

Programmation Python – Algorithme – Fiche de cours

1. Les bibliothèques

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

Pour appeler une bibliothèque on peut utiliser l'instruction :

```
from fichier import *
```

2. 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)
```

3. Types de données

a. Typage, création et initialisation d'une variable

Une variable Python est typée (catégorie), créée (réservation d'une zone mémoire / affectation d'un identifiant) et initialisée au cours d'une seule instruction :

```
variable = valeur
```

b. Conversion de type (transtypage)

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

c. Variables booléennes

- définition : Une variable booléenne peut prendre 2 valeurs : True / False

3. Opérations et calculs

a. Opérations algébriques

+	addition	%	reste de la division
-	soustraction	**	puissance
*	multiplication	//	division entière
/	division	=	Affecter une valeur

b. Opérations booléennes

- fonction NON : **not** condition
- fonction ET : condition1 **and** condition2
- fonction OU : condition1 **or** condition2

d. Tests logiques

- condition1 == condition2 est égal ?
- condition1 != condition2 est différent ?
- condition1 < condition2 est inférieur ?
- condition1 > condition2 est supérieur ?
- condition1 <= condition2 est inférieur ou égal ?
- condition1 >= condition2 est supérieur ou égal ?

4. Instructions sélectives

if *condition* :

Instruction 1

Instruction 2

else :

Instruction 1

Instruction 2

5. Les listes de valeur

- `liste = [a,b,c]` Crée une variable *liste* avec les valeurs a,b,c
- `liste[n]` Renvoie la nième valeur de la variable *liste*
- `liste.append(a)` Ajoute a en fin de la variable *liste*
- `len(liste)` Renvoie le nombre de valeurs de la variable *liste*
- `min(liste)` Renvoie le minimum de la variable *liste*
- `max(liste)` Renvoie le maximum de la variable *liste*
- `sum(liste)` Renvoie la somme des valeurs de la variable *liste*
- `liste.sort()` Trie les valeurs par ordre croissant de la variable *liste*

6. Instructions itératives

- nombre de boucles définies

for *n* in range (*nombre*) :

Instruction 1

Instruction 2

...

- nombre de boucles non définies

while *condition* :

Instruction 1

Instruction 2

...

7. Définir une fonction

def *nom_fonction* (paramètre1, paramètre2, etc...):

instructions

...

return resultat

8. Utilisation des fichiers

`fichier.open(« fichier », « options »)` : ouverture d'un fichier avec options

r en lecture uniquement

r+ en lecture / écriture

w en écriture uniquement

w+ en écriture / lecture

`fichier.readline()` : lecture de la ligne courante

`fichier.readlines()` : lecture de toutes les lignes du fichier

`fichier.write(« ligne »)` : ajoute une ligne

`fichier.writelines(« lignes »)` : ajoute plusieurs ligne

`fichier.close()` : fermeture du fichier

9. Fonctions mathématiques : « *math* »

fabs(x)	: retourne la valeur absolue de x
factorial(x)	: retourne la valeur absolue de x
exp(x)	: retourne l'exponentielle de x
log(x)	: retourne le logarithme naturel de x
log10(x)	: retourne le logarithme décimal de x
pow(x,y)	: retourne x puissance y
sqrt(x)	: retourne la racine carrée de x
cos(x)	: retourne la valeur de cos(x)
sin(x)	: retourne la valeur de sin(x)
tan(x)	: retourne la valeur de tan(x)
acos(x)	: retourne la valeur de arccos(x)
asin(x)	: retourne la valeur de arcsin(x)
atan(x)	: retourne la valeur de arctan(x)
pi	: retourne la valeur du nombre pi
e	: retourne la valeur du nombre e
gcd(a,b)	: retourne le PGCD(a,b)

10. Fonctions mathématiques : « *fractions* »

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

11. Fonctions mathématiques : « *random* »

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

12. Représentation graphique : « *matplotlib* »

import matplotlib.pyplot as graphique	: crée l'objet graphique avec des propriétés d'accès à la carte graphique
graphique.plot(x, y, 'attribut')	: construit la courbe y(x) avec attributs pour les points 'o' : représentés par des ronds '+' : représentés par des croix 'r' : en couleur rouge
graphique.show()	: affiche la fenêtre graphique