

PYTHON

<code>from datetime import datetime</code>	Importe seulement datetime de la librairie datetime
<code>from math import *</code>	Librairie math complète
<code>from math import sqrt</code>	Seulement sqrt de la librairie math
<code>from random import randint</code>	Randint de la librairie random
<code>from itertools import *</code>	Permet de lister les permutations avec : <code>list(permutations("0123456789"))[999999]</code> avec 999999 pour 1 million de permutations
<code>Import re</code>	Pour les expressions régulières

<code>def NOM(reponse):</code> <code> return reponse</code>	Cree une fonction du nom NOM en prenant la valeur de reponse Retourne la valeur de reponse
<code>NOM(reponse)</code>	Appel la fonction NOM en lui envoyant la valeur reponse
<code>def le_plus_grand(*args):</code> <code>le_plus_grand(-10, -5, 5, 10)</code>	<code>*args</code> = TOUT les arguments reçut de <code>le_plus_grand</code>
<code>try:</code> <code> entier = int(input("entier :"))</code> <code> print ("vous avez entré un entier")</code> <code>except:</code> <code> print ("vous n'avez pas entre un entier")</code>	Lance le code TRY, si il y a une erreur, lance le code EXCEPT a la place
<code>Print (« message »)</code> <code>Print (« message », « message », variable)</code>	Ecrit message dans la console Permet de metre des variable et messages a la suite
<code>print ma_variable</code>	Ecrit la valeur de <code>ma_variable</code>
<code>Print (len(string))</code>	Marque le nombre de lettre/taille de la variable string
<code>print (" ".join(animaux))</code> <code>' ; '.join(['a', 'b', 'c'])</code>	Enlevé les guillemets et crocher de la liste animaux en mettant des espaces entre chaque valeur Rassemble les valeurs en une liste avec des ; entre chaque
<code>print (0b1)</code> <code>print (0b10)</code>	0b1 = binaire = 1 0b10 = binaire = 2
<code>.lower()</code>	Met les lettres en minuscule
<code>.upper()</code>	Met les lettres en majuscule
<code>ma_variable = 10</code>	Cree une variable ma_variable de valeur 10
<code>nombres = [5, 6, 7, 8]</code>	Cree un tableau nombre de 4 valeurs
<code>nombre = [[0]*15 for i in range(15)]</code>	Cree un tableau nombre a 2dimensions de 15*15 rempli de 0
<code>liste = entete.split(";")</code>	Les valeurs avec ; de la liste sont séparé et mis dans une liste
<code>'hello crazy world!'.replace("o", "u")</code>	Remplace les "o" par des "u"
<code>Nombre = range(6)</code> <code>Nombre = range(1,6)</code>	Equivalent a nombres [0,1,2,3,4,5] Equivalent a nombres [1,2,3,4,5]
<code>nombre.append(20)</code>	Ajoute une case avec la valeur 20 dans nombre
<code>slice = nombre[1:3]</code>	Slice prend les valeurs 1 et 2 du tableau nombre (donc 6 et 7)
<code>animaux.index("ratel")</code>	Renvoie la position(index) de ratel du tableau animaux
<code>animaux.remove("ratel")</code>	Supprime ratel du tableau animaux

<code>animaux.pop(1)</code>	Supprime le 2eme élément du tableau <code>animaux</code> et renvoie cette valeur avant de supprimer
<code>del(animaux[1])</code>	Supprime le 2eme élément du tableau <code>animaux</code>
<code>animaux.insert(1, "chien")</code>	Met à la position 1 le mot <code>chien</code> dans le tableau <code>animaux</code>
<code>animaux.sort()</code>	Met le tableau <code>animaux</code> dans l'ordre alphabétique
<code>liste = sorted(liste)</code>	Trie dans un ordre croissant
<code>str(ma_variable)</code>	Change <code>ma_variable</code> en string
<code>int(ma_variable)</code>	Transforme <code>ma_variable</code> en entier
<code>int("10",2)</code>	Prévient que l'entrée est du binaire (base 2) et le converti en base 10 (entier)
<code>mon_bool = True</code>	Cree un <code>booléen</code> du nom de <code>mon_bool</code>
<code>type(4.02)</code>	Renvoie le type de 4.02 (ici float)
<code>nom = input("Quel est votre nom ?")</code> <code>chiffre = int(input("Quel est votre nom ?"))</code>	Affiche « quel est votre nom ? » et demande à l'utilisateur de saisir la valeur de <code>nom</code> Pour un chiffre
<code>Maintenant = Datetime.now()</code> <code>Annee_actuelle = Maintenant.year</code> <code>mois_actuelle = Maintenant.month</code> <code>jour_actuelle = Maintenant.day</code> <code>heure_actuelle = Maintenant.hour</code> <code>minutes_actuelle = Maintenant.minute</code> <code>secondes_actuelle = Maintenant.second</code>	<code>Maintenant</code> prend la Date et heure actuelle Année Mois Jour Heure Minute secondes
<code>if reponse == "Left":</code> <code>elif reponse < 5:</code> <code>else:</code>	Si <code>reponse</code> est <code>Left</code> Sinon si <code>reponse < 5</code> sinon
<code>for nombre in ma_liste:</code>	Pour <code>nombre</code> dans <code>ma_liste</code>
<code>for a, b in zip(liste_a, liste_b):</code>	Pour <code>a</code> et <code>b</code> dans <code>liste_a</code> et <code>liste_b</code> (s'arrête à la fin du plus court)
<code>for nombre in range(len(ma_liste))</code>	<code>Nombre</code> prend la valeur du premier <code>nombre</code> , puis du deuxième... jusqu'à que <code>ma_liste</code> ne contienne plus de <code>nombre</code>
<code>for index, element in enumerate(choix):</code>	<code>Index</code> prend l'emplacement et <code>element</code> la valeur dans <code>choix</code>
<code>While nimporte == 9:</code> <code>While nimporte:</code>	Tant que <code>nimporte</code> inférieur à 9 Tant que <code>nimporte</code> est <code>True</code>
<code>break</code>	Termine la boucle sans la continuer
<code>and</code> <code>or</code> <code>not</code>	ET OU NON
<code>Variable.isalpha()</code>	Renvoie vrai s'il n'y a que des lettres Faux si il y a au moins un chiffre
<code>sqrt(25)</code>	Racine carrée de 25
<code>dir(math)</code>	Montre toutes les fonctions de la librairie <code>math</code>
<code>max(args)</code> <code>min(args)</code> <code>abs(arg)</code>	Maximum de... Minimum de... Absolue de...
<code>residents = {'Puffin' : 104, 'Paresseux' : 105, 'Python'}</code>	Affectez un dictionnaire avec trois paires de valeur-clé à <code>residents</code>

: 106}	
print residents['Puffin']	Renvoie la valeur de puffin (ici : 104)
menu['Poulet Alfredo'] = 14.50	Ajoute a menu, la paire Poulet Alfredo : 14,50
menu['Poulet Alfredo'] = 1.50	Modifier Poulet Alfredo : 14,50 par Poulet Alfredo : 1,50
del menu['Poulet Alfredo']	Supprime Poulet Alfredo du menu
menu.items() menu.keys() menu.values()	Affiche le dictionnaire en montrant chaque paire Montre seulement les clés Montre seulement les valeurs
pairs_a_50 = [i for i in range(51) if i % 2 == 0]	Pairs_a_50 a pour valeur seulement les chiffres paires de 1 a 50 : pour i dans range(51) condition i%2 ==2
Liste = autre_liste[debut:fin:pas]	Parcourt autre_liste. Si le pas est negatif, commence par la fin
filter(lambda x: x % 3 == 0, ma_liste)	Renvoie tout les chiffres divisible par 3 de ma_liste
randint(1,10)	Choisi un nombre aléatoire entre 1 et 10
print 5 >> 4 # Décalage à droite	Décale tous les 1 binaire de 5 de 4 emplacement vers la droite
print 5 << 1 # Décalage à gauche	Décale tous les 1 binaire de 5 de 1 emplacement vers la gauche
print 8 & 5 # ET binaire	Ne laisse a 1 seulement si la position du 1 du 8 et du 5 sont les même
print 9 4 # OU binaire	Laisse la position des 1 de 9 et de 4
print 12 ^ 42 # OU EXCLUSIF binaire	Laisse la position des 1 de 9 et de 4 sauf si les 2 sont de même position
print ~88 # NON binaire	Ajoute 1 et le met négatif
bin(1)	= 0b1
class Animal(object): est_vivant = True def __init__(self, nom,age): self.nom = nom self.age = age def description(self): print self.nom print self.age zebre = Animal("Jeffrey",18) print zebre.nom print zebre.age print zebre.description()	Cree une classe Animal Toute variable de cette classe auront par default True à l'appel de variable.est_vivant Constructeur (self obligatoire) Ajoute la variable nom Ajoute la variable age Définie une description de paramètre self Affiche le nom Affiche l'age Donne a zebre le nom de Jeffrey et d'age 18 en passant par la classe Animal. Toujours mettre dans l'ordre des variables Affiche le nom de zebre Affiche l'age de zebre Appel le def description pour l'animal zebre
class Triangle(object):	Classe Triangle

<pre>def __init__(self, angle1, angle2, angle3): self.angle1 = angle1 self.angle2 = angle2 self.angle3 = angle3 nombre_de_cotes = 3 def verifier_angles(self): if self.angle1+self.angle2+self.angle3 == 180: return True else: return False class Equilateral(Triangle): angle = 60 def __init__(self): self.angle1 = self. angle self.angle2 = self. angle self.angle3 = self. angle</pre>	<p>Constructeur</p> <p>Variable</p> <p>Variable</p> <p>Variable</p> <p>Nombre_de_cotes = 3</p> <p>Methode verifier_angle</p> <p>Condition somme des 3 angles = 180</p> <p>Returnne True</p> <p>Sinon</p> <p>Retourne False</p> <p>Classe Equilateral avec les parametres de Triangles</p> <p>Angle = 60</p> <p>Angle1 de Triangle = angle</p> <p>Angle2 de Triangle = angle</p> <p>Angle3 de Triangle = angle</p>
mon_fichier = open("output.txt", "r+")	<p>Ouvre le fichier output.txt</p> <p>W = lecture</p> <p>R = écriture</p> <p>R+ = lecture/écriture</p> <p>A = écriture en fin de fichier</p>
mon_fichier.write("Donnée à écrire")	Ecrit donnée à écrire dans mon_fichier
print mon_fichier.read()	Lit mon_fichier complet
print mon_fichier.readline()	Lit une ligne mon_fichier (va a la suivante si on le refait)
mon_fichier.close()	Ferme mon_fichier

Expressions régulières :

Import re	Pour pouvoir utiliser les expressions régulières
re.search(r"abc", "abcdef")	Renvoie True si c'est vrai
r"guitare piano"	Vérifie si il y a guitare OU piano
r"Quoi \?"	Mettre "\" pour tous les metacaractère : r"!^\$()[\]{}?+*.\\
r"^Bonjour"	Bonjour devra obligatoirement être le premier mot de la chaîne
r"zéro\$"	zéro devra obligatoirement être le dernier mot de la chaîne
r"gr[ioa]s"	Recherche gris OU gros OU gras
r"[a-z]"	Une lettre (de A a Z)
r"[0-9]"	Un chiffre (de 0 a 9)
r"[a-z0-9]"	Une lettre OU un chiffre
r"[^0-9]"	Recherche tout sauf un chiffre
r"a?"	Reconnais 0 ou 1 "a" PAS PLUS de 1
r"a+"	Au moins 1 "a" A PARTIR DE 1
r"a*"	0 ou plusieurs a
r"Ay(ay oy)*"	VRAI pour AY et autant de ay ou de oy que l'on veut
r"a{3}"	Fonctionne que pour aaa
r"a{3,5}"	Fonction pour aaa, aaaa, aaaaa seulement
r"a{3,}"	Fonctionne pour autant de a que l'on veut
\d	= [0-9]
\D	= [^0-9]
\w	= [a-zA-Z0-9_]
\W	= [^a-zA-Z0-9_]
\t	Indique une tabulation
\n	Indique une nouvelle ligne
\r	Indique un retour chariot
\s	Indique un espace blanc
\S	Indique ce qui n'est PAS un espace blanc (\t \n \r)
.	Indique n'importe quel caractère. Sauf entrée "\n"
re.sub(r"(ab)", r"\1 ", "abcdef")	Une parenthèse = une variable chiffrer (\1 pour la premier, \2 pour la deuxième...) \0 = toute la REGEX Dans l'ordre : Fais une recherche, Remplace ce qu'il en a entre parenthèse par un autre code, (ici on rajoute un espace au début et à la fin) la partie où l'on fait la recherche (chaîne de caractère)
(?:nsti)	La parenthèse ne comptera pas pour une variable et sera sauté

MODE FENETRE

<code>import pygame</code> <code>from pygame.locals import *</code>	Importe tout pygame
<code>import pygame.mixer</code>	Importe seulement pour le son
<code>pygame.init()</code>	OBLIGATOIRE après l'import
<code>fenetre = pygame.display.set_mode((640, 480), OPTION)</code>	Crée une fenêtre de 640px de large et 480px de haut
Listes OPTIONS RESIZABLE FULLSCREEN	 Permet de changer la taille fenêtre en temps réel Pleine écran
<code>fond = pygame.image.load("background.jpg").convert()</code>	Charge l'image background.jpg dans fond Pour les images avec de la transparence faire convert.alpha()
<code>position_image = image.get_rect()</code>	Crée variable avec les positions de l'image
<code>fenetre.blit(fond, (0,0))</code>	Place l'image fond en position 0,0
<code>pygame.display.flip()</code>	Initialise l'écran (affiche les images)
<code>while continuer:</code> <code>for event in pygame.event.get():</code> <code>if event.type == QUIT:</code> <code>continuer = 0</code> <code>elif event.type == VIDEORESIZE:</code> <code>if event.w > 500 or event.h > 500:</code> <code>continuer = 0</code> <code>elif event.type == MOUSEBUTTONUP:</code> <code>if event.button == 1 and event.pos[1]<100:</code> <code>print ("ok")</code> <code>elif event.type == MOUSEMOTION:</code> <code>if event.buttons[0] == 1:</code> <code>print("mouvement")</code> <code>elif event.type == KEYDOWN:</code> <code>if event.key == K_SPACE:</code> <code>print("Espace")</code> <code>if event.key == K_RETURN:</code> <code>print("Entrée")</code>	 Penser à mettre une variable continuer Attend un event Si cette event est le clic sur la croix Continuer = 0 Si la fenêtre est redimensionné Si fenêtre dépasse 500px*500px Continuer = 0 Si c'est un clic de souris (relachement) Bouton gauche et position y souris < 100 Affiche « ok » S'il y a mouvement souris Si bouton gauche est pressé Afficher « mouvement » Si c'est une touche du clavier Touche espace Affiche « espace » dans la console Touche entrée Affiche « entrée » dans la console
<code>if event.gain == 1 and event.state == 1:</code>	Sortie de la souris de la fenêtre 0 1 Entrée de la souris dans la fenêtre 1 1 Réduction de la fenêtre 0 6 Réouverture de la fenêtre réduite 1 6 Inactivation de la fenêtre 0 2 Réactivation de la fenêtre 1 6
<code>pygame.key.set_repeat(400, 30)</code>	Répétition quand on reste appuyé
<code>event.pos[0]<100</code>	Position x souris < 100



<code>event.pos[1]<100</code>	Position y souris < 100
<code>pygame.time.Clock().tick(30)</code>	Pause de 30ms
<code>son = pygame.mixer.Sound("son.wav")</code>	Crée un son
<code>son.play()</code>	Lance le son
<code>son.stop()</code>	Arrête le son
<code>son.fadeout(300)</code>	Fondu à la fin du son
<code>pygame.mixer.pause()</code>	Met sur pause TOUS les sons
<code>pygame.mixer.unpause()</code>	Enlève la pause de TOUS les sons
<code>pygame.mixer.stop()</code>	Arrête TOUS les sons
<code>volume = pygame.mixer.music.get_volume()</code>	Retourne la valeur du volume, entre 0 et 1
<code>pygame.mixer.music.set_volume(0.5)</code>	Met le volume à 0.5 (moitié)

TKINTER

From tkinter import *	Importe la Library tkinter
from tkinter.messagebox import *	Importe Library d'alerte
from tkinter.filedialog import *	Pour les ouverture de fichiers
fenetre = Tk()	Cree la fenetre
bouton =Button(fenetre , text="Fermer", command= fenetre .quit) bouton .pack()	Cree un bouton du nom bouton , dans fenetre , avec le texte Fermer avec pour commande quitter le fenetre
label = Label(fenetre , text="Texte par défaut", bg="yellow") label .pack()	Cree un label du nom label , dans fenetre , avec le texte « Texte par défaut. » et un surligne jaune
value = StringVar() value .set("texte par défaut") entree = Entry(fenetre , textvariable= value , width=30) entree .pack()	Cree un champs de texte entree dans fenetre de largeur : 30.
entree = Text(fenetre , width=30,height=30) entree .insert(INSERT, "Hello.....") entree .pack()	Cree une zone de texte de 30*30 Insert Hello..... dans la zone de texte
bouton = Checkbutton(fenetre , text="Nouveau?") bouton .pack()	Cree une case a cocher (Check) button avec le texte « Nouveau »
value = StringVar() bouton1 = Radiobutton(fenetre , text="Oui", variable= value , value =1) bouton2 = Radiobutton(fenetre , text="Non", variable= value , value =2) bouton3 = Radiobutton(fenetre , text="Peu être", variable= value , value =3) bouton1 .pack() bouton2 .pack() bouton3 .pack()	Cree variable value Cree button radio du nom buton1 avec texte "oui" donnant à la variable value la valeur "1" Cree button radio du nom buton2 avec texte "non" donnant à la variable value la valeur "2" Cree button radio du nom buton1 avec texte "peut etre" donnant à la variable value la valeur "3"
liste = Listbox(fenetre) liste .insert(0, "Python") liste .insert(1, "PHP") liste .insert(2, "jQuery") liste .insert(3, "CSS") liste .insert(4, "Javascript")	Variable liste entre listbox Insert Python en position 0 Insert PHP en position 1 Insert jQuery en position 2 Insert CSS en position 3 Insert Javascript en position 4
canvas = Canvas(fenetre , width=150, height=120, background='yellow') ligne1 = canvas .create_line(75, 0, 75, 120) ligne2 = canvas .create_line(0, 60, 150, 60) txt = canvas .create_text(75, 60, text="Cible", font="Arial 16 italic", fill="blue") canvas .pack()	Cree une canvas de 150*120px sur un background jaune Cree une ligne dans la canvas sur x0,y0,x1,y1 Cree une 2 ligne dans la canvas Place un texte cible de police arial 16 en italique de couleur bleu

Elements canvas	Elements canvas
<pre>create_arc() create_bitmap() create_image() create_line() create_oval() create_polygon() create_rectangle() create_text() create_window()</pre>	<pre>arc de cercle bitmap image ligne ovale polygone rectangle texte fenetre</pre>
<code>canvas.coords(élément, x0, y0, x1, y1)</code>	change position d'un élément
<code>canvas.delete(élément)</code>	Supprime un élément
http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/canvas-methods.html	Options de canvas
<pre>value = DoubleVar() scale = Scale(fenetre, variable=value) scale.pack()</pre>	<p>Variable value Value prend la valeur de la scrollbar scale</p>
<pre>Frame1 = Frame(fenetre, borderwidth=2, relief=GROOVE) Frame1.pack(side=LEFT, padx=30, pady=30) Frame2 = Frame(fenetre, borderwidth=2, relief=GROOVE) Frame2.pack(side=LEFT, padx=10, pady=10) Frame3 = Frame(Frame2, bg="white", borderwidth=2, relief=GROOVE) Frame3.pack(side=RIGHT, padx=5, pady=5) Label(Frame1, text="Frame 1").pack(padx=10, pady=10) Label(Frame2, text="Frame 2").pack(padx=10, pady=10) Label(Frame3, text="Frame 3",bg="white").pack(padx=10, pady=10)</pre>	<p>Frame1 dans fenetre de bordure 2 en relief GROOVE Positionne Frame1 à gauche a x=30 et y = 30</p> <p>Frame2 dans fenetre de bordure 2 en relief GROOVE Positionne Frame2 à gauche a x=10 et y = 10</p> <p>Frame3 dans Frame2 avec background blanc de bordure 2 en relief GROOVE Positionne Frame3 à droite a x=5 et y = 5</p> <p>Label dans Frame1 de texte Frame 1 a x = 10, y = 10 Label dans Frame2 de texte Frame 2 a x = 10, y = 10 Label dans Frame3 de texte Frame 3 sur fond blanc a x = 10, y = 10</p>
<pre>p = PanedWindow(fenetre, orient=HORIZONTAL) p.pack(side=TOP, expand=Y, fill=BOTH, pady=2, padx=2) p.add(Label(p, text='Volet 1', background='blue', anchor=CENTER)) p.add(Label(p, text='Volet 2', background='white', anchor=CENTER)) p.add(Label(p, text='Volet 3', background='red', anchor=CENTER)) p.pack()</pre>	<p>Cree un PanedWindow p d'orientation horizontal</p> <p>Ajoute un label dans le paned avec texte Volet 1 d'un fond bleu Ajoute un label dans le paned avec texte Volet 2 d'un fond blanc Ajoute un label dans le paned avec texte Volet 3 d'un fond rouge</p>
<pre>s = Spinbox(fenetre, from_=0, to=10) s.pack()</pre>	Choix d'un nombre dans spinbox entre 0 et 10
<pre>l = LabelFrame(fenetre, text="Titre de la frame", padx=20, pady=20) l.pack(fill="both", expand="yes")</pre>	Cree un cadre, avec un label l de texte Titre de la frame en position 20/20

Label(l, text="A l'intérieure de la frame").pack()	Label dans l de texte A l'interieure de la frame
<pre>def callback(): if askyesno('Titre 1', 'Êtes-vous sûr de vouloir faire ça?'): showwarning('Titre 2', 'Tant pis...') else: showinfo('Titre 3', 'Vous avez peur!') showerror("Titre 4", "Aha") Button(text='Action', command=callback).pack()</pre>	<p>Fonction callback() Cree message d'alerte avec "oui" ou "non" en condition choix avec titre titre 1 et texte « Êtes-vous sûr de vouloir faire ça? » Montre fenetre titre 2 avec texte « Tant pis... » Sinon Montre fenetre Titre 3 avec texte « vous avez peur » Montre fenetre Titre 3 avec texte « aha »</p> <p>Cree un bouton avec texte « action » qui appelle la commande callback</p>
<p style="text-align: center;">ALERTES POSSIBLES</p> <pre>showinfo() showwarning() showerror() askquestion() askokcancel() askyesno() askretrycancel()</pre>	<p style="text-align: center;">ALERTES POSSIBLES</p> <pre>Info (ok) Warning (ok) Erreur(ok) Question (oui / non) Cancel (ok / annulé) Oui/non (oui / non) Retry cancel (ok / recommencer)</pre>

<pre> Button(fenetre, text='L1-C1', borderwidth=1).grid(row=1, column=1) Button(fenetre, text='L1-C2', borderwidth=1).grid(row=1, column=2) Button(fenetre, text='L2-C3', borderwidth=1).grid(row=2, column=3) Button(fenetre, text='L2-C4', borderwidth=1).grid(row=2, column=4) Button(fenetre, text='L3-C3', borderwidth=1).grid(row=3, column=3) </pre>	
<pre> photo = PhotoImage(file="ma_photo.png") canvas = Canvas(fenetre,width=350, height=200) canvas.create_image(0, 0, anchor=NW, image=photo) canvas.pack() </pre>	
<pre> filename = askopenfilename(title="Ouvrir votre document",filetypes=[('txt files','*.txt'),('all files','*.*)']) fichier = open(filename, "r") content = fichier.read() fichier.close() Label(fenetre, text=content).pack(padx=10, pady=10) </pre>	<p>Demande a l'utilisateur d'ouvrir un fichier</p> <p>Ouvre le fichier demander en mode lecture Content prend la valeur du fichier lu Ferme le fichier</p> <p>Affiche le texte content dans un label</p>
<pre> def clavier(event): touche = event.keysym print(touche) canvas = Canvas(fenetre, width=500, height=500) canvas.focus_set() canvas.bind("<Key>", clavier) canvas.pack() </pre>	<p>Fonction event touche clavier Touche prend la valeur de la touche appuyé Affiche la touche</p> <p>Cree une canvas de 500/500</p>

SQL pour PYTHON

<pre> import mysql.connector conn = mysql.connector.connect(host="localhost",user="root",password="XXX", database="test1") cursor = conn.cursor() conn.close() </pre>	<p>Connection à la base</p>
<pre> cursor.execute(""" CODE SQL """) </pre>	<p>Pour cree /modifier/inséré</p>
<pre> cursor.execute("""SELECT id, name, age FROM users WHERE id = %s""", ("5",)) rows = cursor.fetchall() for row in rows: print('{0} : {1} - {2}'.format(row[0], row[1], row[2])) </pre>	<p>Recupere des données de la table</p>

SOURIS	<u>-1 pour le MOUSEMOTION</u>
1	Bouton gauche
2	Bouton molette ou gauche/droite en même temps
3	Bouton droit
4	Molette vers le haut
5	Molette vers le bas
Lettres: K_a ... K_z	Abcdefghijklmnopqrstuvwxyz
Nombres: K_0 ... K_9	0123456789
Controles: K_TAB	Tab
K_RETURN	Entrée
K_ESCAPE	Echap
K_SCROLLOCK	Arret defil
K_SYSREQ	Imp ecran
K_BREAK	pause
K_DELETE	Suppr
K_BACKSPACE	Arriere
K_CAPSLOCK	Verouillage majuscule
K_CLEAR	////////////////////
K_NUMLOCK	Verrouillage pave numerique
Ponctuation: K_SPACE	Espace
K_PERIOD	.
K_COMMA	,
K_QUESTION	?
K_AMPERSAND	&
K_ASTERISK	*
K_AT	@
K_CARET	^
K_BACKQUOTE	`
K_DOLLAR	\$
K_EQUALS	=
K_EURO	€
K_EXCLAIM	!
K_SLASH, K_BACKSLASH	/, \
K_COLON, K_SEMICOLON	,, ;
K_QUOTE, K_QUOTEDBL	‘, ”
K_MINUS, K_PLUS	- , +
K_GREATER, K_LESS	> , <
Parenthèses: K_RIGHTBRACKET, K_LEFTBRACKET	[,]
K_RIGHTPAREN, K_LEFTPAREN	(,)
Touches F: K_F1 ... K_F15	F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14 F15
Touches d'édition: K_HELP	? haut
K_HOME	////////////////////
K_END	Fin
K_INSERT	Inser
K_PRINT	

K_PAGEUP, K_PAGEDOWN	Print ecr
K_FIRST, K_LAST	Page prec , pag suiv
Clavier numérique:	////////////////////
K_KP0 ... K_KP9	
K_KP_DIVIDE	0123456789
K_KP_ENTER	/ numerique
K_KP_EQUALS	Entrer numerique
K_KP_MINUS	=
K_KP_MULTIPLY	-
K_KP_PERIOD	*
K_KP_PLUS	. numerique
SHF,CTL,ALT etc:	+
K_LALT, K_RALT	
K_LCTRL, K_RCTRL	Alt gauche , alt droite
K_LSUPER, K_RSUPER	Ctrl gauche , ctrl droite
K_LSHIFT, K_RSHIFT	////////////////////
K_RMETA, K_LMETA	Shift gauche , shift droite
Flèches:	////////////////////
K_LEFT	
K_UP	Fleche gauche
K_RIGHT	Fleche heure
K_DOWN	Fleche droite
Autres:	Fleche bas
K_MENU	
K_MODE	Touche de menu (à côté de ctrl)
K_PAUSE	////////////////////
K_POWER	Touche pause
K_UNDERSCORE	Touche power (à ne pas utiliser)
K_HASH	_
	r"