



# Initiation à la programmation avec Python



## 1 Téléchargement et installation de Python

### Activité N° 1 :

1. Télécharger la dernière version (Python 3.7) à partir du lien suivant : <https://www.python.org/downloads/>
2. Installer le fichier téléchargé.

## 2 Test de l'environnement de Python

### Activité N° 2 :

1. Lancer la console interactive python (**IDLE**) puis taper le script suivant :

```
>>> print(13+2)
```

2. Créer un nouveau fichier, récrire le script et enregistrer le résultat sous le nom « **test.py** »
3. Exécuter ce script en cliquant sur le menu « **Run** ».

## 3 L'instruction d'affichage en Python

### Activité N° 3 :

Pour les cas suivants, taper le code correspondant puis cocher les cases correspondantes.

N° Cas	Code	Code correcte		Message d'erreur s'il existe
		Oui	Non	
1	<b>Print</b> ("La bienvenue dans Python")			
2	<b>print</b> ("La bienvenue dans Python")			
3	<b>print</b> (La bienvenue dans Python)			
4	<b>Print</b> "La bienvenue dans Python"			

### Activité N° 4 :

1. Créer un nouveau fichier python intitulé « **activite4** », taper le code ci-dessous puis le tester.

```
a = 12
print ( a )
```

2. Modifier le code de l'activité précédente afin d'afficher le résultat suivant « **a = 12** »

## 4 Les variables en Python

### Activité N° 5 :

Créer un nouveau fichier python intitulé « **variable** », écrire un programme permettant de :

1. Affecter l'entier 5 dans une variable « **x** » puis afficher son contenu et son type.
2. Affecter l'entier 5.00 dans une variable « **y** » puis afficher son contenu et son type.
3. Affecter le message "bonjour" dans une variable « **ch** » puis afficher son contenu et son type.

## 5 L'instruction d'entrée en Python

### Activité N° 6 :

Créer un nouveau fichier python intitulé « **entree** », taper le code ci-dessous puis le tester et dégager vos constatations.

```
a = input("saisir la valeur de a")
print(" La valeur saisie = " , a )
```

### Activité N° 7 :

Créer un nouveau fichier python intitulé « **entree2** », écrire un programme permettant de :

1. Saisir le nom, l'âge et la moyenne d'un élève.
2. Afficher le type des variables saisies
3. Convertir âge et moyenne respectivement en entier et réel
4. Afficher les variables saisies.

N.B : afficher la moyenne avec deux chiffres après la virgule

## 6 Les nombres en Python

### Activité N° 8 :

Créer un nouveau fichier python intitulé « **nombre** », écrire un programme permettant de générer aléatoirement 4 chiffres non nuls qui seront les coordonnées (x et y) de deux points A et B puis de calculer et d'afficher la distance entre les deux points avec deux chiffres après la virgule. Sachant que la distance entre deux points

$$\sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}$$

## 7 Les chaînes de caractères

### Activité N° 9 :

Créer un nouveau fichier python intitulé « **chaîne** », taper le code ci-dessous, le tester et terminer le tableau suivant :

Instruction	Résultat	Rôle
ch = "Bonne Année 2019"		
print("Linge N° 1 : ", ch[0], ch[6], ch[-1])		
print("Linge N° 2 : ", ch[0:3])		
print("Linge N° 3 : ", ch[6:])		
print("Linge N° 6 : ", ch[::-1])		
print("Linge N° 7 : ", ch[:3])		

### Activité N° 10 :

Créer un nouveau fichier python intitulé « **nombre** », écrire un programme permettant :

1. Stocker dans la variable « ch » la chaîne « *Le héron au long bec emmanché d'un long cou* »
2. Afficher la taille de ch.
3. Afficher le nombre d'occurrences de « **on** » dans ch.

8

## Les listes (les tableaux) en Python

Vocabulaire et syntaxe : `Nom_tableau = [ Valeur1 , Valeur2 , Valeur3 , ValeurN ]`

### Activité N° 11 :

Créer un nouveau fichier python intitulé « **nombre** », écrire un programme permettant :

1. Stocker dans un tableau les noms des classes où en enseigne l'informatique.
2. Afficher le tableau.
3. Afficher le premier et le dernier élément du tableau.

## Les structures de contrôle conditionnelles en Python

Vocabulaire et syntaxe (Généralisée) :

```
if ( condition ) :
    Traitements1
else :
    if ( condition2 ) :
        Traitements2
    else :
        TraitementsN
```

9

### Activité N° 12 :

Créer un nouveau fichier python intitulé « **Conseil** », écrire un programme qui permet la saisie de la moyenne d'un élève (Moy) et d'afficher un message suivant les cas :

- « **Redouble** » dans le cas où  $Moy < 9$
- « **Racheté** » dans le cas où  $9 \leq Moy < 10$ .
- « **Admis** » dans le cas où  $Moy \geq 10$ .

## La structure de contrôle itérative à forme complète (Pour) en Python

Vocabulaire et syntaxe :

```
for compteur in séquence :
    Instruction1
    Instruction2
    InstructionN
```

N.B : séquence peut être :

- une liste de valeur ( valeur1 , valeur2 , valeurN )
- `enumerate(un tableau)` :
- un intervalle `range ( Borne Inf , Borne Sup )`
- un intervalle de 0 à N-1 `range(N)`
- une chaîne de caractères

10

### Activité N° 13 :

Créer un nouveau fichier python intitulé « **Affichage\_Chaine** » puis écrire un programme qui permet la saisie d'une chaîne de caractères (Ch) puis d'afficher tous les caractères de cette chaîne.

### Activité N° 14 :

Créer un nouveau fichier python intitulé « **Factorielle** » puis écrire un programme qui permet la saisie d'un entier N puis de calculer et d'afficher sa factorielle (On supposera que  $N \geq 0$ ).

**11** Dessin 2D en Python**Activité N° 15 :**

Créer un nouveau fichier python intitulé « **rectangle** », taper le code ci-dessous puis le tester et dégager vos constatations.

```
from turtle import *  
for i in range (4) :  
    forward(60)  
    right(90)
```

**Activité N° 16 :**

Modifier le programme « **rectangle** » afin de tracer une forme de 6 côtés (hexagone)

```
from turtle import *  
.....  
.....  
.....  
.....  
.....  
.....
```

**Activité N° 17 :**

Modifier le programme « **rectangle** » afin de tracer une forme de N côtés avec N saisie par l'utilisateur.

```
from turtle import *  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....
```

## 12 Annexes

Opérateur	Nomination	Effet	Exemple	Résultat (avec x = 7)
+	Addition	Ajouter deux valeurs	x+3	.....
-	Soustraction	Soustraire deux valeurs	x-3	.....
*	Multiplication	Multiplier deux valeurs	x*3	.....
/	Division réelle	Diviser deux valeurs	x/3	.....
%	Reste de la division entière	Le reste de la division entière	x%3	.....
//	Quotient de la division entière	Le quotient de la division entière	x//3	.....

## Les méthodes de formules mathématiques

Méthode	Description
<b>abs</b> ( a )	Retourne la valeur absolue de a
<b>round</b> ( a )	Retourne l'entier arrondi le plus proche de a
<b>divmod</b> ( a , b )	Retourne un couple formé par le quotient et le reste de la division entière ( Q , R )
<b>pow</b> ( a , b )	Retourne a à la puissance b
<b>min</b> ( a , b )	Retourne le plus petit des paramètres a ou b
<b>max</b> ( a , b )	Retourne le plus grand des paramètres a ou b

## Les méthodes de formules mathématiques

Méthode	Description	Bibliothèque
<b>sqrt</b> ( a )	Retourne la racine carrée de a	<code>from math import *</code>
<b>randrange</b> (a,b)	Retourne aléatoirement un entier entre [a,b[	<code>from random import *</code>
<b>randint</b> (a,b)	Retourne aléatoirement un entier entre [a,b]	<code>from random import *</code>
<b>random.random</b> ()	Retourne aléatoirement un réel entre [0,1[	<code>from random import *</code>

Les opérateurs de comparaison				
Opérateur	Dénomination	Effet	Exemple	Résultat (avec x valant 7)
<b>==</b> à ne pas confondre avec le signe d'affectation (=)!!	opérateur d'égalité	Compare deux valeurs et vérifie leur égalité	x == 3	.....
<	opérateur d'infériorité stricte	Vérifie qu'une variable est strictement inférieure à une valeur	x < 3	.....
<=	opérateur d'infériorité	Vérifie qu'une variable est inférieure ou égale à une valeur	x <= 3	.....
>	opérateur de supériorité stricte	Vérifie qu'une variable est strictement supérieure à une valeur	x > 3	.....
>=	opérateur de supériorité	Vérifie qu'une variable est supérieure ou égale à une valeur	x >= 3	.....
!=	opérateur de différence	Vérifie qu'une variable est différente d'une valeur	x != 3	.....
<b>in</b> (liste de valeur)	dans	Vérifier que la valeur d'une variable dans une liste de valeurs	x <b>in</b> (1, 2, 8)	.....
<b>in range</b> (Borne inf , Borne sup)	dans	Vérifier que la valeur d'une variable existe dans un intervalle de borne inférieur à borne supérieur. Ces deux bornes doivent être des entiers.	x <b>in</b> <b>range</b> (8,50)	..... x = 50 .....

Les opérateurs logiques (booléens)			
Opérateur	Nomination	Effet	Syntaxe
ou <b>or</b>	OU logique	Vérifie qu'une des conditions est réalisée	((condition1)   (condition2))
& ou <b>and</b>	ET logique	Vérifie que toutes les conditions sont réalisées	((condition1) & (condition2))
<b>is not</b>	NON logique	Inverse l'état d'une variable booléenne	( <b>is not</b> condition)

## Les méthodes prédéfinies sur les chaînes de caractères dans python

La méthode	Syntaxe	Rôle	Exemple
<b>len</b>	<b>len</b> ( chaîne )	Détermine la longueur d'une chaîne de caractères c.a.d. le nombre de caractères constituant la chaîne	<pre>ch = "cours python" L = len ( ch ) print ( "la longueur " , L )</pre> Le programme affiche 12
<b>find</b>	<b>Chaîne.find</b> ( chaîneR , position initiale gauche , position finale droite)  Position initiale et Position finale sont facultatives	Chercher une chaîne de caractères chaîneR dans une chaîne de caractères chaîne de la gauche vers la droite	<pre>⇒ chaîne = "Cours python 2018-2019" chaîneR = "python" p = chaîne.find ( chaîneR ) print ( chaîneR , " existe à la position = " , p )</pre> le programme affiche : chaîneR existe à la position = 6 <pre>⇒ chaîne = "Cours python 2018-2019" chaîneR = "Python" p = chaîne.find ( chaîneR ) print ( chaîneR , " existe dans la position = " , p )</pre> le programme affiche -1 car la chaîne Python n'existe pas
<b>upper</b>	<b>chaîne.upper</b> ( )	Convertit une chaîne en majuscule	<pre>chaîne = "cours python" ch = chaîne.upper ( ) print ( " La chaîne en majuscule = " , ch )</pre> le programme affiche : COURS PYTHON
<b>count</b>	<b>chaîne.count</b> ( chaîne1 )	Compter le nombre d'apparition de la chaîne chaîne1 dans la chaîne Chaîne.	<pre>chaîne = "Cours python et Cours 2018-2019" chaîne1 = "Cours" p = chaîne.count ( chaîne1 ) print ( chaîne1 , " existe " , p , " fois " , chaîne )</pre> le programme affiche : Cours existe 2 fois Cours python et Cours 2018-2019
<b>capitalize</b> ( )	<b>chaîne.capitalize</b> ( )	Convertit le premier caractère en majuscule	<pre>chaîne = "cours python et cours programmation 2018-2019" chaîne1 = chaîne.capitalize ( ) print ( chaîne1 )</pre> le programme affiche : Cours python et cours programmation 2018-2019
<b>lower</b>	<b>chaîne.lower</b> ( )	Convertir une chaîne en minuscule	<pre>chaîne = "COURS PYTHON" chaîne = chaîne.lower ( ) print ( " La chaîne en minuscule = " , chaîne )</pre> le programme affiche cours python
<b>strip</b>	<b>chaîne.strip</b> ( )	enlève les espaces éventuels au début et à la fin de la chaîne	<pre>chaîne = " Cours python et cours programmation 2018-2019 "</pre> <pre>chaîne1 = chaîne.strip ( ) print ( chaîne1 )</pre> Le programme affiche : Cours python et cours programmation 2018-2019





## Les méthodes prédéfinies sur les tableaux dans python

méthode	Syntaxe	Rôle	Exemple
<b>len</b>	<b>Len</b> ( tableau )	Détermine la longueur d'un tableau	<pre> classe = [ 7 , 8 , 9 , 1 , "2 Tec info", "2 Sciences", "2Lettres",           "2 Eco et services", "2 Mécanique" ]  L = len ( classe ) print ( "La longueur = " , L ) Le programme affiche : La longueur = 9 </pre>
<b>del</b>	<b>del</b> ( tableau [ indice ] )	Supprime d'un tableau un élément quelconque (à partir de son index).	<pre> classe = [ 7 , 8 , 9 , 1 , "2 Tec info", "2 Sciences", "2Lettres",           "2 Eco et services", "2 Mécanique" ]  del ( classe [ 8 ] ) print ( classe ) Le programme affiche : [ 7 , 8 , 9 , 1 , '2 Tec info', '2 Sciences',                         '2Lettres', '2 Eco et services' ] </pre>
<b>append</b>	<b>tableau.append</b> ( valeur )	Ajouter une valeur à un tableau	<pre> classe = [ 7 , 8 , 9 , 1 , "2 Tec info", "2 Sciences", "2Lettres",           "2 Eco et services", "2 Mécanique" ]  classe.append ( "3 Math" ) print ( classe ) Le programme affiche : [ 7 , 8 , 9 , 1 , '2 Tec info', '2 Sciences',                         '2Lettres', '2 Eco et services', '3 Math' ] </pre>
<b>insert</b>	<b>tableau.insert</b> ( position , valeur )	insérer une valeur à un tableau dans une position donnée	<pre> classe = [ 7 , 8 , 9 , 1 , "2 Tec info", "2 Sciences", "2 Lettres",           "3 Math" ]  classe.insert ( 7 , "2 Eco et services" ) print ( classe ) Le programme affiche : [ 7 , 8 , 9 , 1 , '2 Tec info', '2 Sciences', '2Lettres',   '2 Eco et services', '3 Math' ] </pre>
<b>count</b>	<b>tableau.count</b> ( valeur )	Compter le nombre d'apparition de la valeur dans le tableau.	<pre> T = [ 1 , 2 , 3 , 8 , 7 , 2 , 6 , 1 ] Nb = T.count ( 2 ) print ( Nb ) Le programme affiche : 2 </pre>
<b>sort</b>	<b>tableau.sort</b> ( reverse = True/False )	Trier un tableau dans l'ordre croissant si reverse = True ou dans l'ordre décroissant si reverse = False	<pre> ⇒ T = [ 1 , 2 , 3 , 8 , 7 , 2 , 6 , 1 ]    Nb = T.sort ( reverse = False )    print ( T )    Le programme affiche : [ 1 , 1 , 2 , 2 , 3 , 6 , 7 , 8 ] ⇒ T = [ 1 , 2 , 3 , 8 , 7 , 2 , 6 , 1 ]    Nb = T.sort ( reverse = True )    print ( T )    Le programme affiche : [ 8 , 7 , 6 , 3 , 2 , 2 , 1 , 1 ] ⇒ T = [ 1 , 2 , 3 , 8 , 7 , 2 , 6 , 1 ]    Nb = T.sort ( )    print ( T )    Le programme affiche : [ 1 , 1 , 2 , 2 , 3 , 6 , 7 , 8 ] </pre>
<b>index</b>	<b>tableau.index</b> ( valeur )	Retourne la position d'une valeur recherchée dans le tableau	<pre> T = [ 123 , "Ali" , "Kamel" , "abc" ] print ( "Indice de Ali : " , T.index ( "Ali" ) ) Le programme affiche : Index de Ali : 1  T = [ 123 , "Ali" , "Kamel" , "abc" ] print ( "Indice de Ali : " , T.index ( "Alia" ) ) Le programme génère une erreur </pre>