



Talleres POO básicos

campus[®]
{programmers land}





Nota: Los mensajes que retornan los métodos tiene que estar reflejados en el **HTML**.



Los datos solicitados en las clases tiene que obtenerlos del **HTML**, crea formularios y con **CSS** dales estilos.

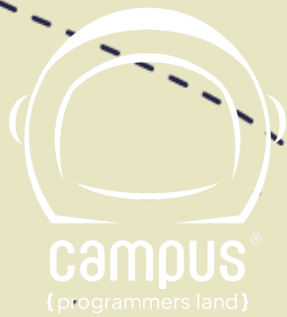




Crea una clase llamada **Persona** que tenga las siguientes propiedades: **nombre**, **edad**, **sexo** y un método llamado **saludar** que imprima en la consola un saludo con el nombre de la persona.

Crea una **instancia** de la clase **Persona** llamada **persona1** y asigna valores a sus propiedades. Luego, llama al método **saludar()** para que la persona imprima su saludo.





Crea una clase llamada **Estudiante** que **herede** de la clase **Persona**. Agrega una nueva propiedad llamada carrera y un método llamado **estudiar** que imprima en la consola un mensaje indicando la **carrera que está estudiando**.

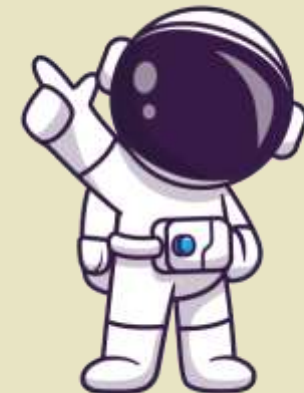
Crea una instancia de la clase **Estudiante** llamada **estudiante1** y asigna valores a sus propiedades. Luego, llama al método **saludar()** para que el estudiante imprima su saludo y al método **estudiar()** para que el estudiante imprima el mensaje de su carrera.





Agrega un método **estático** a la clase **Persona** llamado **esMayorDeEdad** que reciba la edad de una persona como parámetro y retorne true si la persona es mayor de edad (**18 años o más**), o **false en caso contrario**.

Llama al método estático **esMayorDeEdad()** pasando la edad de la **persona1** y muestra el resultado en la consola.





Crea una clase llamada **Animal** con las siguientes propiedades: **nombre y edad**. La clase debe tener un método llamado **hacerSonido** que imprima en la consola el sonido del animal.

Crea una instancia de la clase Animal llamada **animal1** y asigna valores a sus propiedades. Luego, llama al método **hacerSonido()** para que el animal imprima su sonido.



Crea una clase llamada **Perro** que **herede** de la clase **Animal**. Agrega una nueva propiedad llamada **raza** y un método llamado **moverCola** que imprima en la consola un mensaje indicando que el perro está moviendo la cola.





Crea una clase llamada **Figura** con las siguientes propiedades: **color** y **area**. La clase debe tener un método llamado **calcularArea()** que retorne el área de la figura.

• Crea una instancia de la clase **Figura** llamada **figura1** y asigna valores a sus propiedades. Luego, llama al método **calcularArea()** y muestra el resultado en el **HTML**.

Crea una clase llamada **Circulo** que herede de la clase **Figura**. Agrega una nueva propiedad llamada **radio** y redefine el método **calcularArea()** para que calcule el área del círculo.





Crea una instancia de la clase **Circulo** llamada **circulo1** y asigna valores a sus propiedades. Luego, llama al método **calcularArea()** y muestra el resultado en el **HTML**.

• Crea una clase llamada **Rectangulo** que herede de la clase **Figura**. Agrega nuevas propiedades llamadas **largo** y **ancho** y redefine el método **calcularArea()** para que calcule el área del rectángulo.



Crea una instancia de la clase **Rectangulo** llamada **rectangulo1** y asigna valores a sus propiedades. Luego, llama al método **calcularArea()** y muestra el resultado en el **HTML**.





Crea una clase llamada **Vehiculo** con las siguientes propiedades: **marca, modelo y velocidad**. La clase debe tener un método llamado **acelerar()** que incremente la velocidad del vehículo en **10 unidades**.

Crea una instancia de la clase **Vehiculo** llamada **vehiculo1** y asigna valores a sus propiedades. Luego, llama al método **acelerar()** y muestra la velocidad actual en el **HTML**.

Crea una clase llamada **Coche** que **herede** de la clase **Vehiculo**. Agrega una nueva propiedad llamada **combustible** y redefine el método **acelerar()** para que incremente la velocidad del coche en **20 unidades**.





Crea una instancia de la clase **Coche** llamada **coche1** y asigna valores a sus propiedades. Luego, llama al método **acelerar()** y muestra la velocidad actual en el **HTML**.



Agrega un método **estático** a la clase **Vehiculo** llamado **convertirKmHEnMph()** que reciba una velocidad en kilómetros por hora y retorne la velocidad equivalente en millas por hora. Para ello, divide la velocidad en kilómetros por hora entre 1.60934.



Llama al método estático **convertirKmHEnMph()** pasando una velocidad en kilómetros por hora y muestra el resultado en el **HTML**

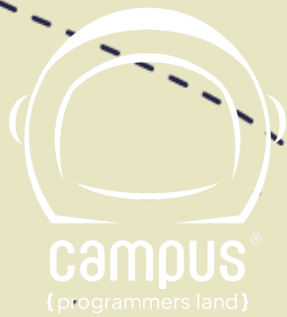


Crea una clase llamada **Empleado** con las siguientes propiedades: **nombre**, **edad** y **sueldo**. La clase debe tener un método llamado **calcularSalarioAnual()** que retorne el salario anual del empleado multiplicando el sueldo mensual por **12**.

- Crea una instancia de la clase **Empleado** llamada **empleado1** y asigna valores a sus propiedades. Luego, llama al método **calcularSalarioAnual()** y muestra el resultado en el **HTML**.

Crea una clase llamada **Gerente** que **herede** de la clase **Empleado**. Agrega una nueva propiedad llamada departamento y redefine el método **calcularSalarioAnual()** para que calcule el salario anual del gerente sumando un bono del **10%** al salario mensual antes de multiplicarlo por **12**.

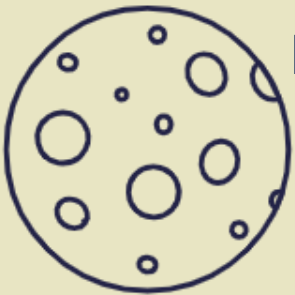




Crea una **instancia** de la clase **Gerente** llamada **gerente1** y asigna valores a sus propiedades. Luego, llama al método **calcularSalarioAnual()** y muestra el resultado en el **HTML**.

Agrega un método **estático** a la clase **Empleado** llamado **generarIdEmpleado()** que retorne un **ID** único para cada empleado. Puedes utilizar una variable **estática** para ir incrementando el **ID** cada vez que se cree una instancia de **Empleado**.

Llama al método estático **generarIdEmpleado()** y muestra el resultado en la consola.





¡Felicitaciones por completar los ejercicios de Programación Orientada a Objetos en JavaScript! Has dado un gran paso en tu camino como desarrollador/a. La POO es una habilidad fundamental en el mundo de la programación, y dominarla te abrirá las puertas a construir aplicaciones más estructuradas, reutilizables y eficientes.



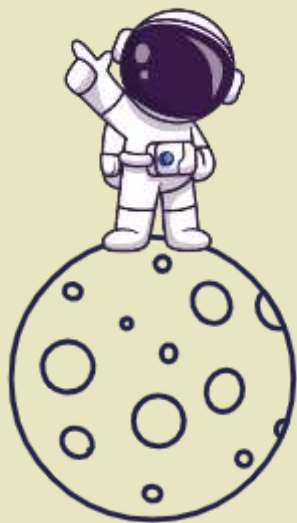
Recuerda que el aprendizaje es un proceso constante, y cada ejercicio que realizas te acerca más a convertirte en un/a programador/a hábil y versátil. A través de la POO, has aprendido a pensar en términos de objetos, propiedades y métodos, lo cual te permitirá abordar problemas de programación de manera más modular y comprensible.





No importa si encuentras dificultades en el camino, lo importante es que te enfrentaste a los desafíos y los superaste. Cada error es una oportunidad de aprendizaje y crecimiento. Continúa practicando, investigando y explorando nuevas posibilidades en el fascinante mundo de la POO.

Recuerda que la programación es un viaje emocionante y en constante evolución. Mantén tu pasión por aprender y experimentar, y verás cómo tus habilidades se fortalecen día a día. ¡Sigue adelante con determinación y confianza!



"Nunca te rindas, porque cada pequeño logro te acerca a tu gran meta. ¡Sigue programando con pasión y alcanzarás nuevos horizontes!"