

Formulaire d'informatique

Mathématiques
Classe Préparatoire PSI*
Lycée Jean Perrin, Marseille

Sylvain Damour
`sylvain.damour@prepas.org`

Année 2020–2021

Formulaire de programmation en Python

1 Types

bool	<i>Booléen.</i> Exemple : True et False.
int	<i>Entier.</i> Exemple : 42, -8.
float	<i>Flottant.</i> Exemple : 36.13, 42.0.
str	<i>Chaîne de caractères.</i> Exemple : "Ceci est une chaîne".
list	<i>Liste.</i> Exemple : [a, b, c].

2 Variables



=	<i>Affectation.</i> Exemple : a = 1.
, = ,	Affectation parallèle. Exemple : a, b, c = 1, 2, 3.
a, b = b, a	Permutation des valeurs de a et b.
	Autre écriture :
	c = a
	a = b
	b = c

3 Opérateurs algébriques

+	Addition.
-	Soustraction.
*	Multiplication.
/	Division.
**	Puissance.
//	Quotient de la division euclidienne des entiers.
%	Reste de la division euclidienne des entiers. (Se prononce « modulo ».)



+=	Affectation et addition combinées. Exemple : i += 1
	Autre écriture :
	i = i + 1
	Il existe de façon similaire les instructions : -= *= /= **= //= %=

4 Opérateurs de comparaison

<code>==</code>	Égal.
<code>!=</code>	Différent.
<code><</code>	Strictement inférieur.
<code><=</code>	Inférieur ou égal.
<code>></code>	Strictement supérieur.
<code>>=</code>	Supérieur ou égal.

5 Opérateurs logiques

<code>and</code>	Et.
<code>or</code>	Ou.
<code>not</code>	Non.

6 NumPy

NumPy, abréviation de « Numerical Python », est la bibliothèque de calcul numérique pour Python. Il faut l'importer avec la commande :

```
import numpy as np
```

<code>np.pi</code>	Nombre π .
<code>np.e</code>	Nombre e .
<code>np.sqrt(x)</code>	Fonction racine carrée : $\sqrt{x}$.
<code>np.exp(x)</code>	Fonction exponentielle : e^x .
<code>np.log(x)</code>	Fonction logarithme népérien : $\ln x$.
<code>np.sin(x)</code>	Fonction sinus : $\sin x$.
<code>np.cos(x)</code>	Fonction cosinus : $\cos x$.
<code>np.tan(x)</code>	Fonction tangente : $\tan x$.
<code>np.abs(x)</code>	Fonction valeur absolue : $ x $.
<code>np.floor(x)</code>	Fonction partie entière : $\lfloor x \rfloor$.

7 Listes

	<code>[]</code>	Liste vide. Exemple : <code>L = []</code> .
	<code>[...]</code>	Création d'une liste. Exemple : <code>L = [1, 5, 7, 3]</code> .
	<code>L[i]</code>	Élément numéro i de la liste. La numérotation commence à 0.
	<code>len(L)</code>	Longueur de la liste, c'est-à-dire son nombre d'éléments.
	<code>L+K</code>	Concaténation des listes L et K . Exemple : <code>[1, 2, 3] + [4, 5, 6]</code> donne <code>[1, 2, 3, 4, 5, 6]</code> .
	<code>L.append(x)</code>	Ajout de l'élément x à la fin de la liste.
	Autre écriture : <code>L = L + [x]</code>	
	<code>L[i:j]</code>	Sous-liste du i -ème élément inclus au j -ème élément exclus .
	<code>L[:j]</code>	Sous-liste du premier élément inclus au j -ème élément exclus .
	<code>L[i:]</code>	Sous-liste du i -ème élément inclus au dernier élément.
	<code>range(n)</code>	Crée la liste <code>[0, 1, 2, ..., n - 1]</code> .
	<code>range(m, n)</code>	Crée la liste <code>[m, m + 1, ..., n - 1]</code> .
	<code>range(m, n, i)</code>	Crée la liste d'entiers de m inclus à n exclus avec un incrément i . Exemple : <code>range(4, 12, 2)</code> donne <code>[4, 6, 8, 10]</code> .

8 Chaînes de caractères

	<code>""</code>	Chaîne vide. Exemple : <code>s = ""</code> .
	<code>" ... "</code>	Création d'une chaîne. Exemple : <code>s = "Bonjour !"</code> .
	<code>s[i]</code>	Élément numéro i de la chaîne. La numérotation commence à 0.
	<code>len(s)</code>	Longueur de la chaîne, c'est-à-dire son nombre de caractères.
	<code>s+t</code>	Concaténation des chaînes s et t . Exemple : <code>"abc"+"def"</code> donne <code>"abcde"</code> .
	<code>s[i:j]</code>	Sous-chaîne du i -ème élément inclus au j -ème élément exclus .
	<code>s[:j]</code>	Sous-chaîne du premier élément inclus au j -ème élément exclus .
	<code>s[i:]</code>	Sous-chaîne du i -ème élément inclus au dernier élément.

9 Entrées-sorties

<code>print</code>	Affiche une chaîne de caractères ou le contenu d'une variable. Exemples : <code>print("Bonjour")</code> , <code>print(a)</code> , <code>print("La valeur est ", a)</code>
<code>input</code>	Affiche un message et attend que l'utilisateur saisisse des caractères au clavier. La valeur retournée est de type chaîne de caractères. Exemple : <code>prenom = input("Entrez votre prénom : ")</code>

10 Instruction conditionnelle IF

10.1 Instruction if

```
if condition :  
    instruction 1  
    instruction 2  
    ...  
suite des instructions
```

10.2 Instruction else

```
if condition :  
    instruction 1  
    instruction 2  
    ...  
else :  
    instruction A  
    instruction B  
    ...  
suite des instructions
```

10.3 Instruction elif

```
if condition :  
    instruction 1  
    instruction 2  
    ...  
elif condition :  
    instruction i  
    instruction ii  
    ...  
elif condition :  
    instructions  
    ...  
...  
else :  
    instructions  
    ...  
suite des instructions
```

10.4 Exemple

```
1 if x==1 :  
2     print("x est égal a 1.")  
3 elif x==2 :  
4     print("x est égal a 2.")  
5 elif x==3 :  
6     print("x est égal a 3.")  
7 else :  
8     print("x est différent de 1, de 2 et de 3.")  
9 print("fin")
```

11 Boucle conditionnelle WHILE

11.1 Instruction while

```
while condition :  
    instruction 1  
    instruction 2  
    ...  
suite des instructions
```

11.2 Exemple

```
1 c = 1  
2 while c <= 100 :  
3     print(c)  
4     c = c + 1  
5 print("fin")
```

12 Boucle inconditionnelle FOR

12.1 Instruction for

```
for compteur in range(entier) :  
    instruction 1  
    instruction 2  
    ...  
suite des instructions
```

12.2 Exemple

```
1 S = 0  
2 for i in range(100) :  
3     S = S + i**2  
4 print(S)
```

13 Fichiers textes

13.1 Ouvrir ou fermer un fichier

`open` Ouvrir un fichier
en lecture: `f = open("nom_du_fichier.txt", "r")`
en écriture: `f = open("nom_du_fichier.txt", "w")`

`close` Fermer un fichier: `f.close()`

13.2 Lire ou écrire dans un fichier

`readline` Lire une ligne: `ligne = f.readline()`



Si on est à la fin du fichier, `readline` retourne la chaîne vide `""`.

`write` Écrire une ligne: `f.write("Ligne à écrire dans le fichier.\n")`



Les caractères `\n` à la fin de la chaîne de caractères indique qu'il faut passer à la ligne.

13.3 Exemple

```
1 def lire_lignes():
2     f = open("toto.txt", "r")
3     ligne = f.readline()
4     while ligne != "":
5         print(ligne)
6         ligne = f.readline()
7     f.close()
8
9 def ecrire_lignes():
10    f = open("toto.txt", "w")
11    f.write("Bonjour,\n")
12    f.write("ce fichier contient deux lignes.\n")
13    f.close()
```