

1. Javascript es un lenguaje dinámico donde no existe el concepto de clase. Las características de los objetos son definidas a través de otros objetos llamados prototipos.
 - a. Investigue y exponga el modelo de objetos de Javascript.
 - b. Implemente en Python la clase `JavascriptObject` que tenga el mismo comportamiento que un objeto en Javascript.
 - c. ¿Al no existir clases en un lenguaje este no es orientado a objetos? Muestre qué características hacen o no a Javascript orientado a objetos.
2. En Python las variables y los parámetros de las funciones no tienen tipo.
 - a. Implemente un decorador `typeCheck` que reciba los tipos de los parámetros de la función a decorar y verifique en cada invocación que los objetos pasados como argumentos correspondan con dichos tipos.
 - b. Implemente una segunda variante del decorador que en vez de recibir los tipos como argumentos lea las *anotaciones* de los parámetros del método a decorar como si fuesen los tipos deseados.
 - c. Valore la utilidad de este decorador. ¿Sirve para algo hacer chequeo de tipos en un lenguaje dinámico?
 - d. ¿Tendría sentido modificar el lenguaje para tener variables y parámetros con tipos? ¿La definición del lenguaje obliga a la necesidad de variables y parámetros sin tipos?
3. Implemente en Python los siguientes patrones:
 - a. Singleton: Clase de la que solo se puede crear una instancia, por lo que todos los intentos de instanciación deben resultar en un mismo objeto (el del primer intento). Tenga en cuenta en su implementación que las clases que hereden de esta clase deberán heredar también el comportamiento Singleton
 - b. Implemente la clase `ObjetoInmutable` en python. Un objeto inmutable, una vez creado, no podrá ser modificado. No ser modificado significa que después de su creación no se puede hacer $o.t = valor$ ni $del(o.t)$, siendo o una instancia de `ObjetoInmutable`. Realice su diseño de forma tal que cualquier clase que herede de `ObjetoInmutable` tenga el mismo comportamiento.
4. Implemente en Python:
 - a. La clase `Matriz`, para representar matrices de enteros, con las operaciones de suma, producto, producto con un número. Implemente además otras funcionalidades que crea necesarias.
 - b. Implemente la indización para la clase `matriz` de forma tal que se puedan hacer construcciones como las siguientes: $a = matriz[0,6]$ o $matriz[1,2] = 9$.
 - c. Los objetos matrices deberán ser iterables. El iterador de una matriz con n filas y m columnas debe devolver los elementos en el siguiente orden:

$$m_{1,1}, m_{1,2}, \dots, m_{1,m}, m_{2,1}, \dots, m_{2,m}, \dots, m_{n,m}$$

- d. Los siguientes iterables, con y sin generadores:
 - i. Conjunto de Wirth
 - ii. `ConcatIterable`: iterable conformado por los elementos de dos iterables puestos a continuación.

- iii. WhereIterable: iterable conformado por aquellos elementos de un iterable que cumplen un determinado predicado.

Valore el mecanismo para indicar fin de iteración en Python.