

## Fiche Outils : Programmer (Python)

Significations	Algorithmme	Commande Ti
Saisir une valeur après demande(ou sans) et affecte à X cette valeur sous forme d'une <b>chaîne</b> , ou d'un <b>entier</b> ou d'un <b>décimal</b> .	Saisir X Ou Lire la valeur de X	<b>X=input() ;</b> <b>X=int(input()) ;</b> <b>X=float(input()) ;</b>
Afficher à l'écran la valeur de la variable X ou affiche X= puis la valeur de la variable X	Afficher la valeur de X	<b>print("X=",X) ;</b>
La valeur $a^2+1$ va se stocker dans la mémoire de la variable X	$X \leftarrow a^2+1$	<b>X=a**2+1 ;</b>
Si les conditions sont vraies alors on exécute instructions1 sinon on exécute instructions2	Si ... Alors ... Sinon ...	<b>if conditions :</b> instructions1 ; <b>else :</b> instructions2 ;
On veut faire n fois les mêmes instructions	Pour i allant de 1 à n Faire ...	<b>for i in range (1,n+1)</b> : instructions;
Tant que des conditions sont vraies, répéter les instructions	Tant que ... Faire ...	<b>while conditions :</b> Instructions ;
Ecrire une fonction	<b>Def NomFonction(paramètres) :</b> Instructions ; <b>Return ValeurReponse</b>	
Quelques instructions importantes Ajouter <b>import math</b> <b>import random</b> au début de votre programme.		
math.pi	Valeur exacte de pi	
math.pow(x,n)	x puissance n	
a%b	Reste de la division de a par b	
a//b	Quotient entier de a par b	
math.sqrt(x)	Racine carré de x	
math.exp	Exponentielle de x	
math.fabs	Valeur absolue de x	
math.e	Valeur exacte de e	
math.ceil(x)	Plus petit entier $\geq x$	
math.floor(x)	Plus grand entier $\leq x$	
random.randint(a,b)	Choisir au hasard un nombre entre a et b	
random.uniform(0,1)	Choisir au hasard un nombre entre 0 et 1	
X==y	Teste si x est égal à y	
X!=y	Teste si X est différent de y	
X+=1	Incrémenter de 1 la valeur de x	
a>=2 and a<=8	Teste si a est dans [2 ;8]	
a<2 or a>8	Teste si a<2 ou a>8	

**Exemple pour la fonction factorielle en python :**

De façon classique

```
def fact(n):
if n>0 :
    result=1 ;
    for i in range (1,n+1) :
        result=result*i ;
return result ;
else:
    return 1 ;
```

De façon récursive

```
def fact(n):
if n>0 :
    return n*fact(n-1);
else:
    return 1
```

**D'autres instructions intéressantes :**

**x in [a,b,c,d]**

Teste si x est un des  
nombres de la chaîne  
[a,b,c,d]

**x\*\*y**

calcule x à la puissance y.