Python pour le lycée

Affectations et saisies :

a=5	Affecte la valeur 5 à la variable a.
a,b=5.1,6.5	Affecte la valeur 5,1 à la variable a et affecte la valeur 6,5 à la variable b.
a=input(" donner la valeur de a : ")	Saisir la chaîne de caractères a.
a=int(input("Donner la valeur de a :"))	Saisir la valeur du nombre entier a.
a=float(input("Donner la valeur de a :"))	Saisir la valeur du nombre décimal a.
def NomDeFonction(a) :	Permet de définir une fonction dont la valeur de l'argument a sera donnée lors de la saisie de l'appel de la fonction.

Affichage :

print(a)	Affiche à l'écran la valeur de la variable a.
return a	Cette instruction permet de renvoyer dans une fonction la valeur d'une variable donnée.
print("Bonjour")	Affiche le texte "Bonjour".
print(" La valeur de a est : ",a)	Affiche à l'écran la phrase " La valeur de a est : " puis la valeur de la variable a.

Calculs :

x+y ; x-y ; x*y ; x/y	
x**y	X puissance y.
abs(x)	Valeur absolue de x.
x//y	Quotient de la division de x par y.
X%y	Reste de la division de x par y.

Pour ces instructions il faut déclarer l'utilisation de la bibliothèque « math » en notant au début de votre code : **from math import***

sqrt(a)	Racine carrée de a.
floor(a)	Partie entière de a.
ceil(a)	Arrondir à l'entier supérieur de a.
round(a,i)	Arrondir a à i chiffres près.
pi	Le nombre pi.
cos(x), sin(x) et tan(x)	x est en radiants.
acos(x) , asin(x), atan(x)	Renvoie l'angle en radians.
radians(x)	Convertir de degrés en radians.
degrees(x)	Convertir de radians en degrés.
pow(a,n)	a puissance n.
eval(a) a=eval(input("Donner la valeur de a :"))	Permet de transformer une chaîne en instruction.
exp(a)	Exponentielle de a.
log(a)	Logarithme népérien de a.
log10(a)	Logarithme en base 10 de a.
comb(n,k)	Combinaison
perm(n,k)	Arrangement
factorial(n)	Factorielle de n.

La bibliothèque **cmath** permet de travailler sur les nombres complexes.

a=complexe(3,4) ou a=3+4*j	Définir un complexe.
a.real et a.imag	Partie réelle et imaginaire.
phase(a)	Argument de a.
polar(a)	Coordonnées polaires de a.
rect(r,phi)	Ecriture algébrique.

La bibliothèque **fractions** permet de travailler sur les fractions.

a=Fraction(4,5)	Définir une fraction.
numerator(a)	Numérateur
denominator(a)	Dénominateur

Les listes :

L=[]	Créer une liste vide.
L=[1,3,7,8]	Créer une liste.
L.clear	Supprime tous les éléments de la liste.
L.insert(i,a)	Insère un élément à la position i.
L.extend(M)	Ajoute à la liste L la liste M.
L.reverse()	Inverse les éléments de la liste.
L.copy()	Renvoie une copie de la liste L.
L.append(a)	Ajouter un élément à la fin d'une liste.
L.remove(a)	Retirer un élément.
L.pop()	Enlève le dernier élément et le renvoie.
L.pop([i])	Enlève l'élément de position i et le renvoie.
len(L)	Longueur de la liste.
L.sort()	Trie la liste dans l'ordre croissant.
n=L.count(a)	Compte le nombre d'apparition de a.
n=choice(L)	Tirage au sort d'un élément dans la liste.
if n in L	Savoir si un élément est dans L.
L.index(a)	Trouver l'index d'un élément.
min(L)	Minimum de la liste.
max(L)	Maximum de la liste.
del L[i]	Retire l'élément d'index i.
del L[-1]	Retire le dernier élément.
L[:n]	Retourne les n premiers éléments.
L[n :]	Enlève les n premiers éléments.
L[a :b]	Retourne les éléments d'indice a à b-1.
sum(L)	Somme des éléments de la liste.
L=[i for i in range(n)]	Créer une liste des n premiers entiers naturels.
L=[i**2 for i in range(n)]	Créer la liste des n premiers carrés.
L=[i**2 for i in M]	Créer la liste des carrés des éléments de M.
L=[f(i) for i in M if i !=b]	Créer le liste des valeurs d'une fonction à partir d'une liste M avec condition.

La bibliothèque **turtle** permet de tracer des figures avec le module tortue.

reset()	Effacer le dessin.
Goto(x,y)	Aller aux coordonnées (x,y).
forward(a)	Avancer de a pixels.
backward(a)	Reculer de a pixels.
up()	Relever le crayon.
down()	Abaisser le crayon.
color("red ")	Choisir la couleur.
left(a)	Tourner vers la gauche d'un angle a en degrés.
right(a)	Tourner vers la droite d'un angle a en degrés.
mainloop()	Terminer le programme.

La bibliothèque **random** permet de travailler sur des nombres aléatoires.

random()	Nombre réel aléatoire dans [0 ;1[
randint(a,b)	Nombre entier aléatoire entre a et b.
uniform(a,b)	Nombre réel aléatoire entre a et b.
L=[randint(a,b) for i in range(n)]	Créer une liste de n entiers aléatoires entre a et b.

Instructions conditionnelles

If conditions : Instructions elif : Instructiions else : instrustions	Exemples de conditions : **A<=B** ou **A>B** ou **A>= B** ou **A>B** Test d'égalité : **A==B** Test de différence : **A !=B** On peut aussi mettre plusieurs instructions en utilisant **and** ou **or** ou **not.**

Les boucles bornées

for i in range(n+1) : instructions	Pour i allant de 0 à n répéter les instructions.
for i in range (a,n+1) : instructions	Pour i allant de a à n répéter les instructions.
for i in range (a,n+1,t) : instructions	Pour i allant de a à n avec un pas de t répéter les instructions.
for i in L : instructions	Pour i allant de L[0] à L[len(L)] répéter les instructions.

Les boucles non bornées

while conditions : instructions	Tant que la condition est vérifiée faire les instructions. Exemples de conditions : **A<=B** ou **A>B** ou **A>= B** ou **A>B** Test d'égalité : **A==B** Test de différence : **A !=B** On peut aussi mettre plusieurs instructions en utilisant **and** ou **or** ou **not.**

La bibliothèque **matplotlib** permet de travailler sur des représentations graphiques.

plt.show()	Afficher le graphique.
Plt.axis([0,10,0,10])	Affiche les axes[Abscisses, Ordonnées]
x=np.linspace(-2,2,15) plt.plot(x,f(x))	Créer une liste de 15 valeurs entre -2 et 2 puis représente la fonction des images de ses valeurs.
plt.plot(L,M)	Représente le graphique avec les valeurs de L en abscisses et celles de M en ordonnées.
plt.plot(L,M,'o')	Représente un nuage de points de forme 'o' avec L en abscisses et M en ordonnées. On peut remplacer 'o' par '-' ou par '^' pour avoir des tirets ou des triangles.
plt.plot(L,M,'r') plt.plot(L,M,'ro')	Trace le graphique en rouge. Trace le nuage de points avec de 'o rouges.
plt.pause(1)	Faire une pause de 1s pour animer.
plt.xlabel('…') plt.xlabel('…')	Indiquer une légende sur l'axe des abscisses et sur l'axe des ordonnées
plt.plot(L,M,label=''f(x)'') plt.legend()	Affiche une légende sur le graphique.
plt.grid()	Affiche le quadrillage.

La bibliothèque **sympy** permet de travailler sur du calcul formel (calcul littéral).

x=symbol('x')	Déclarer une variable.
expr=x**2 diff(expr,x)	Dériver.
integrate(expr,x)	Primitive.
integrate(expr, (x,a,b))	Integrale entre a et b.
series(expr,x,0,8)	Développement en 0 à l'ordre 8.
solve(expr,x)	Résoudre dans C, expr=0
expand(expr)	Développer.
factor(expr)	Factoriser.
limit(expr,x,oo)	Limite de expr en + infinie.
limit(expr,x,-oo)	Limite de expr en – infinie.
limit(expr,x,a)	Limite de expr en a.
limit(expr,x,a,'+')	Limite de expr en a+.
x,y=symbols('xy') solve([2x+y-2,-3x-5y-1],[x,y])	Résoudre 2x+y=2 et -3x-5y=1.

La bibliothèque **NumPy** permet de travailler sur des matrices.

M=np.array([..],[..],[..], …,[..])	Créer une matrice.
dot(a,b)	Produit de deux matrices a et b.
2.+M	Ajoute 2 à tous les coefficients de M.
2*M	Multiplie par 2 les coefficients de M.
M+N	Ajoute deux matrices de mêmes dimensions.
vdot(u,v)	Produit scalaire de u et v.
M[i,j]	Elément de M à la ligne i et colonne j.
zeros(n) ou ((n,p))	Vecteur nul ou matrice nulle
eye(n) ou ((n,p))	Matrice unité de taille n ou (n,p) avec 1 sur diagonale.
ones(n) ou ((n,p))	Vecteur ou matrice avec que des 1.
diag(v)	Matrice dont la diagonale est le vecteur v.
random.rand(n) ou ((n,p))	Vecteur ou matrice dont les coefficient sont des nombres aléatoires uniforme sur [0,1].

Lecture de fichiers :

fichier=open(''texte.txt'', ''r '') fichier=open(''texte.txt'', 'w '') fichier=open(''texte.txt'', ''a '') fichier=open(''texte.txt'', ''t '')	Ouvrir un fichier en mode lecture. Ouvrir un fichier en mode écriture. Ouvrir un fichier et se positionne à la fin. ouvrir un fichier pour mise à jour (lecture et écriture)
Texte=fichier.read()	Lit le fichier et créer une liste d'éléments.
fichier.write(''texte \n '')	Ecriture dans un fichier.
f.seek(0)	Positionner le curseur au début.
with open("etudiants.txt", encoding='utf-8') as f: for ligne in f: print(ligne)	Lire les lignes d'un fichier puis les afficher à l'écran.
split() texte.split('\n') texte.split(',')	Permet de transformer une chaîne en liste. Délimiteur de fin de ligne. Mettre une virgule dans un fichier csv.
f.close()	Fermer le fichier.
f.readlines()	Lit les lignes du fichier.

La bibliothèque **csv** permet de travailler sur des fichiers csv.

fichier=open(''Fichier.csv'','a', newline	Ouvrir un fichier csv et se positionne à la fin.
FichierCSV=csv.writer(fichier,délimiter='','')	Déclarer un fichier csv
FichierCSV.writerow(Liste)	Ecrire la liste à la fin du fichier.