MEMENTO PYTHON

LES INSTRUCTIONS CONDITIONNELLES ET LES BOUCLES

```
if conditions1 :
        instructions1
elif conditions2 :
        instructions2
else :
        instructions3
```

elif est la contraction de else if

a<10 renvoie *True* si a<10, *False* sinon
a>10 renvoie *True* si a>10, *False* sinon
a<=10 renvoie *True* si a $\leqslant$ 10, *False* sinon
a>=10 renvoie *True* si a $\geqslant$ 10, *False* sinon

```
for variable in seq :
        instructions
```

seq est une liste
(voir la fonction range dans OPÉRATIONS SUR LES LISTES)

```
while conditions :
        instructions
```

while signifie tant que
Dès que les conditions ne sont plus vraies, la boucle s'arrête

LES FONCTIONS (PYTHON)

```
def joli(prenom) :
        txt=prenom+' est joli(e)'
        return txt
```

joli('johan') renverra la phrase 'johan est joli(e)'
joli('amelie') renverra la phrase 'amelie est joli(e)'

```
def g(x) :
        if x<3 :
                return 4x+5
        else :
                return x+2
```

Cette fonction renvoie l'image d'un nombre par la fonction g définie par $g(x) = \begin{cases} 4x+5 \text{ si } x<3 \\ x+2 \text{ si } x \geqslant 3 \end{cases}$.
g(2) renverra donc 13 et g(4) renverra 6

LES BIBLIOTHÈQUES DE PYTHON

from *bibliotheque* import *fonction*	importe la *fonction*
import *bibliotheque* as *nom*	importe la *bibliotheque* et la nomme *nom* il suffit alors d'écrire *nom.fonction*
help('*bibliotheque*')	affiche l'aide de *bibliothèque*
from *bibliothèque* import *	importe toutes les fonctions de *bibliothèque* mais ceci est déconseillé car certaines fonctions sont communes à plusieurs bibliothèques et ne donnent pas les mêmes résultats, comme *pow*

VARIABLES ET OPÉRATIONS

bibliothèque	fonction	explications
	input() input(message)	Affiche message et invite l'utilisateur à rentrer une valeur
	print(objet)	Affiche objet print(2+3) affiche 5 print("Hello world !") affiche la chaîne Hello world !
	a*b	Multiplication de a par b 2*3 renvoie 6 2*1/2 renvoie 1.0
	x**y ou pow(x,y)	Renvoie x^y 2**3 renvoie 8 2**0,5 renvoie 1.4142135623730951 2**60 renvoie 1152921504606846976
	pow(x,y,m)	Renvoie x^y modulo m pow(2,10,3) renvoie 1 car 2^{10}=1 024=3x341+1
math	pow(x,y,m)	Renvoie x^y (résultat de type float) pow(2,3) renvoie 8.0 pow(2,60) renvoie 1.152921504606847e+18 pow(2,10,3) n'existe pas : il faut alors écrire pow(2,10)%3 pour obtenir 1.0
	a/b	Quotient de a par b 1/3 renvoie 0.3333333333333333 4/2 renvoie 2.0
	a//b	Quotient de la division euclidienne de a par b
	a%b	Reste de la division euclidienne de a par b 14%3 renvoie 2 car 14=3x4+2 14%1.1 renvoie 0.7999999999999989 (au lieu de 0.8) 14%1.4 renvoie 8.881784197001252e-16 (au lieu de 0)
math	sqrt(x)	$\sqrt{x}$ sqrt(2) renvoie 1.4142135623730951
	max(a,b,c)	Maximum de (a,b,c)
	min(a,b,c)	Minimum de (a,b,c)
	eval(expression)	eval("a") renvoie la valeur contenue dans la variable a
math	isnan(x)	Renvoie *True* si x n'est pas un nombre, et *False* sinon
	int(x)	Troncature de x int(12.958) renvoie l'entier 12 int(-2.57) renvoie l'entier -2
	int(chaîne)	Convertir chaîne en entier int("54") renvoie 54
	float(chaîne)	Convertir chaîne en nombre float("54.27") renvoie 54.27

bibliothèque	fonction	explications
	round(x)	Arrondi entier de x round(12.958) renvoie l'entier 13 round(-2.57) renvoie l'entier -3
	round(x,nb)	Arrondi de x avec une précision de nb décimales round(12.958,2) renvoie 12.96
	abs(x)	Valeur absolue de x
	str(x)	Renvoie une chaîne formée avec le nombre x str(25) renvoie "25"
	isinstance(objet,class)	Renvoie *True* si objet est du type class, et *False* sinon. isinstance(2,int) renvoie *True* car 2 est un entier isinstance(2.54,float) renvoie *True* car 2.54 est un réel
math	pi	La constante π à la précision disponible
math	ceil(x)	Arrondi supérieur de x (ceil = plafond) ceil(12,958) renvoie l'entier 13 ceil(-2,57) renvoie l'entier -2
math	floor(x)	Arrondi inférieur de x (floor = plancher) floor(12,958) renvoie l'entier 12 floor(-2,57) renvoie l'entier -3
math	cos(x), sin(x), tan(x)	Fonctions trigonométriques (mesures en radians)
math	degrees(x)	Convertit l'angle x de radians en degrés
math	radians(x)	Convertit l'angle x de degrés en radians
math	exp(x), ln(x)	Fonctions exp et ln
math	factorial(x)	Factorielle x!
math	log(x) ou log(x,e)	Renvoie ln(x)
random	randint(a,b)	Entier pseudo-aléatoire compris entre a et b inclus
random	random()	Nombre (float) pseudo-aléatoire dans $[0;1[$
random	uniform(a,b)	Renvoie un nombre compris entre a et b inclus
	lambda car : exp	Crée la fonction qui à car associe exp f=lambda x : x**2 crée la fonction f qui à x associe son carré et donc f(3) renvoie 9

LES LISTES

bibliothèque	fonction	explications
	[]	Crée une liste vide
	seq*nb	Agrandit la liste seq avec elle-même [0,1]*3 renvoie [0,1,0,1,0,1] [0]*5 renvoie [0,0,0,0,0]
	seq[i]	Renvoie le i-ème élément de la liste seq Si l=['a','b','d','c'] alors l[1] renvoie 'b' Si l='mathematiques' alors l[5] renvoie 'm'
	seq[i:j]	Renvoie la liste contenant les éléments de la liste seq entre la position i (inclus) et la position j (exclu) Si l=['a','b','d','c'] alors l[1:3] renvoie ['b','d'] Si l='mathematiques' alors l[1:5] renvoie 'athe'
	seq[:]	Crée un nouvelle liste identique à la liste seq Pour copier une liste seq, il faut donc écrire seq2=seq[:]
	seq[-i:]	Renvoie les i derniers éléments de la liste seq Si l=['a','b','d','c'] alors l[-2:] renvoie ['d','c'] Si l='mathematiques' alors l[-3:] renvoie 'ues'
	seq[:i]	Renvoie les i premiers éléments de la liste seq Si l=['a','b','d','c'] alors l[:2] renvoie ['a','b'] Si l='mathematiques' alors l[:3] renvoie 'mat'
numpy	multiply(seq1,nb)	Multiplie la liste seq par le nombre nb
	len(seq)	Renvoie le nombre d'éléments de la liste seq
	range(a)	Renvoie la liste de 0 à $a-1$ range(3) renvoie [0,1,2]
	range(a,b)	Renvoie la liste de a à $b-1$ range(3,6) renvoie [3,4,5]
	range(a,b,c)	Renvoie la liste de a à $b-1$ avec un pas de c range(1,10,2) renvoie [2,4,6,8,10]
xrange(a) peut être utilisé à la place de range(a) si a est trop grand (voir https://docs.python.org/2/library/functions.html#xrange)		
	seq.append(elt)	Rajoute l'élément elt à la liste seq Si l=['a','b','d','c'] alors l.append['e'] renvoie ['a','b','d','c','e']
numpy	append(seq1,seq2)	Renvoie la liste formée de seq1 suivie de seq2 append([0,1],[3,4]) renvoie [0,1,3,4]
	sorted(seq)	Renvoie la liste seq triée dans l'ordre croissant (aussi bien pour des nombres que des chaînes) sorted([1,25,8,12]) renvoie [1,8,12,25] sorted(['a','d','c']) renvoie ['a','c','d'] sorted(['1','25','8','12']) renvoie ['1','12','25','8']
	sorted(seq,reverse=True)	Renvoie une nouvelle liste seq triée dans l'ordre décroissant
	seq.reverse()	Trie la liste seq dans l'ordre décroissant
	sum(seq)	Renvoie la somme des valeurs de la liste seq
	seq.count(car)	Renvoie le nombre d'occurrences de car dans la liste seq

LES LISTES (SUITE)

bibliothèque	fonction	explications
	enumerate(seq,start)	enumerate(["mauvais","moyen","bon"],0) renvoie la liste [(0,"mauvais"),(1,"moyen"),(2,"bon")]
	val in seq	Renvoie *True* si la valeur val est dans la liste seq, *False* sinon
	del seq[pos]	Dans la liste seq, supprimer l'item à la position pos
numpy	delete(seq,pos)	Renvoie une nouvelle liste seq sans la valeur à la position pos delete([0,1,2,3],1) renvoie [0,2,3]
numpy	insert(seq,elt,pos)	Liste seq dans laquelle on a rajouté elt en position pos insert([5,6,8,9],1,4) renvoie [5,4,6,8,9]
numpy	add(seq1,seq2)	Additionne les deux listes seq1 et seq2 (de même taille) add([1,5],[2,6]) renvoie [3,11]
numpy	unique(seq)	Renvoie la liste seq sans aucune répétition unique([1,2,1,3,1,4]) renvoie [1,2,3,4]
numpy	linspace(start,stop,num)	Renvoie la liste de start à stop, avec num nombres linspace(2,3,5) renvoie [2.,2.25,2.5,2.75,3.] linspace(2,3,5,endpoint=False) renvoie [2.,2.2,2.4,2.6,2.8]
random	choice(seq)	Renvoie un élément aléatoire de la liste seq choice([1,2,3]) renvoie 1, 2 ou 3
random	shuffle(seq)	Renvoie une liste formée par un pseudo-mélange de la liste seq
random	sample(seq,k)	Renvoie une liste pseudo-aléatoire de longueur k à partir de la liste seq

CHAÎNES DE CARACTÈRES

bibliothèque	fonction	explications
	str.lower()	Met la chaîne str en minuscules
	str.upper()	Met la chaîne str en majuscules
	str.replace(car1,car2)	Dans la chaîne str, remplace le caractère car1 par car2
	str.split(car)	Renvoie une liste contenant les mots de la chaîne str séparés par le caractère car "vive les maths".split('e') renvoie ['viv', ' l', 's maths'] "vive les maths".split(' ') renvoie ['vive', ' les', 'maths']
	str.count(car)	Nombre de fois que la chaîne str contient le caractère car "mathematiques".count('a') renvoie 2
	str.find(car)	Renvoie la position dans la chaîne str où se trouve le caractère car (et renvoie -1 si car n'apparaît pas) "mathematiques".find('a') renvoie 1

```
for car in chaine :
    instructions
```

Fait prendre à la variable car toutes les valeurs des caractères de la chaîne chaine

```
liste = ["a","d","m"]
for val in enumerate(liste):
    print val
```

Affiche :
(0, 'a')
(1, 'd')
(2, 'm')

GRAPHIQUES

Principales fonctions

bibliothèque	fonction	explications
matplotlib.pyplot	axis[v1,v2,v3,v4]	Définit les dimensions du repère
matplotlib.pyplot	grid()	Affiche le quadrillage
matplotlib.pyplot	plot(seq1,seq2)	seq1 est la liste des abscisses seq2 est la liste des ordonnées plot(seq1,seq2) trace les points de coordonnées associées
matplotlib.pyplot	show()	Permet l'affichage de la courbe

Un exemple :

```
import matplotlib.pyplot as plt
from numpy import linspace

x_list,y_list=[],[]
for x in linspace(-10,10,100000):
    x_list.append(x)
    y_list.append(x**2-3*x+x-1)

plt.plot(x_list,y_list)
plt.grid()
plt.savefig("fonction_poly.png")
plt.show()
```