usuario verá una ventana sobre la web que muestra un mensaje.

Al valor que mostramos al usuario como un resultado se lo conoce por el término de ***salida***.

127.0.0.1:5500 dice

¡Hola Coder!

Aceptar

ALGORITMO

En programación, un algoritmo es un **conjunto de procedimientos o funciones ordenados que se necesitan para realizar cierta operación o acción**. Por ejemplo, en una suma el algoritmo implica tomar un dato, sumarlo a otro y obtener un resultado.

EJEMPLO DE SCRIPT COMPLETO

Este es un ejemplo de un Script JS corriendo en un archivo HTML.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Mi primer App - CoderHouse</title>
    <script>
      let entrada = prompt("Ingresar una letra");
      let salida  = entrada + " " + "ingresada";
      alert(salida);
    </script>
  </head>
  <body>
    <h2>Esta página contiene una app</h2>
  </body>
</html>
```


Si ingreso "A"...



127.0.0.1:5500 dice

Ingresar una letra

Aceptar Cancelar

Obtengo...



127.0.0.1:5500 dice

A ingresada

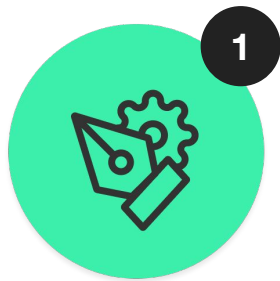
Aceptar

Ejemplo
en vivo



¡VAMOS A PRACTICAR LO VISTO!

CODER HOUSE



CREAR UN ALGORITMO JS SIMPLE

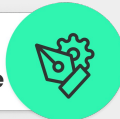
Crea un script en JS que le solicite al usuario ingresar datos y luego, mediante JavaScript, realiza operaciones sobre los mismos.

CREAR UN ALGORITMO JS SIMPLE

Formato: Código fuente en JavaScript. Debe identificar el apellido del alumno/a en el nombre de archivo por "claseApellido".

Sugerencia: Usamos prompt() para solicitar datos al usuario y console.log() o alert() para mostrar el resultado de las operaciones realizadas con esos datos. Si vas a sumar una entrada tene en cuenta que tenes que parsearla antes. Usando parseInt() o parseFloat()

Desafío
entregable



>> Consigna: Crea un script en JS que le solicite al usuario ingresar uno o más datos. Luego, con JavaScript, realiza operaciones matemáticas o de concatenación sobre las entradas teniendo en cuenta el tipo de dato. Al finalizar mostrar el resultados con alert() o console.log()

>>Aspectos a incluir en el entregable:

Archivo HTML con código JavaScript entre etiquetas <script></script>, que incluya la definición de un algoritmo.

CODER HOUSE

CREAR UN ALGORITMO JS SIMPLE

>>Ejemplo:

- Pedir nombre mediante prompt y mostrarlo en consola junto con algún texto de saludo.
Ejemplo: ¡Hola, Juan!
- Pedir un número mediante prompt, parsearlo, sumarlo a otro que se encuentre almacenado en una variable y luego mostrar el resultado en consola.
- Pedir un texto mediante prompt, luego otro, concatenarlos y mostrarlo en un alerta.

¿PREGUNTAS?





***TE INVITAMOS A QUE COMPLEMENTES
LA CLASE CON LOS SIGUIENTES
CODERTIPS***



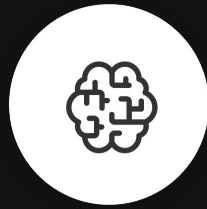
VIDEOS Y PODCASTS

- [Aprende Programación Web y construye el futuro de nuestra humanidad](#) | **Coderhouse**
- [Desarrollo freelance](#) | **Coderhouse**
- [Desarrollo profesional](#) | **Coderhouse**



VIDEOS Y PODCASTS

- [CoderNews](#) | **Coderhouse**
- [Serie de Branding](#) | **Coderhouse**
- [Serie para Emprendedores](#) | **Coderhouse**
- [Serie Aprende a Usar TikTok](#) | **Coderhouse**
- [Serie Finanzas Personales](#) | **Coderhouse**
- [CoderConf](#) | **Coderhouse**



¡PARA PENSAR!

¿Te gustaría comprobar tus conocimientos de la clase?

Te compartimos a través del chat de zoom
el enlace a un breve quiz de tarea.

Para el profesor:

- Acceder a la carpeta "Quizzes" de la camada
 - Ingresar al formulario de la clase
 - Pulsar el botón "Invitar"
 - Copiar el enlace
- Compartir el enlace a los alumnos a través del chat



RECURSOS:

- Consola, variables y tipos de datos |
Los apuntes de Majo (Página 1 a 8).
- Variables, valores y referencias |
Te lo explico con gatitos.
- Práctica interactiva sobre Algoritmia |
La aventura del punto.
- Herramienta recomendada |
Visual Studio Code.



***¿YA CONOCES LOS BENEFICIOS QUE TIENES
POR SER ESTUDIANTE DE CODERHOUSE?***

CODER HOUSE

BENEFICIOS

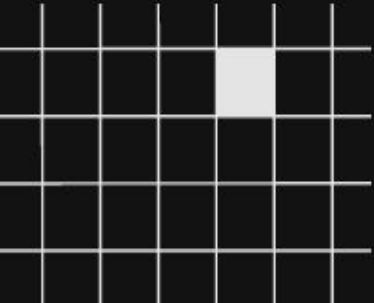


Haz clic [aquí](#) y conoce todos nuestros beneficios exclusivos para estudiantes de Coderhouse.



¡MUCHAS GRACIAS!

Resumen de lo visto en clase hoy:

- Fundamentos de desarrollo con JS.
 - ¿Cómo escribir JavaScript?
 - Declaración de Variables.
 - Funciones de prompt, alert y console.
- 



OPINA Y VALORA ESTA CLASE

#DEMOCRATIZANDO LA EDUCACIÓN

CODER HOUSE

CODER HOUSE

***¡GRACIAS POR ESTUDIAR CON
NOSOTROS!***
