

Exercices d'informatique - Série n° 1

Cours 3INOC01

Série distribuée le 29.8.2017

1. L'instruction *print*.

Déterminer l'effet de l'instruction `print` en exécutant le script suivant:

```
1 print "Bienvenue"
2 print " au cours",
3 print " d'informatique"
4 print "2007-", "2008"
```

2. Variables.

Les scripts

```
1 print "Bienvenue au cours d'informatique"
```

et

```
1 bernard="Bienvenue au cours d'informatique"
2 print bernard
```

produisent-ils le même résultat ? Expliquer la différence.

3. Variables.

Les scripts

```
1 print 3.14
```

et

```
1 bernard=3.14
2 print bernard
```

produisent-ils le même résultat ? Expliquer la différence.

4. Types de nombres.

Exécuter le script suivant:

```
1 n1=123
2 print "n1=", n1, type(n1)
3 #
4 n2=123456789012345678
5 r2=123456789012345678.
6 n3=123456789012345677
7 r3=123456789012345677.
8 #
9 print "n2=", n2, type(n2)
10 print "r2=", r2, type(r2)
11 #
12 print "n2-n3=", n2-n3
13 print "r2-r3=", r2-r3
14 #
15 n4=2**1256
16 print n4, type(n4)
```

Quelle est la différence entre les nombres de type `int`, `long` et `float`.

5. Opérations sur les nombres.

Déterminer l'effet des opérations `*`, `**`, `/`, `//` et `%` pour les nombres de type `int` et `float` en exécutant le script suivant:

```

1 a=3
2 b=5
3 print "a=",a,type(a)
4 print "b=",b,type(b)
5 print "a*b=",a*b,"a**b=",a**b,"b/a=",b/a,"b//a=",b//a,"b%a=",b%a
6 a=3.
7 b=5.
8 print "a=",a,type(a)
9 print "b=",b,type(b)
10 print "a*b=",a*b,"a**b=",a**b,"b/a=",b/a,"b//a=",b//a,"b%a=",b%a

```

6. Conversion de type.

Exécuter le script suivant et expliquer les résultats obtenus.

```

1 t1="123"
2 print type(t1)
3 print 2*t1
4 #
5 t2=int(t1)
6 print type(t2)
7 print 2*t2
8 #
9 t3=float(t1)
10 print type(t3)
11 print 2*t3
12 #
13 t4='t3'
14 print type(t4)
15 print 2*t4
16 #
17 t5=t3/7
18 print t5
19 #
20 t6=int(t5)
21 print type(t6)
22 print t6
23 #
24 t7='t5'
25 print type(t7)
26 print 2*t7

```

7. L'instruction `range`.

(1) Déterminer l'action de l'instruction `range(1,10,1)` en exécutant le script suivant:

```

1 a=range(1,10,1)
2 print a

```

(2) Déterminer l'action de l'instruction `range(x,y,pas)` pour

- (a) $(x, y, pas) = (1, 10, 3)$
- (b) $(x, y, pas) = (1, 10, -1)$
- (c) $(x, y, pas) = (10, 1, -1)$
- (d) $(x, y, pas) = (10, 1, 1)$
- (e) $(x, y, pas) = (10, 10, 1)$

8. L'instruction *range*.

- (1) En utilisant uniquement les instructions `range` et `print`, rédiger un script *Python* qui affiche [1, 4, 7, 10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43, 46, 49]
- (2) En utilisant uniquement les instructions `range` et `print`, rédiger un script *Python* qui affiche [50, 46, 42, 38, 34, 30, 26, 22, 18, 14, 10, 6]

9. Le type de données *list*.

Décrire les objets de type *list* en exécutant le script suivant:

```
1 l1=[1,2,3]
2 print type(l1)
3 print l1
4 #
5 l2=[27,"Bonjour comment allez-vous ?",3.1415,l1]
6 print type(l2)
7 print l2
8 #
9 l3=range(1,4,1)
10 print type(l3)
11 print l3
12 #
13 l4=[]
14 print type(l4)
15 print l4
16 #
17 l5=[l1,l2,l3,l4]
18 print type(l5)
19 print l5
```

Les éléments d'un objet de type *list* doivent-ils tous être du même type ?

10. Appartenance.

Déterminer l'effet de l'instruction `in` en exécutant le script suivant:

```
1 print 'a' in 'bonjour'
2 print 'b' in 'boujour'
```

11. Test d'appartenance.

Déterminer l'action des opérateurs `in` et `not in` sur une liste en exécutant le script suivant:

```
1 texte="Bonjour, comment allez-vous ?"
2 l=list(texte)+[[1,2,3],7]
3 print l
4 #
5 print "o" in l
6 print "o" in texte
7 print 1 in l
8 print 7 not in l
```

12. L'instruction *for*.

Déterminer l'action de l'instruction `for` en exécutant le script suivant:

```
1 for n in range(1,10,1):
2     print "Bienvenue au cours"
3     print "d'informatique"
```

13. Boucles.

- (1) Rédiger un script qui affiche, dans l'ordre, les nombres 1, 2, 3, ..., 9 en utilisant l'instruction `for`.
- (2) Rédiger un script qui affiche les nombres 1, 0, 1, 4, 9, 16, 25, 36 et 49 en utilisant l'instruction `for`.

14. Boucles.

Rédiger un script qui affiche toutes les lettres du texte "Bienvenue au cours d'informatique" en colonne comme ci-dessous:

```
B...
i...
e...
n...
v...
⋮
```

15. Opérations sur les chaînes.

Déterminer l'action des opérations `+` et `*` sur les chaînes en exécutant le script suivant:

```
1 t1=" Bonjour ,"
2 t2=" comment allez-vous ?"
3 print t1+t2
4 print t1*2
```

16. Variables.

Quelle est la valeur de la variable `gustave` après l'exécution du code *Python* suivant:

```
1 gustave=2
2 gustave=gustave+5
3 gustave=gustave+5
4 gustave=gustave+5
5 gustave=gustave+5
6 gustave=gustave+5
7 print gustave
```

17. Variables.

Quelle est la valeur de la variable `a` après l'exécution du code *Python* suivant:

```
1 a=2
2 a=2*a
3 a=2*a
4 a=2*a
5 print a
```

18. Variables.

Quelle est la valeur de la variable `a` après l'exécution du code *Python* suivant:

```
1 a=' Bonjour '
2 a=2*a
3 a=2*a
4 a=2*a
5 print a
```

19. Boucles.

Rédiger un script *Python* de 6 lignes au maximum qui affiche :

```
a
b b
c c c
d d d d
e e e e e
f f f f f f
g g g g g g g
h h h h h h h h
i i i i i i i i i
j j j j j j j j j
k k k k k k k k k k
l l l l l l l l l l l
m m m m m m m m m m m
n n n n n n n n n n n n
o o o o o o o o o o o o o
p p p p p p p p p p p p p
q q q q q q q q q q q q q
r r r r r r r r r r r r r r r
s s s s s s s s s s s s s s s
t t t t t t t t t t t t t t t
u u u u u u u u u u u u u u u u
v v v v v v v v v v v v v v v v
w w w w w w w w w w w w w w w w
x x x x x x x x x x x x x x x x
y y y y y y y y y y y y y y y y
z z z z z z z z z z z z z z z z z
```

20. Les instructions *if elif et else*.

Déterminer l'action des instructions **if**, **elif** et **else** en exécutant le script suivant:

```
1 print "Tapez un nombre: ",
2 n=input()
3 if n>100:
4     print "Ce nombre est plus grand que 100."
5 elif n<10:
6     print "Ce nombre est plus petit que 10."
7 else:
8     print "Ce nombre est compris entre 10 et 100."
```

21. Boucles.

Rédiger un programme *Python* qui affiche

```
PP
rrrr
oooooo
gggggggg
rrrrrrrrrr
aaaaaaaaaaaa
mmmmmmmmmmmmmmmm
mmmmmmmmmmmmmmmm
eeeeeeeeeeeeeeeeee
rrrrrrrrrrrrrrrrrrrrrr
aaaaaaaaaaaaaaaaaaaaa
vvvvvvvvvvvvvvvvvvv
eeeeeeeeeeeeeeeeee
cccccccccccccccccc
PPPPPPPPPPPPPP
yyyyyyyyyyyyyy
tttttttttt
hhhhhhhh
oooooo
nnnn
```

!!

22. Le module *turtle*.

Déterminer l'action des fonctions du module *turtle* (tortue en anglais) en exécutant le script suivant:

```

1 from turtle import *
2 title("Le module turtle")
3 setup (width=0.6, height=0.8)
4 reset ()
5 forward(100)
6 left(90)#Pour fixer les unites des angles: radians() ou degrees()
7 delay(50)
8 a=position ()
9 write(a,1)
10 color('red')#Couleurs possibles: 'blue', 'green', 'yellow', 'magenta', ...
11 right(30)
12 clear ()
13 fill(1)
14 backward(50)
15 color(0.5,0.2,0.7)#Les trois nombres sont compris entre 0 et 1
16 left(50)
17 delay(1)
18 forward(30)
19 up()
20 goto(20,30)
21 fill(0)
22 down()
23 width(3)
24 forward(30)
25 circle(20)
26 write("C'est fini !")
27 done()
28 print a

```

23. L'instruction *list*.

Déterminer l'action de l'instruction `list` en exécutant le script suivant:

```

1 t="Bonjour, comment allez-vous ?"
2 l=list(t)
3 print l
4 print type(t)
5 print type(l)

```

24. Eléments d'une liste.

Déterminer le résultat de l'instruction `l[i:j:k]` pour une liste `l` en exécutant le script suivant:

```

1 texte="Bonjour, comment allez-vous ?"
2 l=list(texte)
3 print l
4 print len(l)
5 #
6 print l[0]
7 print l[5]
8 #
9 print l[-1]
10 print l[-3]
11 #
12 print l[3:9:2]
13 print l[3:]
14 print l[: -3]
15 #
16 print l[::2]

```

```
17 print l[1:-1:2]
18 print l[::-1]
```

25. Boucles et dessins.

Rédiger un script qui reproduise (en respectant les couleurs) le dessin de la figure 1 en utilisant l'instruction `for` et le module `turtle`.

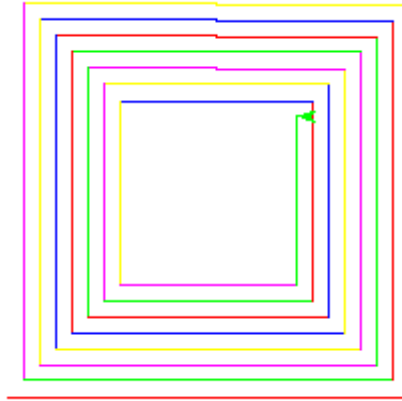


FIGURE 1. Exercice 25

26. Boucles et dessins.

Rédiger un script qui reproduise (en respectant les couleurs) le dessin de la figure 2 en utilisant l'instruction `for` et le module `turtle`.

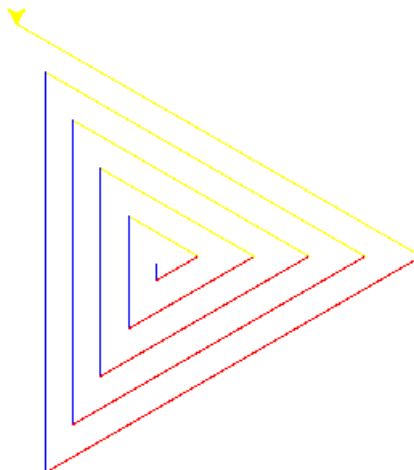


FIGURE 2. Exercice 26

27. L'instruction *while*.

Déterminer l'action de l'instruction `while` en exécutant le script suivant:

```
1 n=0
2 while n<10:
3     print(n),
4     n=n+1
```

Que se passe-t-il si on efface la ligne 4 ? Donner une explication.

28. La fonction *input()*.

Déterminer l'action de la fonction `input()` en exécutant le script suivant:

```
1 print "Entrez un nombre:",
2 entree=input()
3 print entree
```

Que se passe-t-il si l'utilisateur ne tape pas un nombre mais des lettres ?

29. La fonction *raw_input()*.

Déterminer l'action de la fonction `raw_input()` en exécutant le script suivant:

```
1 texte=raw_input("Tapez un texte: ")
2 print texte
```

30. Un petit jeu.

Rédiger un programme qui demande à l'utilisateur de taper un nombre et qui affiche "NON" tant que le nombre choisi par l'utilisateur est différent de 3 et affiche "OUI", puis s'arrête, dès que ce nombre est 3.

31. Un petit jeu.

Rédiger un programme qui demande à l'utilisateur de taper un nombre et qui affiche "Ce nombre est impair" tant que le nombre choisi par l'utilisateur est impair et affiche "Ce nombre est pair", puis s'arrête, dès que ce nombre est pair.

32. L'instruction *break*.

Déterminer l'action de l'instruction `break` en exécutant le script suivant:

```
1 print "Devinez le nombre qui se cache ..."
2 while 1<2:
3     print "Tapez un nombre: ",
4     n=input()
5     if n==3:
6         print "OUI"
7         break
8     else:
9         print "NON"
```

33. Jeu du chaud-froid.

Avec les instructions `while`, `if`, `elif` et `break`, programmer un jeu où le joueur doit deviner un nombre en faisant des essais. A chaque essai, le programme donne une réponse plus ou moins encourageante en fonction de l'erreur commise par le joueur.