Code en Python

Les modules

Pour pouvoir utiliser des fonctions mathématiques comme mettre au carré, calculer une racine carrée, etc, il faut faire appel à la bibliothèque **math**. Sans importer la bibliothèque math vous ne pourrez pas travailler correctement. Pour les fonctions aléatoires il faut importer la bibliothèque **random** et pour les calculs sur les fractions, la bibliothèque **fractions**.
▷ la bibliothèque **math** possède plusieurs fonctions et constantes mathématiques usuelles.
▷ la bibliothèque **fractions** possède le nécessaire pour manipuler des fractions, parfois utiles pour la précision des calculs.
▷ la bibliothèque **random** permet de générer des nombres pseudo-aléatoires.

Au début du programme, inscrire :
▷ **from math import *** : Si vous voulez importer toutes les fonctions et constantes mathématiques.
▷ **from fractions import *,** : Si vous voulez importer toutes les fonctions sur les fractions.
▷ **from random import *** : Si vous voulez importer toutes les fonctions aléatoires.

Il y a plein d'autres bibbliothèques que vous pouvez utiliser. Vous pouvez même créer votre propre bibliothèque avec des fonctions que vous utilisez souvent.
Principales bibliothèques :

sys : fonctions systèmes.
os : fonctions permettant d'interagir avec le système d'exploitation.
time : fonctions permettant de travailler avec le temps.
calendar : fonctions de calendrier.
profile : fonctions permettant d'analyser l'exécution des fonctions.
urllib2 : fonctions permettant de récupérer des informations sur internet.
re : fonctions permettant de travailler sur des expressions régulières.
turtle : fonctions permettant de réaliser des dessins géométriques.
numpy : fonctions permettant de faire du calcul scientifique.
sympy : fonctions permettant de faire du calcul formel.
matplotlib : fonctions permettant de faire des graphiques en tout genre.

Instructions les plus utilisées en seconde.

NomVariable=input() : Permet de lire la valeur d'une variable et de la considérer comme du texte.
NomVariable=int(input()) : Permet de lire la valeur d'une variable et de la considérer comme un nombre entier.
NomVariable=float(input()) : Permet de lire la valeur d'une variable et de la considérer comme un nombre réel à virgule.
NomVariable=int(input("Donner votre valeur : ")) : Affiche à l'écran "Donner votre valeur" et permet de lire la valeur de cette variable et de la considérer comme un nombre entier.

print("La valeur du résultat est : ",Resultat) : Affiche à l'écran "La valeur du résultat est :" puis affiche la valeur de la variable Resultat.
Resultat=4 : Affecte à la variable Resultat la valeur 4.
Resultat=Resultat+4 : Affecte à la variable Resultat la valeur précédemment dans Resultat et lui ajoute 4.

x=pi : Affecte à la variable x la valeur π.
x=pow(a,3) ou **x=math.pow(a,3)** : Affecte à la variable x la valeur de a^3.
x=a%b : Affecte à la variable x la valeur du reste de la division euclidienne de a par b.
x=a//b : Affecte à la variable x la valeur du quotient de la division euclidienne de a par b.
x=sqrt(a) ou **x=math.sqrt(a)** : Affecte à la variable x la valeur de $\sqrt{a}$.
x=fabs(a) ou **x=math.fabs(a)** : Affecte à la variable x la valeur de la valeur absolue de a.
x=ceil(a) ou **x=math.ceil(a)** : Affecte à la variable x la valeur du plus petit entier supérieur ou égal à a.

x==y : Permet de tester si x est égal à y.
x!=y : Permet de tester si $x \neq y$.
x+=1 ou **x=x+1** : Incrémente la valeur de x de 1.
a>=2 and a<=8 : Permet de tester $a \geq 2$ ET $b \leq 8$.
a<2 or a>8 : Permet de tester $a < 2$ OU $b > 8$.

random() : Permet de choisir un nombre réel aléatoirement entre les bornes 0 et 1.
uniform(a,b) : Permet de choisir un nombre réel aléatoirement entre les bornes a et b.
randint(a,b) : Permet de choisir un nombre entier aléatoirement entre les bornes a et b.

▷ **Instructions conditionnelles :**
Le squelette est le suivant : Attention il faut bien être rigoureux sur les " :" car c'est eux

qui définissent les décalages à la ligne suivante, ce qui permet de ne pas écrire de "fin du si" en Python.

```
if conditions:
    instruction 1
    instruction 2
    ... etc
else:
    instruction 3
    instruction 4
    ... etc
```

▷ **Les boucles "Pour i allant de a à b" :**
Le squelette est le suivant : Attention il faut bien être rigoureux sur les " :" car c'est eux qui définissent les décalages à la ligne suivante, ce qui permet de ne pas écrire de "fin du pour" en Python.

```
for i in range (a,b+1):
    instruction 1
    instruction 2
    ... etc
```

▷ **Les boucles "Tant que ....." :**
Le squelette est le suivant : Attention il faut bien être rigoureux sur les " :" car c'est eux qui définissent les décalage à la ligne suivante, ce qui permet de ne pas écrire de "fin du tant que" en Python.

```
while conditions:
    instruction 1
    instruction 2
    ... etc
```

Fiche Outils : Programmer (Python)

Significations	Algorithme	Commande Python
Saisir une valeur après demande(ou sans) et affecte à X cette valeur sous forme d'une **chaîne**, ou d'un **entier** ou d'un **décimal**.	Saisir X Ou Lire la valeur de X	**X=input() ;** **X=int(input()) ;** **X=float(input()) ;**
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	**print("X=",X) ;**
La valeur a^2+1 va se stocker dans la mémoire de la variable X	X ← a^2+1	**X=a**2+1 ;**
Si les conditions sont vraies alors on exécute instructions1 sinon on exécute instructions2	Si ... Alors ... Sinon ...	**if** conditions : instructions1 **else :** instructions2
On veut faire n fois les mêmes instructions	Pour i allant de 1 à n Faire ...	**for i in range (1,n+1) :** instructions
Tant que des conditions sont vraies, répéter les instructions	Tant que ... Faire ...	**while** conditions : instuctions
Ecrire une fonction	**Def NomFonction(**paramètres**) :** instructions **Return** ValeurReponse	

Quelques instructions importantes
Ajouter
from math import * (Pour avoir les fonctions mathématiques)
from random import * (Pour avoir les fonctions de hasard)
au début de votre programme.

math.pi	Valeur approchée de pi plus précise que 3,14
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 approchée de e
math.ceil(x)	Plus petit entier >= x
math.floor(x)	Plus grand entier <= 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 chaine [a,b,c,d]

xy** ou **pow(x,y)**

calcule x à la puissance y.