



ALGORITHMIQUE

↔ Prise en main du langage python ↔

Saisir les commandes suivantes directement dans l'interpréteur afin de comprendre leur rôle :

-  >>> est l'invite de l'interpréteur : on ne saisit que ce qui est après ;
-  # indique que ce que l'on saisit est un commentaire, et donc non traité.

```
>>> #Opérations
>>> 1+1
>>> 7*5
>>> 4/2
>>> 7/2
>>> 7//2
>>> 11/7
>>> round(11/7)
>>> round(11/7,2)
>>> 10**2
>>> 2**8
>>> 2**1000
>>> 23%5
>>> 21%7
>>> 21%10
```

```
>>> #Variables et types de données.
>>> a = 123
>>> a
>>> print(a)
>>> type(a)
>>> c = 1.23**10
>>> print(c)
>>> type(c)
>>> ch = "bonjour"
>>> print(ch)
>>> type(ch)
>>> B = (5 < 3)
>>> print(B)
>>> type(B)
```

```
>>> #Transtypage
>>> str(52)
>>> int('52')
>>> float('3.14')
>>> int('3.14') #expliquer l'affichage obtenu
```

```
>>> #Pour certaines commandes, il faut le module "math".
>>> from math import *
>>> pow(5,2)
>>> sqrt(25)
>>> divmod(17,3)
>>> abs(-3)
>>> floor(3.2)
```

```
>>> #Pour des nombres pseudo-aléatoires, importer "random".
>>> from random import *
>>> random()
>>> randint(1,10)
>>> a = 0
>>> b = 50
>>> randint(a,b)
```


```
>>> #Initialisations groupées
>>> a, b, c = 1, 2, 3
>>> d = e = f = 5
```

```
>>> #Variables booléennes
>>> 5 == 3
>>> 5 != 3
>>> b1 = (5 > 3)
>>> b2 = (9 < 7)
>>> b3 = (-1 < 0)
>>> not b1
>>> b1 or b2
>>> b1 and b3
>>> not not b1
>>> not(b1 and b2)
```

```
>>> #Entrées et sorties
>>> message = input("Saisissez votre message")
>>> a = float(input("Saisissez un nombre réel"))
>>> b = int(input("Saisissez un nombre entier"))
>>> print("Somme : ", a + b)
>>> print("Différence : ", a - b)
>>> print("Produit : ", a * b)
```

↔ Chaînes de caractères ↔


Opérations sur les chaînes de caractères

- **Longueur**(chaîne) : `len` (chaîne) en  python.

```
>>> s = "abcde"
>>> len(s)
5
```

- **Concaténation** : `+` (en pseudo-langage et en  python)

```
>>> s1 = "abc"
>>> s2 = "defg"
>>> s3 = s1 + s2
>>> s3
abcdefg
```

- **Répétition** : `*` (en pseudo-langage et en  python)

```
>>> s4 = "Fi! "
>>> s5 = s4*3
>>> s5
Fi! Fi! Fi!
```

Tester et commenter chacune des instructions suivantes :

```
>>> a = "étudiant-du-bts-services-informatiques-aux-organisations"
>>> print(len(a))
>>> print("*****")

# Dans une chaîne, les caractères sont numérotés de gauche à droite 0,
  1, 2,..., n :
>>> print(a[0])
>>> print(a[1])
>>> print(a[55])
>>> print(a[54])
>>> print("*****")

# Extractions :
>>> print(a[:8])
>>> print(a[::2])
>>> print(a[16:])
>>> print(a[12:15])      # dans une tranche [p:q] le p-ième caractère est
  inclus mais pas le q-ième
>>> print("*****")

# Parcours de droite à gauche :
>>> print(a[::-1])
>>> print(a[::-2])
>>> print("*****")

# Parcours d'une chaîne :
>>> for car in a :
...     print(car + '*', end = '')      #ne pas omettre l'indentation

# Recherche de la position d'un caractère ou d'une sous-chaîne dans une
  chaîne :
>>> c = "bonjour tout le monde"
>>> c.index('m')
(affichage : 17)
>>> c.index('jour')
(affichage : 3)
>>> 'b' in c      # teste si la sous-chaîne 'b' est dans la chaîne c
(affichage : True)
>>> 'B' in c
(affichage : False)

# Modification d'une chaîne de caractères (programme à tester dans IDLE)
m = "bonjour"
M = 'B' + m[1:]
print(M)
```


Exercice 1

Tester les lignes d'instructions suivantes. Décrire ce qui se passe :

```
>>> r , pi = 12, 3.14159
>>> s = pi * r**2
>>> print(s)
>>> print(type(r), type(pi), type(s))
```

Quelle est, à votre avis, l'utilité de la fonction `type()` ?

Exercice 2

Par défaut, en  python, l'instruction `a = input()` traite a comme une chaîne de caractères. Commenter chacune des instructions suivantes :

```
a = input("entrez un nombre entier (à 8 chiffres par exemple) : ")
b = input("entrez un autre entier (à 6 chiffres par exemple): ")
c = a+b
d = b*4
print(c)
print(d)
```

Exercice 3

Tapez ce qui est demandé, observez attentivement et notez les effets (on a le droit de prendre des notes!).

```
>>> a = 3
>>> type(a)

>>> b = 5.3
>>> type(b)

>>> c = "Bonjour"
>>> c
>>> print(c)
>>> type(c)

>>> d = (a>1)
>>> d
>>> type(d)

>>> e = 6/3
>>> e
>>> type(e)

>>> f = 6//3
>>> f
>>> type(f)

>>> g = a+b
>>> type(g)
>>> g

>>> h = a+f
>>> type(h)
>>> h

>>> i = "3.14"
>>> type(i)
>>> j = float(i)
>>> type(j)
>>> k = i+j
```

```
>>> l = str(a)
>>> m = str(f)
>>> print(a, l, f, m, sep="***")
>>> print("a+f = ",a+f," et l+m = ",l+m)

>>> texte = "Salut \n tu vas bien ?"
>>> texte
>>> print(texte)
>>> print(texte, "texte")

>>> prenom = input("Tapez votre prénom : ")
>>> print("vous vous appelez : ", prenom)

>>> n = input("Donnez-moi un nombre : ")
>>> p = n+7

>>> q = float(input("Donnez-moi un entier : "))
>>> q

>>> r = int(input("Donnez-moi un entier : "))
>>> r

>>> s = int(input("Tapez PI à 10(-2) : "))

>>> t = 8
>>> u = "Truc"
>>> P = (t <= 10)
>>> Q = (u == "truc")
>>> R = (u < "truc")
>>> P
>>> Q
>>> R
>>> P and Q
>>> P or Q
>>> P or R
>>> not R
>>> P or (not R)
>>> P and (not R)
>>> (not P) and Q
>>> not(P and Q)

>>> v = 37
>>> w = 7
>>> quotient = v//w
>>> reste = v%w
>>> print(v, "=", w, "x", quotient, "+", reste)
```

Exercice 4

Tapez dans l'éditeur : `from math import *`.

Essayez de prévoir ce que va vous renvoyer la fonction avant de valider la commande (en utilisant éventuellement votre calculatrice)... et prenez des notes!

```
>>> x = 2.718
>>> round(x, 2)
>>> round(x**3, 1)
>>> abs(1 - x)
>>> floor(x)
>>> floor(1 - x)
>>> floor(abs(1 - x))

>>> round(exp(2), 3)
>>> round(log(2), 2)
>>> round(sqrt(2), 3)

>>> chr(75)
>>> ord('m')

>>> texte = "Voici Henri"
>>> float(texte)
>>> len(texte)
>>> texte.upper()
>>> texte.lower()
>>> texte[6]
>>> texte[7 :8]
>>> texte[4 :]
>>> texte[:3]
>>> texte[-3 :]
>>> texte.find("i")
>>> texte.find("i",5)
>>> texte.find("i",5,9)
```