PRÀCTICA 1-UF1-NF1

OBJECTIU DE LA PRÀCTICA - CARACTERÍSTIQUES DEL LLENGUATGE PYTHON

Aquest treball es realitzarà per tal de buscar informació i posar exemples de les característiques del llenguatge de programació Python.

A la Wikipedia podem trobar la següent entrada relatva a Python (programming language):

> "...
>
> Python is an interpreted high-level general-purpose programming language. .....
>
> Python is dynamically-typed and garbage-collected. It supports multple programming paradigms, including structured (partcularly, procedural), object-oriented and functonal programming.
>
> ..."

EXERCICI 1. EXPLICACIÓ DE CONCEPTES

Cal cercar informació de dos dels conceptes que es mencionen a l'entrada de la wikipedia i explicar-los, si a l'explicació s'inclouen diagrames millor. Els conceptes a recercar són:

1. dynamically-typed
2. garbage-collected

EXERCICI 2. EXPLICACIÓ I EXEMPLES DE PROGRAMACIÓ

Posar una breu explicació dels diferents paradigmes de programació que ofereix Python i posar exemples (trossos de codi) de cada un d'ells:

1. structured (partcularly, procedural) programming
2. object-oriented programming
3. functonal programming.

LLIURAMENT

S'entregarà al Moodle en format PDF i en la data indicada a la tasca.

En el document caldrà definir una capçalera a on apareguin les següents dades: Nom complet de l'alumne, nom del Mòdul, Unitats formatva i Nucli formatiu.

El nom del document haurà de ser "M5_UF1_NF1_P1_Cognom1_Nom.pdf".

AVALUACIÓ

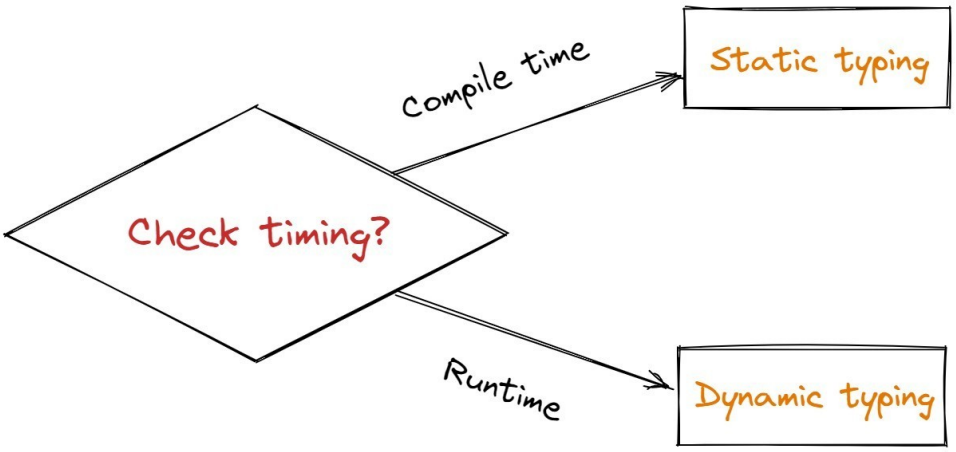
S'avaluarà:

- El contingut de les respostes
- El disseny del document tenint en compte la facilitat de lectura.
- L'aspecte general i la correcta utilització del llenguatge.

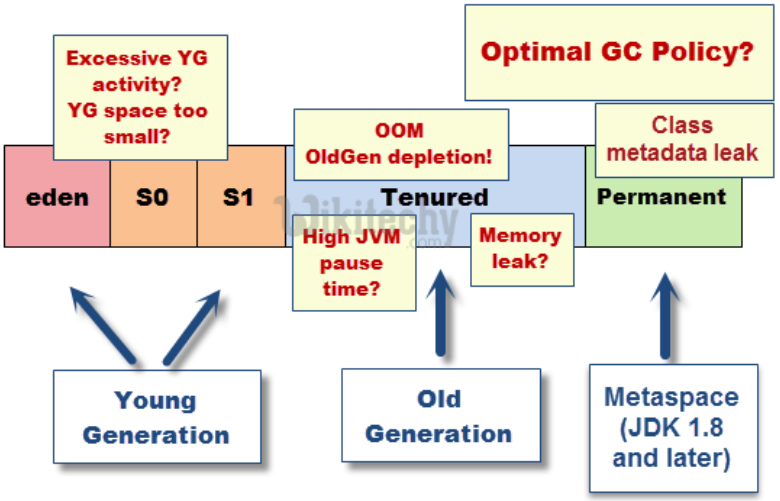
Ex1

Dynamically-Typed:

Los lenguajes de tipado dinámico son aquellos donde el interprete realiza la comprobación de tipificado durante la ejecución del programa. Python pertenece a este tipo de lenguaje dinámico a causa de su capacidad comenzar teniendo un tipo de dato y cambiar en cualquier momento a otro tipo diferente.



Garbage-Collected: Hace referencia a un mecanismo automático que gestiona la memoria mediante la eliminación de objetos no utilizados, evitando que los desarrolladores tengan que controlar y realizar una asignación o liberación manual de memoria. De esta forma evitamos de forma automática un uso de espacio elevado en el dispositivo producido por los diferentes programas.

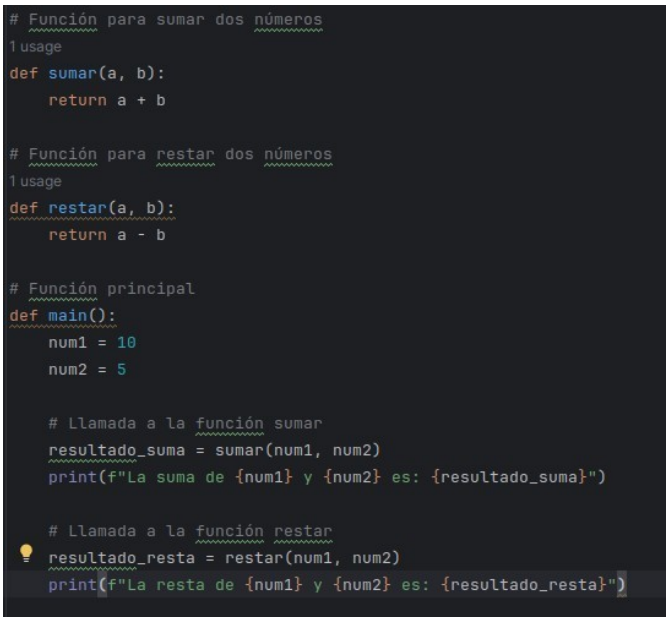


Diferentes tipos de GC en Java.

Ex2

structured (partcularly, procedural) programming: La programación estructurada se centra en la organización del código en procedimientos o funciones de forma definida, haciendo referencia al control lineal y la ausencia de saltos incontrolados (uso de tabulaciones).

Se puede implementar la programación procedural enfocándose en la creación de funciones y establecer una estructura ordenada.



```python
# Función para sumar dos números
1 usage
def sumar(a, b):
    return a + b

# Función para restar dos números
1 usage
def restar(a, b):
    return a - b

# Función principal
def main():
    num1 = 10
    num2 = 5

    # Llamada a la función sumar
    resultado_suma = sumar(num1, num2)
    print(f"La suma de {num1} y {num2} es: {resultado_suma}")

    # Llamada a la función restar
    resultado_resta = restar(num1, num2)
    print(f"La resta de {num1} y {num2} es: {resultado_resta}")
```

object-oriented programming:

La programación orientada a objetos es un paradigma que se basa en el concepto "objeto". Estos objetos son entidades que combinan datos (atributos) y funciones (métodos) para su funcionamiento.

La OOP se centra en la organización del código de software alrededor de estos objetos, permitiendo la reutilización, la modularidad y el Mantenimiento del código.

Los objetos interactúan entre sí a través de mensajes, encapsulando datos y comportamientos en un solo lugar.
Los conceptos fundamentales de la OOP incluyen:

-Abstracción: Representar conceptos del mundo real en un modelo computacional.

-Encapsulamiento: Ocultar detalles internos de un objeto y exponer solo lo que es necesario.

-Herencia: Permite la creación de nuevas clases basadas en clases ya existentes, heredando sus atributos.

-Polimorfismo: Es la capacidad de que objetos diferentes sean tratados de manera uniforme.

Mascotas.py

```python
#! usr/bin/python

#CREADO POR PYTHON DIARIO (www.pythondiario.com)

class Mascota(object):

    def __init__(self, nombre, especie):
        self.nombre = nombre
        self.especie = especie

    def darNombre(self):
        return self.nombre

    def darEspecie(self):
        return self.especie

    def __str__(self):
        return "%s es un %s" % (self.nombre, self.especie)
```

Line 22, Column 1 | Tab Size: 4 | Python

functional programming: La programación funcional es un paradigma que utiliza expresiones y funciones sin cambiar el estado ni los datos. Es decir tratar de solucionar el problema como una función matemática.

La programación funcional pretende escribir un código más fácil de entender y más resistente a los errores con el uso de funciones puras. Esto se consigue evitando el uso de sentencias de control de flujo (for, while, break), las cuales dificultan el seguimiento del código.