

# Programmation Python - Algorithme - Fiche de cours

## 1. Entrées, sorties et variables

Pour lire un message on peut utiliser l'instruction :

```
variable = input(« Message »)
```

Pour afficher un message on peut utiliser l'instruction :

```
print (« Message », variable)
```

Pour convertir le type des variables on peut utiliser :

**str()** pour les chaînes de caractères

**float()** pour les nombres décimaux

**int()** pour les nombres entiers

## 2. Calcul et test avec Python

Les opérations de base :

+	addition	%	reste de la division
-	soustraction	==	est égal ?
*	multiplication	!=	est différent ?
/	division	<	est inférieur ?
**	puissance	>	est supérieur ?
//	division entière	<=	est inférieur ?
=	Affecter une valeur	>=	est inférieur ?

## 3. Les listes de valeur

```
liste = [a,b,c]
```

```
liste[n]
```

```
liste.append(a)
```

```
len(liste)
```

```
min(liste)
```

```
max(liste)
```

```
sum(liste)
```

```
liste.sort()
```

Crée une variable *liste* avec les valeurs a,b,c

Renvoie la nième valeur de la variable *liste*

Ajoute a en fin de la variable *liste*

Renvoie le nombre de valeurs de la variable *liste*

Renvoie le minimum de la variable *liste*

Renvoie le maximum de la variable *liste*

Renvoie la somme des valeurs de la variable *liste*

Trie les valeurs par ordre croissant de la variable *liste*

## 4. if else

```
if condition :
```

```
    Instruction 1
```

```
    Instruction 2
```

```
else :
```

```
    Instruction 1
```

```
    Instruction 2
```

## 5. for

```
For n in range (nombre) :    Execute nombre-1 séquences
```

```
    Instruction 1
```

```
    Instruction 2
```

```
    ...
```

## 6. while

```
while condition :
```

```
    Instruction 1
```

```
    Instruction 2
```

```
    ...
```

## 7. Importer des fonctions

`from fichier import *`

Les divers fichiers (bibliothèques) en Python :

- *math* : contient une partie des fonctions mathématiques
- *cmath* : contient les fonctions mathématiques pour nombres complexes
- *fractions* : contient les fonctions mathématiques pour nombres rationnels
- *random* : contient les fonctions de tirages aléatoires et les fonctions de probabilités

## 8. Définir une fonction

```
def nom_fonction (paramètre1, paramètre2, etc...):  
    instructions  
    ...  
    return résultat
```

## 9. Fonctions mathématiques : « *math* »

<b>fabs(x)</b>	: retourne la valeur absolue de x
<b>factorial(x)</b>	: retourne la valeur absolue de x
<b>exp(x)</b>	: retourne l'exponentielle de x
<b>log(x)</b>	: retourne le logarithme naturel de x
<b>log10(x)</b>	: retourne le logarithme décimal de x
<b>pow(x,y)</b>	: retourne x puissance y
<b>sqrt(x)</b>	: retourne la racine carrée de x
<b>cos(x)</b>	: retourne la valeur de cos(x)
<b>sin(x)</b>	: retourne la valeur de sin(x)
<b>tan(x)</b>	: retourne la valeur de tan(x)

<b>acos(x)</b>	: retourne la valeur de arccos(x)
<b>asin(x)</b>	: retourne la valeur de arcsin(x)
<b>atan(x)</b>	: retourne la valeur de arctan(x)
<b>pi</b>	: retourne la valeur du nombre pi
<b>e</b>	: retourne la valeur du nombre e

## 10. Fonctions mathématiques : « *fractions* »

<b>Fraction(a,b)</b>	: définit une fraction irréductible = $a/b$
<b>Fraction(nombre)</b>	: définit la fraction irréductible = $a/b$
<b>fraction.numerator</b>	: retourne le numérateur de <i>fraction</i>
<b>fraction.denominator</b>	: retourne le dénominateur de <i>fraction</i>
<b>gcd(a,b)</b>	: retourne le PGCD(a,b)

## 11. Fonctions mathématiques : « *random* »

<b>random(x)</b>	: retourne un nombre décimal entre 0,0 et 1,0
<b>randint(1,5)</b>	: retourne un nombre entier entre 1 et 5
<b>uniform(a,b)</b>	: retourne un nombre décimal entre a et b
<b>expovariate(a)</b>	: retourne un nombre décimal distribué selon la loi exponentielle de paramètre a
<b>normalvariate(mu,sigma)</b>	: retourne un nombre décimal distribué selon la loi normale de paramètre mu et sigma

## 12. Logiciel à télécharger

PC : <https://edupython.tuxfamily.org/>

Smartphone : « QPython 3 » pour Android ou Iphone