

JavaScript

GUÍA DE ACTIVIDADES

hacia el proyecto final

¿Qué es?



La **Guía de Actividades** es un espacio que nuclea todas las actividades prácticas que se relacionan directamente con los temas del proyecto final, abordados en el curso y que luego se evaluarán en las **preentregas correspondientes**.

La Guía fue creada para que puedas afianzar, potenciar y poner en práctica los saberes adquiridos en clase. Se desarrolla de manera **asincrónica**. Sin embargo, su resolución es fundamental para la construcción del proyecto final.

¡A practicar! 😊

Nota: Te recomendamos que descargues el archivo para que lo puedas editar

Instancias prácticas



Actividades de clase

Ayudan a poner en práctica los conceptos y la teoría vista en clase. No deben ser subidas a la plataforma y se desarrollan en la clase sincrónica.

¿Dónde las encontraré?

En las clases identificadas correspondientemente.



Crear un algoritmo JS Simple

Crea un script en JS que le solicite al usuario ingresar mínimo 1 dato y luego, mediante JavaScript, realiza operaciones sobre los mismos.

Duración: 25/30 minutos

CODERHOUSE

Instancias prácticas

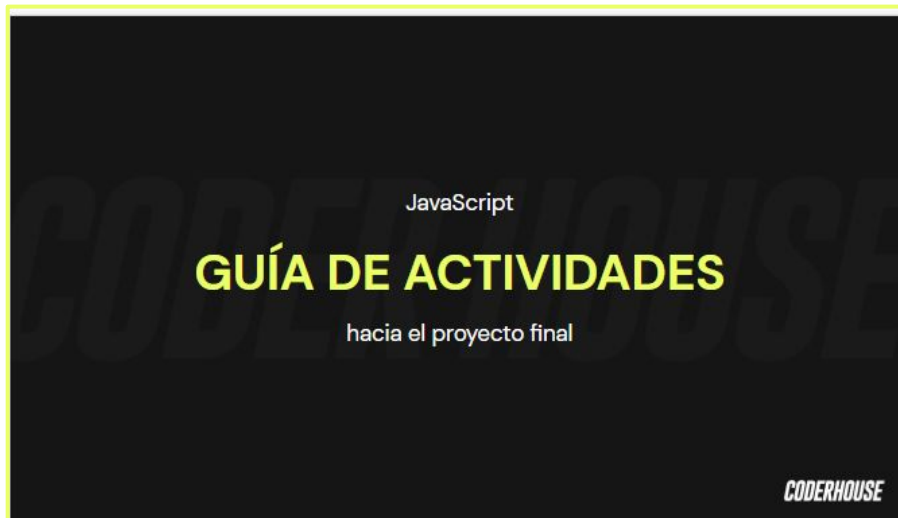


Actividades hacia tu Proyecto final

Actividades relacionadas con el Proyecto Final. Su resolución es muy importante para llegar con mayor nivel de avance a las preentregas. Se desarrollan de **forma asincrónica**.

¿Dónde las encontraré?

Dentro de la plataforma en la [Guía de Actividades](#) y linkeadas en cada clase que se corresponda a la actividad



Instancias prácticas



Pre-entregas

Entregas obligatorias 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 tutor/a.

¿Dónde la encontraré?

Dentro de la plataforma en la **Hoja de Ruta del curso** y linkeadas en cada clase que se corresponda.



Primera entrega de tu Proyecto final

Empieza a armar la estructura inicial de tu proyecto integrador.

CODERHOUSE

CODERHOUSE



GRILLA DE PRÁCTICAS Y PRE ENTREGAS

Clases	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Actividades Prácticas				🎯			🎯					🎯	🎯	🎯	🎯		🎯	
Entregas Proyecto Final					📊					📊				📊				📊

Proyecto Final

Alertas



CoderAlert

Son avisos creados para comunicar cuándo los temas de una clase están directamente relacionados con alguna actividad de la **Guía de Actividades** de modo que puedas ir construyendo con antelación parte de la consigna de cada **preentrega**.

¿Dónde la encontraré?

La conseguirás usualmente al final de la presentación de la clase.



#CoderAlert

Encontrarás en la [Guía de Actividades](#) del curso un ejercicio para aplicar todo lo aprendido hoy sobre **ciclos e iteraciones** a tu Proyecto. ¡Será fundamental al momento de realizar tu primera pre entrega en clase N°4!

CODERHOUSE

CODERHOUSE

ACTIVIDADES

CLASE 03

Ciclos e iteraciones



Crea un algoritmo utilizando un ciclo

Consigna.

- ✓ Tomando como base los ejemplos de la estructura for y while, crear un algoritmo que repita un bloque de instrucciones. En cada repetición es necesario efectuar una operación o comparación para obtener una salida por alerta o consola.

Aspectos a incluir

- ✓ Archivo HTML y Archivo JS, referenciado en el HTML por etiqueta `<script src="js/miarchivo.js"></script>`, que incluya la definición de un algoritmo en JavaScript que emplee bucles e instrucciones condicionales.

Podrás encontrar un ejemplo en la carpeta "Ejemplos de actividades"



Crea un algoritmo utilizando un ciclo

Sugerencias

- ✓ Usamos la instrucción **for** para repetir un número fijo de veces. Mientras que usamos **while** cuando queremos repetir algo hasta que se deje de cumplir una condición.

Ejemplo

- ✓ Pedir número mediante prompt y sumarle otro número en cada repetición, realizando una salida por cada resultado
- ✓ Pedir un texto mediante prompt, concatenar un valor en cada repetición, realizando una salida por cada resultado, hasta que se ingresa "ESC".
- ✓ Pedir un número por prompt, repetir la salida del mensaje "Hola" la cantidad de veces

CLASE 06

Arrays



Incorporar Arrays

Consigna

Traslada al proyecto integrador el concepto de objetos, visto en la clase de hoy. A partir de los ejemplos mostrados la primera clase, y en función del tipo de simulador que hayas elegido, deberás:

- ✓ Incorporar al menos un Array en tu proyecto.
- ✓ Utilizar algunos de los métodos o propiedades vistos en la clase.

Formato

- ✓ Página HTML y código fuente en JavaScript.

Podrás encontrar un ejemplo en la carpeta "Ejemplos de actividades"



Incorporar Arrays

Aspectos a incluir

- ✓ Archivo HTML y Archivo JS, referenciado en el HTML por etiqueta `<script src="js/miarchivo.js"></script>`, que incluya la definición de un algoritmo en JavaScript que emplee array para agrupar elementos similares.

Sugerencias

- ✓ Los Array cumplen el papel de listas en el programa. Principalmente, los usamos para agrupar elementos de un mismo tipo. Siempre que sea posible emplear los métodos disponibles para trabajar con ellos

Ejemplo

- ✓ Podemos crear arrays para los objetos identificados en el simulador de la clase anterior, Ejemplo: Array de Productos, Array de Personas, Array de Libros, Array de Autos, Array de Comidas, Array de Bebidas, Array de Tareas, etc.

Podrás encontrar un ejemplo en la carpeta "Ejemplos de actividades"

CLASE 10

Eventos



Interactuar con HTML

Consigna

Traslada al proyecto integrador el concepto de objetos, visto en la clase de hoy. En función del tipo de simulador que hayas elegido, deberás:

- ✓ Crear elementos manipulando el DOM a partir de la información de tus objetos.
- ✓ Modificar etiquetas existentes en función del resultado de operaciones.

Formato

- ✓ Página HTML y código fuente en JavaScript.

Podrás encontrar un ejemplo en la carpeta "Ejemplos de actividades"



Interactuar con HTML

Aspectos a incluir

- ✓ Archivo HTML y Archivo JS, referenciado en el HTML por etiqueta `<script src="js/miarchivo.js"></script>`, que incluya la definición de un algoritmo en JavaScript que opere sobre el DOM, modificando, agregando o eliminado elementos.

Sugerencias

- ✓ Generalmente, identificamos a un único elemento del DOM con el atributo `id` y a un conjunto asociado por `class`.

Ejemplo

- ✓ Podemos crear elementos HTML en función del listado de nuestros objetos identificados en la clase 6.
- ✓ Establecer un mensaje de bienvenida aleatorio usando un array de mensajes.
- ✓ Capturar una o más entradas por `prompt()` y mostrarlas en el HTML, modificando el DOM

Podrás encontrar un ejemplo en la carpeta "Ejemplos de actividades"

CLASE 12

Operadores avanzados I



Incorporar eventos

Consigna

- ✓ Con lo que vimos sobre DOM, ahora puedes sumarlo a tu proyecto, para interactuar entre los elementos HTML y JS. Es decir, asociar eventos que buscamos controlar sobre los elementos de la interfaz de nuestro simulador

Formato

- ✓ Página HTML y código fuente en JavaScript.

Podrás encontrar un ejemplo en la carpeta "Ejemplos de actividades"



Incorporar eventos

Aspectos a incluir

- ✓ Archivo HTML y Archivo JS, referenciado en el HTML por etiqueta `<script src="js/miarchivo.js"></script>`, que incluya la definición de un algoritmo en JavaScript que opere sobre el DOM manejando eventos.

Sugerencias

- ✓ Es posible asociar más de un evento a un elemento y se pueden emplear función comunes, anónimas y arrow para los manejadores de eventos.

Ejemplo

- ✓ Cuando el usuario completa algún dato, por ejemplo cantidad de cuotas, se captura ese dato y se agregan elementos al DOM mediante JS.
- ✓ Capturar la tecla ENTER para confirmar alguna acción.

Podrás encontrar un ejemplo en la carpeta "Ejemplos de actividades"

CLASE 13

Operadores Avanzados II + Workshop



Optimizando el proyecto final

Consigna

- ✓ Optimizarás tu proyecto final a través de la puesta en práctica de lo visto en esta clase según sea conveniente en cada caso.

Aspectos a incluir

- ✓ Operador Ternario / AND / OR. Busca estructuras condicionales simples en tu proyecto y simplificalas utilizando operador ternario u operadores lógicos AND y OR.
- ✓ Optimización. Con lo visto en clase, optimiza la asignación condicional de variables.
- ✓ Desestructuración. Aplica la desestructuración según corresponda para recuperar propiedades de objetos con claridad y rapidez.
- ✓ Spread. Usa el operador spread para replicar objetos o arrays o, también, para mejorar la lógica de tus funciones.

CLASE 14

Librerías



Incorporando librerías



Consigna

- ✓ Con lo visto en clase, incorpora al menos una librería a tu proyecto, brindándole un uso relevante. Si lo ves necesario, también es válida la opción de investigar por tu cuenta e integrar una librería diferente a las vistas en clase. Cualquiera sea tu elección de librería, justifica tu decisión.

Aspectos a incluir

- ✓ Incorporar una librería al proyecto de manera coherente, cuya aplicación se torne significativa para tu proyecto.
- ✓ Justificá tu elección según la naturaleza de tu proyecto y la utilidad que la librería seleccionada pueda tener en él.

Ejemplo

- ✓ Algunas pistas 🕶 Para lograr coherencia entre la librería seleccionada y el proyecto, deberás pensar en su aplicación real. Supongamos:
- ✓ Trabajo con fechas, en este caso lo más útil sería elegir librería Luxon 
- ✓ Trabajo con alert, entonces lo más coherente sería tomar Toastify o Sweet Alert 

Podrás encontrar un ejemplo en la carpeta "Ejemplos de actividades"

CLASE 16

Ajax & Fetch



Fetch en tu proyecto

Consigna

- ✓ Utiliza fetch() para cargar datos en tu aplicación de forma asincrónica.
- ✓ Puedes consumir una API que ofrezca recursos relevantes para tu app
- ✓ O bien,
- ✓ Crea un archivo .JSON y carga los datos de tu app usando fetch y una ruta relativa.

Aspectos a incluir

- ✓ Archivo HTML y Archivo JS, referenciado en el HTML por etiqueta `<script src="js/miarchivo.js"></script>`, que incluya la definición de un algoritmo en JavaScript que realice peticiones http usando fetch.

Ejemplo:

- ✓ Escribir los datos de tus productos en venta en un archivo .json y cargarlo en el inicio usando fetch()
- ✓ Si tengo una app sobre películas y series, armar mi catálogo consultando una API que envíe datos sobre ésto.
- ✓ Si trabajo con un formulario, al hacer submit puedo hacer una petición POST a alguna API de emails para enviar correos (ej, [Email JS](#))

Podrás encontrar un ejemplo en la carpeta "Ejemplos de actividades"