

OBJECTS, CLASSES AND INHERITANCE

ERRORS

Los materiales relacionados son

Título	Descripción	Enlace
Introducción a objetos	Una pequeña introducción a qué son los objetos, cómo se definen y cómo se utilizan sus métodos y propiedades	https://www.aprendejavascript.dev/clase/objetos/introduccion
Iterar sobre objetos	Una de las prácticas más habituales es iterar sobre los objetos, es decir, sacar todos los valores que hay dentro.	https://www.aprendejavascript.dev/clase/objetos/iterar
Objetos y .	Otra breve introducción a los objetos que aporta algunos datos más de interés.	https://www.javascripttutorial.net/javascript-objects/
Introducción a clases	Introducción a las clases, mejor empezar desde “ES6 class declaration”	https://www.javascripttutorial.net/javascript-class/
Herencias	Sección en la que hablamos sobre herencias. Es muy importante entender todo el texto, ya que hay mucha información.	https://www.javascripttutorial.net/javascript-inheritance/
¿Composición o herencia?	Descripción sobre herencia y composición cuando se usan clases, y explicación de cuándo usar cada uno.	https://www.linkedin.com/pulse/herencia-y-composici%C3%B3n-cu%C3%A1l-elegir-robert-cabrera/
Errores	En este apartado explicaremos cómo gestionar errores en Javascript.	https://es.javascript.info/try-catch
Errores customizados	En este apartado se explica cómo customizar errores y cómo hacer herencia de los mismos, muy útil en temas como HTTP.	https://es.javascript.info/custom-errors
OPCIONAL: otras páginas sobre errores	Otras páginas que pueden completar o servir como explicación alternativa a la gestión de errores.	<ul style="list-style-type: none">- https://www.javascripttutorial.net/javascript-throw-exception/- https://www.javascripttutorial.net/javascript-try-catch-finally/

EJERCICIOS

- 1.1 Crea una jerarquía de clases teórica de tres niveles, la que quieras, sobre un caso que podría ser utilizado en un programa. Ejemplo: Animales -> Mamíferos -> Perros. Después, impleméntalo de manera práctica. Intenta llamar a las diferentes clases, atributos, métodos. Prueba a sustituir métodos, a extenderlos, etc.
- 1.2 Realiza todos los ejercicios que quieras de la página: <https://www.w3resource.com/javascript-exercises/javascript-object-exercises.php>
- 1.3 Realiza los siguientes ejercicios (<https://medium.com/@andrey.igorevich.borisov/10-javascript-exercises-with-objects-8942cc502754>):
- 1.4 De la página web: <https://github.com/Klod19/JS-Classes-Exercises> , haz los ejercicios que quieras. Recuerda que se trata de un repositorio, lee bien las instrucciones y comprueba muy bien los datos y los archivos que aparecen en la propia página.
- 1.5 Suponiendo que tienes esta clase (escrita a propósito en Python, la tienes que convertir a Javascript. ¿Cómo lo harías? ¿Son iguales los lenguajes? ¿Te vale cualquier cosa?):

```
class Vehicle:
    def __init__(self, name, max_speed, mileage):
        self.name = name
        self.max_speed = max_speed
        self.mileage = mileage

    def seating_capacity(self, capacity):
        return f"The seating capacity of a {self.name} is {capacity} passengers"
```

Después, crea una clase llamada Bus que hereda de Vehicle y que:

- Tenga como valor por defecto de la propiedad max_speed a 50.
 - También debe tener un método que sirva para imprimir (**display_stops**) todas las direcciones que tiene en su itinerario (mínimo 10 direcciones). ¿Cómo vas a gestionar la introducción de estos datos?
 - Si ahora quisiéramos instanciar varios objetos Bus con las direcciones de pamplona de la 16 y la 4, por ejemplo, ¿Cómo harías para meterle las direcciones? Recrea el ejemplo
- 1.6 Crea una clase *RomanNumberTransformer* para transformar números naturales a números romanos y viceversa (máximo que admite 1000, si no, da un error).

1.7 Crea una clase llamada *StringTransformer* para trabajar diferentes transformaciones de un *string* inicial con diferentes métodos. El constructor debe admitir un dato de tipo string y guardarlo como atributo, y no trabajar sobre él. Debe implementar diferentes métodos:

- **split_to_char**: Conversión de todo el string a array de caracteres uno por uno
- **random**: La ordenación de los caracteres de manera aleatoria
- **invert_char_order**: La inversión del orden de caracteres
- **without_vowels**: Quitar las vocales
- **without_consonants**: Quitar las consonantes.
- **to_word_array**: Array de palabras
- **invert_word_order**: Inversión del orden de las palabras del string

1.8 Crea una clase *ElementsSumZero* para encontrar todos los grupos de tres elementos dado un set de n números reales que sumen zero. El método se debería llamar **get_sum_zero_three_elements**. Ejemplo input-output:

```
Input      array      :      [-25,      -10,      -7,      -3,      2,      4,      8,      10]
Output : [[-10, 2, 8], [-7, -3, 10]]
```

1.9 Crea una clase llamada *Circle* que tenga como entrada el radio en el constructor. Debería tener dos métodos, uno para calcular el perímetro (**get_perimeter**) y otro para calcular el área (**get_area**). También debería poder calcular el volumen del cilindro que quedaría dada una altura (**get_volume_by_height**).

BIBLIOGRAFÍA

Aprende JavaScript - Curso de JavaScript desde cero y paso a paso. (s. f.-f).

<https://www.aprendejavascript.dev/>

El tutorial de JavaScript Moderno. (s. f.). <https://es.javascript.info/>

JavaScript tutorial. (2024b, junio 10). JavaScript Tutorial.

<https://www.javascripttutorial.net/>

Lara, R. C. (2020, 28 enero). *Herencia o Composición. ¿Cuál debo elegir?*

<https://www.linkedin.com/pulse/herencia-y-composici%C3%B3n-cu%C3%A1l-elegir-robert-cabrera/>