

Week3_Exercice1_Corrigé

October 15, 2019

1 Algorithmique et Pensée Computationnelle

2 Algorithmique et Pensée Computationnelle

3 Types Scala

Contrairement à Python, Scala est un langage à typage statique. C'est-à-dire que le type d'une variable est donné à son initialisation.

Définissez une variable `note` égale à 4 qui est un integer.

Définissez une variable `note_maximale` à 5.5 qui est un float.

Définissez une variable `note_souhaite` à 5.25 à laquelle sera automatiquement assignée un type.

Affichez le type de la variable `note_souhaite`.

Modifiez la variable `note_souhaite` en integer.

Essayez de modifier la `note` à ce qu'elle soit égale à la 'note_maximale'.

```
[ ]: var note: Int = 4
    // Afficher note
    println(note)

var note_maximale: Float = 5.5f
// Afficher note_maximale et son type
println(note_maximale)
println("Type de note_maximale: ")
println(note_maximale.getClass)

var note_souhaite = 5.25
// Afficher note_souhaite et son type
println(note_souhaite)
println("Type de note_souhaite: ")
println(note_souhaite.getClass)

note_souhaite = note_souhaite.toInt
println(note_souhaite)
```

```
[ ]: note= note_maximale // ne marche pas
```

```
[ ]:
```

```
[ ]:
```

Week3_Exercice2_Corrigé

October 15, 2019

1 Algorithmique et Pensée Computationnelle

2 Algorithmique et Pensée Computationnelle

3 Introduction à Python

3.0.1 A. Fonction input

Demande input d'utilisateur

Completez le code afin que le contenu de `input_string` soit l'output.

```
[ ]: input_string = input("Quel est votre prénom? ")  
print("Bonjour, ", input_string)
```

3.0.2 Exercice:

Modifiez le code ci dessus pour également demander le nom de famille puis affichez `Bonjour, {prénom} {nom}`

```
[ ]: input_string = input("Quel est votre prénom? ")  
nom = input("Quel est votre nom? ")  
print("Bonjour, ", input_string, nom)
```

3.0.3 B. Variables

Définition de variable, type de variable, conversion de type, opérateurs arithmétiques, opérateurs booléens et opérateurs de comparaison.

Introduisez une variable de type string `nom_chien`, une variable de type integer `age_chien`.

Affichez "Mon chien nommé (`nom_chien`) et agé de (`age_chien`) ans." en utilisant la fonction `format()`.

Affichez "Mon chien nommé (`nom_chien`) et agé de (`age_chien`) ans." en utilisant le symbole `%`.

```
[ ]: nom_chien = "Lola"
      age_chien = 3

      print("Mon chien nommé {0} et agé de {1} ans.".format(nom_chien,age_chien))
      print("Mon chien nommé %s et agé de %d ans." % (nom_chien,age_chien))
```

3.0.4 Exercice:

Changez le code ci dessus pour demander à l'utilisateur d'entrer le nom de son chien

```
[ ]: nom_chien = input("Quel est le nom de votre chien? ")
      age_chien = 3

      print(type(nom_chien))
      print(type(age_chien))
```

3.0.5 Exercice:

Modifiez la valeur de la variable `age_chien` afin que le chiffre soit un float (nombre à virgule).

Assurez vous que la variable `age_chien` soit un float en affichant le type de la variable.

Affichez la valeur de la variable `age_chien` en float et en integer.

Vous pouvez vérifier le type de chaque variable dans la cellule ci dessus

```
[ ]: age_chien = 3.75
      print(type(age_chien))
      print(float(age_chien))
      print(int(age_chien))
```

3.0.6 Exercice:

Créez une variable `poids_chien`, `resultat`, `remainder`.

Divisez la valeur de la variable `poids_chien` par deux et enregistrez le résultat dans la variable `resultat`.

Trouvez le reste de la division grâce à la fonction modulo `%` et enregistrez le résultat dans la variable `remainder`.

```
[ ]: from assertion import assert_division

      # VOTRE CODE ICI
      poids_chien = 11
      poids_ideal=5
      resultat = poids_chien/2
```

```
remainder = poids_chien%2
# FIN DE VOTRE CODE

assert_division(poids_chien, resultat, remainder)
```

Diminuez la valeur de la variable `poids_ideal` de 2.

Diminuez la valeur de la variable `poids_chien` de 1 .

```
[ ]: from assertion import State, assert_sub
prev = State("sub", poids_chien=poids_chien, poids_ideal=poids_ideal)

# VOTRE CODE ICI
poids_ideal -=2
poids_chien -=1
# FIN DE VOTRE CODE

assert_sub(prev, State(poids_chien=poids_chien, poids_ideal=poids_ideal))
```

3.0.7 Exercice:

Vérifiez si la valeur de la variable `poids_chien` est égale à celle de la variable `poids_ideal` en utilisant un opérateur qui donne une réponse booléenne `True` ou `False`. Stockez la réponse booléenne dans la variable `egalite_poids`.

Affichez la valeur de la variable `egalite_poids`.

```
[ ]: from assertion import assert_equal

# VOTRE CODE ICI
egalite_poids = poids_chien==poids_ideal
print(egalite_poids)
# FIN DE VOTRE CODE

assert_equal(poids_chien, poids_ideal, egalite_poids)
```

3.0.8 Exercice:

Vérifiez que la valeur de la variable `poids_chien` est plus petite ou égale que celle de la variable `poids_ideal`. Stockez la réponse booléenne dans la variable `comparaison_poids`.

Affichez la valeur de la variable `comparaison_poids`.

Vérifiez que la valeur de la variable `poids_ideal` est plus grande que celle de la variable `poids_chien` en une seule ligne grâce à la fonction `print`.

```
[ ]: from assertion import assert_comparaison

# VOTRE CODE ICI
comparaison_poids = poids_chien <= poids_ideal
print(comparaison_poids)
print(poids_ideal > poids_chien)
# FIN DE VOTRE CODE

assert_comparaison(poids_chien, poids_ideal, comparaison_poids)
```

3.0.9 C. Strings

Index, slicing, in, length

Définissez une nouvelle variable du nom `python3` de type `string` contenant "Hard but cool!" ou un autre texte de votre choix (d'au moins 13 caractères).

Définissez la valeur de la variable `premiere_lettre` comme le premier caractère de `python3` en utilisant l'indexation.

Définissez la valeur de la variable `derniere_lettre` comme le dernier caractère de `python3` en utilisant l'indexation.

Affichez la valeur de `premiere_lettre` et `derniere_lettre` sur deux lignes distinctes.

```
[ ]: from assertion import assert_strings

# VOTRE CODE ICI
python3 = "Hard but cool!"
premiere_lettre = python3[0]
derniere_lettre = python3[-1]
print(premiere_lettre)
print(derniere_lettre)
# FIN DE VOTRE CODE

assert_strings(python3, premiere_lettre, derniere_lettre)
```

3.0.10 Exercice:

Définissez une variable `quatre` qui reprend les 4 premiers caractères de `python3` en utilisant le slicing `str[ind1:ind2]`.

Définissez une variable `neuf_treize` qui reprend les caractères 9 à 13 de `python3` en utilisant le slicing `str[ind1:ind2]`.

Définissez une variable `cinq_huit` qui reprend le "but" de `python3` en utilisant le slicing `str[ind1:ind2]`.

Affichez les valeurs de `quatre`, `neuf_treize` et `cinq_huit` sur une seule ligne.

```
[ ]: from assertion import assert_slicing

# VOTRE CODE ICI
quatre = python3[0:4]
neuf_treize = python3[9:13]
cinq_huit = python3[5:8]
print(quatre, neuf_treize, cinq_huit)
# FIN DE VOTRE CODE

assert_slicing(python3, quatre,neuf_treize,cinq_huit)
```

Vérifiez et affichez que le mot “joke” se trouve dans `python3` grâce à l’opérateur `in`.

Vérifiez et affichez que le mot “but” se trouve dans `python3` grâce à l’opérateur `in`.

```
[ ]: print("joke" in python3)
print("but" in python3)
```

3.0.11 Exercice:

Définissez `first_half` comme étant la première moitié de caractères stocké dans `python3` en utilisant la fonction `len()`.

Affichez `first_half`.

```
[ ]: from assertion import assert_half

# VOTRE CODE ICI
first_half = python3[0:int(len(python3)/2)]
# FIN DE VOTRE CODE

print(first_half)

assert_half(python3, first_half)
```

3.0.12 Exercice:

Définissez et affichez `dont_worry` comme “Don’t worry, you’re gonna become a king!”

Affichez “It’s just like Rock’n’Roll!” sans créer de nouvelle variable.

Affichez `dont_worry` en majuscule et minuscule sur deux lignes.

```
[ ]: from assertion import assert_dont_worry

# VOTRE CODE ICI
```

```

dont_worry = "Don't worry, you're gonna become a king!"
print( "It's just like Rock'n'Roll!")
upper = dont_worry.upper()
lower = dont_worry.lower()
# FIN DE VOTRE CODE

assert_dont_worry(dont_worry, upper, lower)

```

3.0.13 D. Conditions

Opérateurs booléens, If, else

Définissez une variable string `name`, une variable integer `age`, une variable integer `brothers` et une variable `sisters`.

Vérifiez et affichez que `name` est égal à la valeur de `name` OU que `age` est égal à la valeur de `age` en une seule ligne. Faites la même chose avec AND NOT.

Vérifiez et affichez que `name` est égal à “Miro” OU PAS que `age` est supérieur à 5 en une seule ligne.

Affichez l’ordre de priorité des opérateurs booléens.

Affichez “La condition est vraie, donc le code peut être exécuter!” si `age` est supérieur à 5 ET longueur de `name` inférieur à 10. Utilisez if et l’indentation.

Affichez “I’m not alone” si `brothers` OU `sisters` supérieur à 1. Affichez “We have boys at home” si `brothers` supérieur à 1 ET `sisters` égal à 0. Affichez “We have girls at home” si `brothers` égal à 0 ET `sisters` supérieur à 1. Sinon affichez “I have a little goat and a skateboard”.

```

[ ]: name="Paulina"
age=9
brothers=2
sisters=0
print(name=="Paulina" or age==9)
print(name=="Paulina" and not age==9)
print(name=="Miro" or not age>5)
print("Not, And, Or")
if age>5 and len(name)<10:
    print("La condition est vraie, donc le code peut être exécuter!")

if brothers>1 or sisters>1:
    print("I'm not alone")
elif brothers>1 and sisters==0:
    print("We have boys at home")
elif brothers==0 and sisters>1:
    print("We have girls at home")
else:
    print("I have a little goat and a skateboard")

```


3.0.14 F. Fonctions

Définition, appel, return, paramètre par défaut

Définissez une fonction nommée `vin` qui affiche “Neuchâtel?” une fois que la fonction est appelée.

Appelez la fonction `vin`.

Pour chaque `i` de rang 3, appelez la fonction `vin`.

```
[ ]: def vin():  
    print("Neuchâtel?")  
    vin()  
    for i in range(3):  
        vin()
```

Définissez une fonction `multip` ayant deux paramètres `a` et `b`. La fonction retourne le résultat de la multiplication `a*b`.

Créez un variable `result` qui appelle la fonction `multip` de 3 et 7.

Affichez `result`.

```
[ ]: def multip(a,b):  
    return a*b  
result=multip(3,7)  
print(result)
```

```
[ ]:
```

Algorithmique et Pensée Computationnelle

Algorithmique et Pensée Computationnelle

Compréhension de Python

Dans les prochains exercices, il vous sera demandé de comprendre le code qui vous sera montré, ainsi que d'anticiper ce qu'il va afficher

/!\ Ces questions sont tirées d'anciens examens!

1) Anticipez quel texte va être affiché dans la console ¶

In []:

```
x = True
y = False
z = False

if y or (x and z):
    print("output1")
if not x or (y and z):
    print("output2")
if (x or z) or (y and z):
    print("output3")
if (y != z):
    print("output4")
```

2) Anticipez quel texte va être affiché dans la console

In []:

```
age = 20
output = ""

if age < 18:
    output += "Vous n'avez pas l'age requis "
elif age >= 18:
    output += "Vous avez l'age requis "
elif age < 25:
    output += "Vous avez un rabais "
print(output)
```

3) Modifiez la case précédente pour demander l'âge de l'utilisateur

In []:

```
age = int(input("Quel est votre âge? "))
output = ""

if age < 18:
    output += "Vous n'avez pas l'age requis "
elif age >= 18:
    output += "Vous avez l'age requis "
elif age < 25:
    output += "Vous avez un rabais "
print(output)
```

4) Anticipez quel texte va être affiché dans la console

In []:

```
num_int = 456
num_string = "456"
total_num = num_int + int(num_string)
print(type(total_num))
```

5) Deviner un nombre aléatoire

Dans la case suivante, nous vous donnons un nombre aléatoire entre 0 et 30 dans la variable `number`, écrivez un programme qui demande à l'utilisateur de deviner le nombre, l'utilisateur a 5 chances pour le trouver, s'il se trompe, donnez-lui un indice (le nombre qu'il a écrit est-il plus grand ou plus petit que celui qu'il cherche?)

In []:

```
from random import randint
number = randint(0, 30)

x = int(input("Choisissez un nombre: "))
if x == number:
    print("Yeeah!")
elif x < number:
    print("Trop bas!")
else:
    print("Trop haut!")

x = int(input("Choisissez un nombre: "))
if x == number:
    print("Yeeah!")
elif x < number:
    print("Trop bas!")
else:
    print("Trop haut!")

x = int(input("Choisissez un nombre: "))
if x == number:
    print("Yeeah!")
elif x < number:
    print("Trop bas!")
else:
    print("Trop haut!")

x = int(input("Choisissez un nombre: "))
if x == number:
    print("Yeeah!")
elif x < number:
    print("Trop bas!")
else:
    print("Trop haut!")

x = int(input("Choisissez un nombre: "))
if x == number:
    print("Yeeah!")
elif x < number:
    print("Trop bas!")
else:
    print("Trop haut!")

# VOTRE CODE ICI
print(number)
```

In []:

```
#Autre manière de faire:
from random import randint
number = randint(0, 30)

x = int(input("Choisissez un nombre: "))
for i in range(5):
    if x==number:
        print("Yeah!")
        break
    elif x<number:
        print("Trop petit!")
    else:
        print("Trop grand!")
x = int(input("Choisissez un nombre: "))
```

In []: