

Indicar **Verdadero (V)** o **Falso(F)** las siguientes sentencias, según correspondan.

1. Python permite la Programación Orientada a Objetos (POO), el cual es un paradigma de programación también. Basado en esto los valores asignados a las variables como diccionarios, listas son llamadas también Objetos. **(VERDADERO)**

2. Para declarar o crear una clase se utiliza la palabra clave **class** seguida del nombre de la clase y dos puntos, de acuerdo a la siguiente sintaxis:

```
class <nombreclase>:  
    <atributos>  
    <métodos>
```

¿El nombre de la clase se puede escribir iniciando con minúscula sin ningún problema al momento de correr tu programa? **(VERDADERO)**

3. Para la definición de una clase el método **__init__** que es llamado como inicializador o constructor de una clase. A su vez este método especial se invoca cada vez que se instancia una clase, entonces siempre debe tener obligatoriamente como primer parámetro a **self**. **(VERDADERO)**

4. El **encapsulamiento** o **encapsulación** en programación es un concepto relacionado con la programación orientada a objetos, y hace referencia al ocultamiento de los estados internos de una clase al exterior. Dicho de otra manera, encapsular consiste en hacer que los atributos o métodos internos a una clase no se puedan acceder ni modificar desde fuera, sino que tan solo el propio objeto pueda acceder a ellos.

Para la siguiente declaración estoy protegiendo completamente el valor de mi estado o variable: **(VERDADERO)**

```
class Persona:  
    def __init__(self):  
        self._nombre = ""
```

5. Python es un lenguaje que debe utilizar la indentación o sangrado para delimitar los bloques de código en las clases o funciones, por lo que tenemos que verificar que las líneas que queremos que pertenezcan a la función estén correctamente indentadas y que generalmente son 4 espacios. **(VERDADERO)**

```
def observer (...):  
    print("Hi everyone!")
```

6. En la siguiente función, al hacer el llamado del mismo y ejecutar el programa. Este no tendrá ningún tipo de error o excepción (Leer cuidadosamente el fragmento de este código). **(FALSO)**

```
def multiplicador():  
    return 4 + spam*5  
print(multiplicador())
```

7. El método constructor en Python se utiliza para inicializar una instancia de una clase. Cuando se crea una instancia de una clase, el método `_init_` es llamado automáticamente por el intérprete de Python. Sino se escribe este método al crear una clase, una instancia de la clase debería funcionar sin inconvenientes. **(VERDADERO)**
8. Los atributos puede ser de diferentes tipos: enteros, flotantes, cadenas, listas, diccionarios, etc. Además, los atributos pueden tener accesibilidad dependiendo de su tipo de encapsulamiento. Entonces, los atributos son públicos en Python, lo que significa que puede accederse a ellos desde cualquier parte de la clase. **(VERDADERO)**