Initiation à l'algorithmique et programmation

L3 2023-2024

Travaux Pratiques 6

Voici quelques exercices sur la Portée des Variables en Python :

Exercice 1. *Comprendre la Portée Locale*

```python
def add_five(x):
    y = 5
    return x + y

print(add_five(10))
print(y)
```

Question : Quel sera le résultat de ce code, et pourquoi ?

Exercice 2. *Variables Globales vs Locales*

```python
x = 10

def print_x():
    x = 5
    print(A l'interieur de la function :", x)

print_x()
print(A l'exterieur de la function :", x)
```

Question : Expliquez le résultat de ce code, en mettant l'accent sur la portée de la variable x.

Exercice 3. *Modification d'une Variable Globale dans une Fonction*

```python
z = 3

def modify_global():
    global z
    z = 0

print("Avant l'appel de la fonction :", z)
modify_global()
print("Apr s l'appel de la fonction :", z)
```

Question : Discutez de l'utilisation du mot-clé `global` dans ce contexte et de son effet sur la variable z.

`global` : *Le mot-clé global est utilisé pour modifier une variable définie dans la portée globale à l'intérieur d'une fonction. Sans l'utilisation du mot-clé global, la modification dans la fonction aurait créé une nouvelle variable locale* ***z*** *sans affecter la variable globale* ***z****.*

Exercice 4. *Fonctions Imbriquées et Portée*

```python
def outer_function():
    a = 20
    def inner_function():
        a = 30
        print("Inner a :", a)
    inner_function()
    print("Outer a :", a)

a = 10
outer_function()
print("Global a :", a)
```

Question : Expliquez la portée de la variable a dans chacune de ses occurrences.

Exercice 5. *Masquage de Variable*

```python
a = 1

def outer():
    a = 2
    def inner():
        a = 3
        print("Inner a :", a)
    inner()
    print("Outer a :", a)

outer()
print("Global a :", a)
```

Question : Illustrez le concept de masquage de variable dans ce code.

Masquage de variable : *également connu sous le nom de "shadowing" en anglais, est un concept important en programmation, y compris en Python. Il se produit lorsqu'une variable dans une portée interne (comme à l'intérieur d'une fonction) a le même nom qu'une variable dans une portée externe. Dans ce cas, la variable interne "masque" la variable externe, c'est-à-dire qu'elle cache ou éclipse la variable externe de même nom. Voici quelques points clés à comprendre sur le masquage de variable :*

- *Portée Locale Prioritaire : Lorsqu'une variable interne et une variable externe portent le même nom, toute référence à ce nom dans la portée interne fait référence à la variable interne.*
- *Indépendance des Variables : La variable interne et la variable externe sont indépendantes l'une de l'autre ; la modification de l'une n'affecte pas l'autre.*

Exercice 6. *UnboundLocalError*

```python
counter = 0

def increment():
    counter += 1
    return counter

print(increment())
```

Question : Pourquoi ce code génère-t-il une `UnboundLocalError` et comment peut-on le corriger ?