

Mémento Python

■ Commentaires

```
# commentaires
```

■ Aide interactive dans python

```
help() lance l'aide interactive
help(m) aide sur le module m
help(f) aide sur la fonction f
dir(m) affiche les commande du module m
```

■ Importer un module

```
import module importe tout le module, fonctions
préfixées par le nom du module
import module as préfixe importe tout le module, fonctions
préfixées par le préfixe
from module import * importe tout le module sans préfixe
from module import nom importe la fonction nom du mod-
ule, sans préfixe
```

■ Principaux types de données

Type	Description	Exemple
int	entiers 32 bits	3,-4
long	entiers > 32 bits	101L
float	nombre à virgule	3.0, -6.55
complex	nombre complexe	1.2j
bool	booléen	True, False
str	chaîne de caractères	"Joffre", 'az'
tuple	séquence non mutable	(2,4,7)
list	séquence mutable	[2,'x',3.1]
dict	dictionnaire	{'x':2, 'y':5}

■ Principales méthodes des chaînes de caractères

S.method()	Sortie
capitalize	met la première lettre en majuscule
join(seq)	concatène tous les items de seq
lower()	met S en minuscule
upper()	met S en majuscule
split([sep])	coupe S en blocs limités par sep

■ Principales méthodes des tuples

T.method()	Sortie
count(obj)	retourne le nombre d'occurrences de obj dans T
index(obj)	retourne la position de la première occurrence de obj dans T

■ Syntaxe des principales structures

Affectation

```
var = expr
```

Entrée/sortie en console

```
var = input([prompt])
var = raw_input([prompt])
print expr[,...] 
```

Choix

```
if (expression booléenne):
    expression
elif (expression booléenne):
    expression
[else:
    expression]
```

Répétition

```
while expression booléenne:
    expression
```

Boucler sur un itérable

```
for var in objet_iterable:
    expression
```

Définition de fonction

```
def nom_de_fonction(paramètres):
    expression
```

Appel de fonction

```
nom_de_fonction(arguments)
```

■ Principales méthodes des listes

L.method()	Sortie
append(obj)	ajoute obj à la fin de L
count(obj)	retourne le nombre d'occurrences de obj dans L
index(obj)	retourne la position de la première occurrence de obj dans L
pop([pos])	retourne l'item en position pos dans L (ou le dernier élément)
remove(obj)	enlève la première occurrence de obj dans L
reverse()	retourne la liste L sur place
sort()	tri la liste L sur place

■ Principales méthodes des fichiers

F.method()	Sortie
read([n])	retourne les n caractères suivants de F (ou tout le fichier si n est absent)
write(s)	écrit la chaîne de caractères s dans F
close()	ferme le fichier

■ Fonctions disponibles par défaut

Fonction	Sortie
abs(x)	valeur absolue de x
dict(x)	dictionnaire vide
float(x)	convertit int ou string en float
id(obj)	adresse mémoire de obj
int(x)	convertit float ou string en int
len(s)	nombre d'objets dans la séquence s
list()	liste vide
max(s)	valeur maximale dans la séquence s
min(s)	valeur minimale dans la séquence s
open(f)	ouvre le fichier f en lecture
ord(c)	code ASCII de c
pow(x,y)	x**y
range(x)	liste de x entiers allant de 0 à x-1
round(x,n)	arrondi le float x au rang n
str(obj)	convertit obj en chaîne de caractères
sum(s)	somme de la séquence numérique s
tuple(items)	convertit les items en tuple
type(obj)	donne le type de obj

■ Fonctions principales du module math

Fonction	Sortie
ceil(x)	plus petit entier >= x
cos(x)	cosinus de x radians
degrees(x)	convertit x en degrés
radians(x)	convertit x en radians
exp(x)	e**x
floor(x)	plus grand entier <=x
log(x,b)	log de x (en base b si précisée)
pow(x,y)	x**y
sin(x)	sinus de x radians
sqrt(x)	racine positive de x
tan(x)	tangente de x radians
pi	constante π à 15 chiffres significatifs
e	constante e à 15 chiffres significatifs

■ Principaux opérateurs

Fonction	Sortie
**	exponentiation
*, /	multiplication, division
//, %	division entière, reste
+, -	addition, division
>, <, >=, <=, !=, ==	comparaisons
in, not in	tests d'appartenance
not, and, or	opérateurs booléens non, et, ou