**Introducción a Programación Orientada a Objetos**

**👋 Introducción**

**¡Hola! 👋 Te damos la bienvenida a este nuevo paso.**

En los siguientes pasos aprenderás sobre la Programación Orientada a Objetos (POO). Aprenderás cómo diseñar y crear objetos, clases y métodos, permitiéndote escribir código más modular, reutilizable y estructurado.

Prepárate para descubrir cómo la Programación Orientada a Objetos puede transformar tu manera de pensar y abordar problemas en el mundo de la programación. ¡Vamos juntos a explorar este fascinante universo de posibilidades! 🚀

**📖 Material Descargable**

Aquí te dejamos acceso al material teórico pertinente que te acercará a los conceptos que pondrás en práctica en los próximos pasos.

👉[**Teoría Programación Orientada a Objetos**](https://drive.google.com/file/d/1XwTlxcTfXcrPem_IwB_UVhAmqXCtL9ys/view?usp=drive_link)

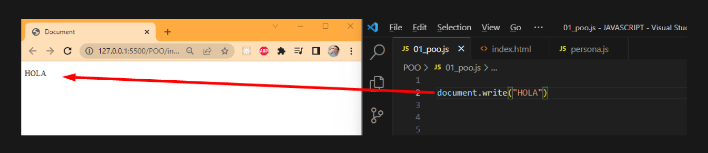
**⏭️ En el paso siguiente** tendrás ejercicios para practicar lo aprendido.

# Práctica: Iniciando con Objetos

Vamos a practicar con la creación de objetos a través de los siguientes ejercicios que proponemos a continuación.

❗ Importante: Para poder llevar a cabo los ejercicios tendrás que apoyarte en la teoría.

**💡 TIP : Para visualizar lo que estamos haciendo tenemos diferentes opciones. Pero una de ellas es poder crear un archivo index.html, asociar el archivo javascript y luego ver en el navegador de la consola lo que estamos haciendo. O utilizar document.write(“algo”), para verlo en el archivo.**



### ✏️ Actividad Crear un objeto "Persona"

1. Crea un objeto llamado "persona" que tenga las siguientes propiedades: nombre, edad, ciudad y profesión. Luego, agrega un método llamado "saludar".
2. Deberás instanciar al objeto y llamar al método saludar para que imprima en la consola el saludo utilizando los datos del objeto.

### ✏️ Actividad Crear un objeto "Libro"

1. Crea un objeto llamado "libro" que tenga las siguientes propiedades: título, autor, año de publicación y género. Luego, agrega un método llamado "información" que imprima en la consola la información del libro utilizando los datos del objeto.
2. Deberás instanciar al objeto y llamar al método información para que imprima en la consola los datos utilizando los atributos del objeto.

### ✏️ Actividad Getters y Setters

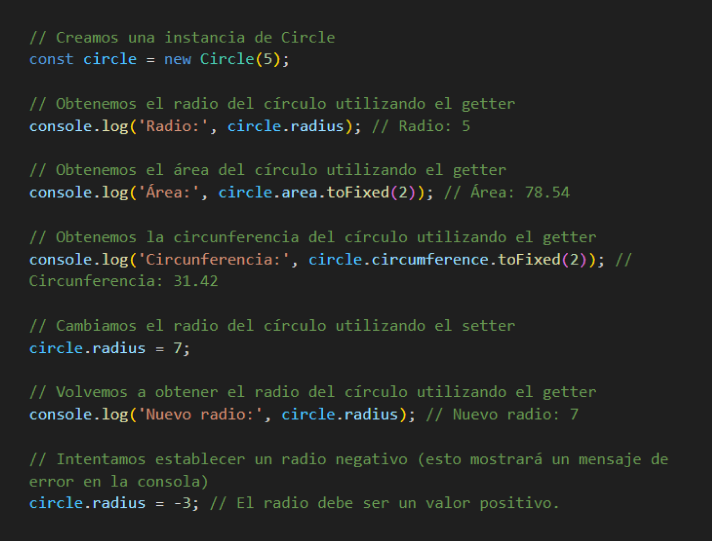
Crea una clase llamada **Círculo**que represente un círculo. La clase debe tener las siguientes características:

* Debe tener un constructor que acepte el radio del círculo como parámetro y lo almacene en una propiedad interna protegida.
* Implementa un getter llamado **radio**que retorne el valor del radio del círculo.
* Implementa un setter llamado **radio**que permita cambiar el radio del círculo. El setter debe validar que el nuevo radio sea un valor positivo y, en caso contrario, mostrar un mensaje de error en la consola.
* Implementa un getter llamado área que calcule y retorne el área del círculo.
* Implementa un getter llamado **circunferencia**que calcule y retorne la circunferencia del círculo.

Una vez que hayas definido la clase Circle con los getters y setters adecuados, crea una instancia de la clase, establece un radio inicial y muestra por consola el radio, el área y la circunferencia del círculo. Luego, cambia el radio utilizando el setter y vuelve a mostrar los resultados del radio, el área y la circunferencia con el nuevo valor

*💡Recuerda utilizar el prefijo "\_" para indicar que las propiedades del radio son protegidas y solo se pueden acceder y modificar a través de los getters y setters.*

Te dejamos una captura de pantalla de cómo deberían verse las pruebas que realizamos en Visual Studio Code



### ✏️ Actividad Empresa de celulares

Una empresa de tecnología desea llevar un registro de los diferentes modelos de celulares que ofrece. Para ello, han creado una clase llamada "celular" que representa las características de un teléfono móvil. La clase cuenta con los siguientes atributos:

* color: el color del celular.
* peso: el peso del celular en gramos.
* rdp: la resolución de la pantalla del celular en pulgadas.
* rdc: la resolución de la cámara del celular en megapíxeles.
* ram: la capacidad de memoria RAM del celular en gigabytes.
* encendido: un booleano que indica si el celular está encendido (true) o apagado (false).

Además, la clase tiene los siguientes métodos:

* **btnEncendido()**: si el celular está apagado, este método lo enciende y muestra una alerta indicando que se está encendiendo. Si ya está encendido, muestra una alerta informando que el celular ya está encendido.
* **btnReiniciar():** si el celular está encendido, este método lo reinicia y muestra una alerta indicando que se está reiniciando. Si el celular está apagado, muestra una alerta informando que no se puede reiniciar porque está apagado.
* **tomarFoto()**: este método muestra en pantalla un mensaje indicando que se ha tomado una foto con la resolución de la cámara del celular.
* **mostrarInfo():** este método devuelve un texto con las características principales del celular, como el color, el peso y la cantidad de memoria RAM.

**Tu tarea es crear tres instancias de la clase "celular" con sus respectivos datos y una vez creados los celulares, deberás realizar las siguientes acciones:**

a) Enciende el Celular.

b) Reinicia el Celular .

c) Toma una foto.

c) Muestra en pantalla la información del Celular.

Recuerda utilizar los métodos de la clase "celular" para realizar las acciones solicitadas.

### ✏️ Actividad Empresa de celulares 2

Tomando como referencia el ejercicio anterior crear una nueva clase para celulares de alta gama que **extienda**de la clase anterior y le agregue un atributo de una cámara más.

Tu tarea es crear dos instancias de la clase "**celularAltaGama**" con sus respectivos datos y una vez creados los celulares, deberás realizar las siguientes acciones:

a) Mostrar en pantalla la información de los celulares, agregando la cámara extra.

💡 *Pongan en común las diferentes formas de llegar al mismo resultado. ¿Hay alguna que les resulte óptima?*

**Cadena de Caracteres**

Continuamos practicando objetos. ¡Es muy importante que tengas esto claro para continuar con los temas que siguen!

¡Manos a la obra! 🚀

**✏️ Actividad Formato de título**

Escribe una función que tome una cadena como entrada y la convierta en formato de título, donde la primera letra de cada palabra esté en mayúscula y el resto en minúscula. Utiliza el método **split**() para dividir la cadena en palabras y **toUpperCase**() y **toLowerCase**() para formatear cada palabra.

💡*console.log(formatoTitulo("bienvenido a mi mundo"));****// Debe imprimir: Bienvenido A Mi Mundo***

**✏️ Actividad Palíndromos**

Escribe una función que tome una cadena como entrada y determine si es un palíndromo. Un palíndromo es una palabra o frase que se lee igual de izquierda a derecha que de derecha a izquierda, ignorando espacios y puntuación.

💡console.log(esPalindromo("Anita lava la tina."));**// Debe imprimir: true**

**✏️ Actividad Reemplazar palabras**

Escribe una función que tome una cadena y dos palabras como entradas, y reemplace todas las ocurrencias de la primera palabra con la segunda palabra en la cadena.

💡console.log(reemplazarPalabra("La casa es bonita, la casa es grande.", "casa", "hogar")); // Debe imprimir: "La hogar es bonita, la hogar es grande."

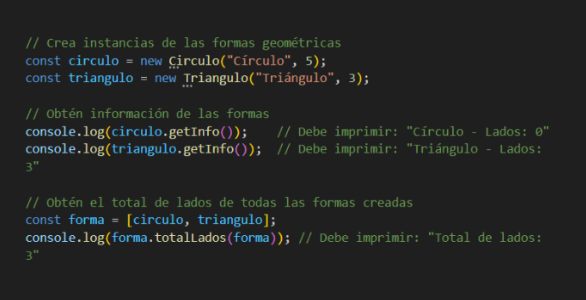
**✏️ Actividad Ejercicios Complementarios**

✨ Estos ejercicios son de tipo complementario. Esto quiere decir que te ayudará a avanzar en profundidad en el tema visto, pero no son obligatorios. Te recomendamos intentar con tu equipo trabajar al menos uno de ellos.

**Crear un sistema de formas geométricas**

* Crea una clase base llamada forma que tenga las siguientes propiedades y métodos:
  + Propiedades: nombre (cadena que representa el nombre de la forma) y lados (número que representa la cantidad de lados).
  + Método: obtenerInfo() que devuelve una cadena con el nombre y la cantidad de lados de la forma.
* Crea subclases para diferentes formas geométricas. Por ejemplo:
  + Circulo: Debe tener un radio como propiedad. El número de lados debe ser 0.
  + Triangulo: Debe tener una longitud de lado como propiedad. El número de lados debe ser 3.
* Agrega un método estático en la clase base forma llamado totalLados() que, dada una matriz de formas, calcule el total de lados de todas las formas y lo devuelva.
* Asegúrate de que el polimorfismo funcione implementando el método getInfo() en las subclases.

Ejemplo de uso:



**⏭️ En el paso siguiente** continuarás trabajando con objetos.