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

|  |  |
| --- | --- |
| def NOM(reponse):  return reponse  NOM(reponse) | Cree une fonction du nom NOM en prenant la valeur de reponse  Retourne la valeur de reponse  Appel la fonction NOM en lui envoyant la valeur reponse |
| def le\_plus\_grand(\*args):  le\_plus\_grand(-10, -5, 5, 10) | \*args = TOUT les arguments reçut de le\_plus\_grand |
|  |  |
| try:  entier = int(input("entier :"))  print ("vous avez entré un entier")  except:  print ("vous n'avez pas entre un entier") | Lance le code TRY, si il y a une erreur, lance le code EXCEPT a la place |
|  |  |
| Print (« message »)  Print (« message » , « message », variable) | Ecrit message dans la console  Permet de metre des variable et messages a la suite |
| print ma\_variable | Ecrit la valeur de ma\_variable |
| Print (len(string)) | Marque le nombre de lettre/taille de la variable string |
| print (" ".join(animaux))  ' ; '.join(['a', 'b', 'c']) | 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 |
| print (0b1)  print (0b10) | 0b1 = binaire = 1  0b10 = binaire = 2 |
| .lower() | Met les lettres en minuscule |
| .upper() | Met les lettres en majuscule |
| ma\_variable = 10 | Cree une variable ma\_variable de valeur 10 |
| nombres = [5, 6, 7, 8]  nombre = [[0]\*15 for i in range(15)] | Cree un tableau nombre de 4 valeurs  Cree un tableau nombre a 2dimensions de 15\*15 rempli de 0 |
| liste = entete.split(";") | Les valeurs avec ; de la liste sont séparé et mis dans une liste |
| 'hello crazy world!'.replace("o", "u" ) | Remplace les “o” par des “u” |
| Nombre = range(6)  Nombre = range(1,6) | Equivalent a nombres [0,1,2,3,4,5]  Equivalent a nombres [1,2,3,4,5] |
| nombre.append(20) | Ajoute une case avec la valeur 20 dans nombre |
| slice = nombre[1:3] | Slice prend les valeurs 1 et 2 du tableau nombre (donc 6 et 7) |
| animaux.index("ratel") | Renvoie la position(index) de ratel du tableau animaux |
| animaux.remove("ratel") | Supprime ratel du tableau animaux |
| animaux.pop(1) | Supprime le 2eme élément du tableau animaux et renvoie cette valeur avant de supprimer |
| del(animaux[1]) | Supprime le 2eme élément du tableau animaux |
| animaux.insert(1, "chien") | Met à la position 1 le mot chien dans le tableau animaux |
| animaux.sort() | Met le tableau animaux dans l’ordre alphabétique |
| liste = sorted(liste) | Trie dans un ordre croissant |
| str(ma\_variable) | Change ma\_variable en string |
| int(ma\_variable) | Transforme ma\_variable en entier |
| int("10",2) | Prévient que l’entrée est du binaire (base 2) et le converti en base 10 (entier) |
| mon\_bool = True | Cree un booleen du nom de mon\_bool |
| type(4.02) | Renvoi le type de 4.02 (ici float) |
|  |  |
| nom = input("Quel est votre nom ?")  chiffre = int(input("Quel est votre nom ?")) | Affiche « quel est votre nom ? » et demande à l’utilisateur de saisir la valeur de nom  Pour un chiffre |
| Maintenant = Datetime.now()  Annee\_actuelle = Maintenant.year  mois\_actuelle = Maintenant.month  jour\_actuelle = Maintenant.day  heure\_actuelle = Maintenant.hour  minutes\_actuelle = Maintenant.minute  secondes\_actuelle = Maintenant.second | Maintenant prend la Date et heure actuelle  Année  Mois  Jour  Heure  Minute  secondes |
|  |  |
| if reponse == "Left":  elif reponse < 5:  else: | Si reponse est Left  Sinon si reponse < 5  sinon |
| for nombre in ma\_liste:  for a, b in zip(liste\_a, liste\_b):  for nombre in range(len(ma\_liste)) | Pour nombreDans ma\_liste  Pour a et b dans liste\_a et liste\_b (s’arrête à la fin du plus cour)  Nombre prend la valeur du premier nombre, puis du deuxième… jusqu’à que ma\_liste ne contienne plus de nombre |
| for index, element in enumerate(choix): | Index prend l’emplacement et element la valeur dans choix |
| While nimporte == 9:  While nimporte: | Tant que nimporte inférieur à 9  Tant que nimporte est True |
| break | Termine la boucle sans la continuer |
|  |  |
| and  or  not | ET  OU  NON |
| Variable.isalpha() | Renvoie vrai s’il n’y a que des lettre  Faux si il y a au moins un chiffre |
|  |  |
| sqrt(25) | Racine carré de 25 |
| dir(math) | Montre toutes les fonctions de la librairie math |
| max(args)  min(args)  abs(arg) | Maximum de…  Minimum de…  Absolue de… |
| residents = {'Puffin' : 104, 'Paresseux' : 105, 'Python' : 106} | Affectez un dictionnaire avec trois paires de valeur-clé à residents |
| 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  print 5 << 1 # Décalage à gauche  print 8 & 5 # ET binaire  print 9 | 4 # OU binaire  print 12 ^ 42 # OU EXCLUSIF binaire  print ~88 # NON binaire | Décale tous les 1 binaire de 5 de 4 emplacement vers la droite  Décale tous les 1 binaire de 5 de 1 emplacement vers la gauche  Ne laisse a 1 seulement si la position du 1 du 8 et du 5 sont les même  Laisse la position des 1 de 9 et de 4  Laisse la position des 1 de 9 et de 4 sauf si les 2 sont de même position  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 defaut 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):  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 | Classe Triangle  Constructeur  Variable  Variable  Variable  Nombre\_de\_cotes = 3  Methode verifier\_angle  **Condition somme des 3 angles = 180**  Returne True  Sinon  Retourne False  Classe Equilateral avec les parametres de Triangles  Angle = 60  Angle1 de Triangle = angle  Angle2 de Triangle = angle  Angle3 de Triangle = angle |
|  |  |
| mon\_fichier = open("output.txt","r+") | Ouvre le fichier output.txt  W = lecture  R = écriture  R+ = lecture/écriture  A = écriture en fin de fichier |
| 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" | Verifie sur il y a guitare OU piano |
| **r"Quoi \?**" | **Metrre "\" pour tous les metacaractère :**  **r" ! ^ $ ( ) [ ] { } ? + \* . \ |** |
| r"^Bonjour" | Bonjour devra obligatoirement être le premier mot de la chaine |
| r"zéro$" | zéro devra obligatoirement être le dernier mot de la chaine |
| r"gr[ioa]s" | Recherche gris OU gros OU gras |
| r"[a-z] "  r"[0-9] "  r"[a-z0-9] " | Une lettre (de A a Z)  Un chiffre (de 0 a 9)  Une lettre OU un chiffre |
| r"[^0-9] " | Recherche tout sauf un chiffre |
| r"a? "  r"a+"  r"a\*" | Reconnais 0 ou 1 "a" **PAS PLUS de 1**  Au moins 1 "a" **A PARTIR DE 1**  0 ou plusieurs a |
| r"Ay(ay|oy)\* " | VRAI pour AY et autant de ay ou de oy que l'on veux |
| r"a{3}"  r"a{3,5}"  r"a{3,} " | Fonctione que pour aaa  Fonction pour aaa, aaaa, aaaaa seulement  Fonctionne pour autant de a que l'on veut |
| \d  \D  \w  \W  \t  \n  \r  \s  \S  . | = [0-9]  = [^0-9]  = [a-zA-Z0-9\_]  = [^a-zA-Z0-9\_]  Indique une tabulation  Indique une nouvelle ligne  Indique un retour chariot  Indique un espace blanc  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 ou l'on fais la recherche (chaine de caractère) |
| (?:nsti) | La parenthèse ne comptera pas pour une variable et sera sauté |

MODE FENETRE

|  |  |
| --- | --- |
| import pygame  from pygame.locals import \* | Importe tout pygame |
| import pygame.mixer | Importe seulement pour le son |
| pygame.init() | OBLIGATOIRE après l’import |
|  |  |
| fenetre = pygame.display.set\_mode((640, 480),OPTION) | Cree une fenetre de 640px de large et 480px de haut |
| **Listes OPTIONS**  RESIZABLE  FULLSCREEN | Permet de changer la taille fenêtre en temps réel  Pleine écran |
| fond = pygame.image.load("background.jpg").convert() | Charge l’image background.jpg dans fond  Pour les images avec de la transparence faire convert.alpha() |
| position\_image = image.get\_rect() | Cree variable avec les positions de l’image |
| fenetre.blit(fond, (0,0)) | Place l’image fond en position 0,0 |
| pygame.display.flip() | Initialise l’ecran (affiche les images) |
| while continuer:  for event in pygame.event.get():  if event.type == QUIT:  continuer = 0  elif event.type == VIDEORESIZE:  if event.w > 500 or event.h > 500:  continuer = 0  elif event.type == MOUSEBUTTONUP:  if event.button == 1 and event.pos[1]<100:  print ("ok")  elif event.type == MOUSEMOTION:  if event.buttons[0] == 1:  print(“mouvement”)  elif event.type == KEYDOWN:  if event.key == K\_SPACE:  print("Espace")  if event.key == K\_RETURN:  print("Entrée") | Penser à mettre une variable continuer  Attend un event  Si cette event est le clic sur la croix  Continuer = 0  Si la fenetre est redimensionné  Si fenetre depasse 500px\*500px  Continuer = 0  Si c’est un clic de souris (relachement)  Boutton gauche et position y souris < 100  Affiche « ok »  S’il y a mouvement souris  Si bouton gauche est pressé  Affichicher « mouvement »  Si c’est une touche du clavier  Touche espace  Affiche « espace » dans la console  Touche entrée  Affiche « entrée » dans la console |
| if event.gain == 1 and event.state == 1: | 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 |
| pygame.key.set\_repeat(400, 30) | Répétition quand on reste appuyé |
| event.pos[0]<100  event.pos[1]<100 | Position x souris < 100  Position y souris < 100 |
| pygame.time.Clock().tick(30) | Pause de 30ms |
|  |  |
|  |  |
| son = pygame.mixer.Sound("son.wav") | Cree un son |
| son.play() | Lance le son |
| son.stop() | Arrête le son |
| son.fadeout(300) | Fondu à la fin du son |
| pygame.mixer.pause() | Met sur pause TOUS les sons |
| pygame.mixer.unpause() | Enlève la pause de TOUS les sons |
| pygame.mixer.stop() | Arrête TOUS les sons |
| volume = pygame.mixer.music.get\_volume() | Retourne la valeur du volume, entre 0 et 1 |
| pygame.mixer.music.set\_volume(0.5) | 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 fenêtre |
| 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 |
| boutton = 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  create\_arc()  create\_bitmap()  create\_image()  create\_line()  create\_oval()  create\_polygon()  create\_rectangle()  create\_text()  create\_window() | Elements canvas  arc de cercle  bitmap  image  ligne  ovale  polygone  rectangle  texte  fenetre |
| canvas.coords(élément, x0, y0, x1, y1) | change position d’un élément |
|  |  |
| canvas.delete(élément) | Supprime un élément |
| http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/canvas-methods.html | Options de canvas |
| value = DoubleVar()  scale = Scale(fenetre, variable=value)  scale.pack() | Variable value  Value prend la valeur de la scrollbar scale |
| 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) | Frame1 dans fenetre de bordure 2 en relief GROOVE  Positionne Frame1 à gauche a x=30 et y = 30  Frame2 dans fenetre de bordure 2 en relief GROOVE  Positionne Frame2 à gauche a x=10 et y = 10  Frame3 dans Frame2 avec background blanc de bordure 2 en relief GROOVE  Positionne Frame3 à droite a x=5 et y = 5  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 = 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() | Cree un PanedWindow p d’orientation horizontal  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 |
| s = Spinbox(fenetre, from\_=0, to=10)  s.pack() | Choix d’un nombre dans spinbox entre 0 et 10 |
| l = LabelFrame(fenetre, text="Titre de la frame", padx=20, pady=20)  l.pack(fill="both", expand="yes")    Label(l, text="A l'intérieure de la frame").pack() | Cree un cadre, avec un label l de texte Titre de la frame en position 20/20  Label dans l de texte A l’interieure de la frame |
| 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() | 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 fenertre Titre 3 avec texte « vous avez peur »  Montre fenetre Titre 3 avec texte « aha »  Cree un bouton avec texte « action » qui appelle la commande callback |
| ALERTES POSSIBLES  showinfo()  showwarning()  showerror()  askquestion()  askokcancel()  askyesno()  askretrycancel() | ALERTES POSSIBLES  Info (ok)  Warning (ok)  Erreur(ok)  Question (oui / non)  Cancel (ok / annulé)  Oui/non (oui / non)  Retry cancel (ok / recommencer) |
| CREATION D'UNE BARRE DE MENU  def alert():  showinfo("alerte", "Bravo!")  menubar = Menu(fenetre)  menu1 = Menu(menubar, tearoff=0)  menubar.add\_cascade(label="Fichier", menu=menu1)  menu1.add\_command(label="Créer", command=alert)  menu1.add\_command(label="Editer", command=alert)  menu1.add\_separator()  menu1.add\_command(label="Quitter", command=fenetre.quit)  menu2 = Menu(menubar, tearoff=0)  menubar.add\_cascade(label="Editer", menu=menu2)  menu2.add\_command(label="Couper", command=alert)  menu2.add\_command(label="Copier", command=alert)  menu2.add\_command(label="Coller", command=alert)  menu3 = Menu(menubar, tearoff=0)  menubar.add\_cascade(label="Aide", menu=menu3)  menu3.add\_command(label="A propos", command=alert)  fenetre.config(menu=menubar) | CREATION D'UNE BARRE DE MENU  Fonction alert  Envoie une alerte avec marquer "bravo"  Cree un menu bar  Cree le premier menu  Cree le menu "fichier" dans menu1  Ajouter "créer" dans le menu1 et appelle la fonction alert()  Ajouter "editer" dans le menu1 et appelle la fonction alert()  Ajoute un separateur dans menu1  Ajouter "Quitter" dans le menu1 et appelle la fonction fenetre.quit  Cree le deuxieme menu  Cree le menu "Editer" dans menu2  Ajouter "Couper" dans le menu2et appelle la fonction alert()  Ajouter "Copier" dans le menu2et appelle la fonction alert()  Ajouter "Coller" dans le menu2 et appelle la fonction alert()  Cree le troisieme menu  Cree le menu "Aide" dans menu2  Ajouter "A propo" dans le menu2 et appelle la fonction alert()  Configure la barre de menu |
| PARAMETRES DU .pack  side=TOP : haut  side=LEFT : gauche  side=BOTTOM : bas  side=RIGHT : droite  padx = 5  pady = 60 | PARAMETRES DU .pack  Place en haut  Place a gauche place en bas  Place a droite  Deplace de 5pixel a droite  Deplace de 60 pixel en bas |
| b1 = Button(fenetre, text ="FLAT", relief=FLAT).pack()  b2 = Button(fenetre, text ="RAISED", relief=RAISED).pack()  b3 = Button(fenetre, text ="SUNKEN", relief=SUNKEN).pack()  b4 = Button(fenetre, text ="GROOVE", relief=GROOVE).pack()  b5 = Button(fenetre, text ="RIDGE", relief=RIDGE).pack() | http://apprendre-python.com/images/tkinter-relief.png |
| 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) | http://apprendre-python.com/images/tkinter-grid-2.png |
| photo = PhotoImage(file="ma\_photo.png")  canvas = Canvas(fenetre,width=350, height=200)  canvas.create\_image(0, 0, anchor=NW, image=photo)  canvas.pack() | http://apprendre-python.com/images/tkinter-image.png |
|  |  |
| 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) | Demande a l'utilisateur d'ouvrir un fichier  Ouvre le fichier demander en mode lecture  Content prend la valeur du fichier lu  Ferme le fichier  Affiche le texte content dans un label |
| def clavier(event):  touche = event.keysym  print(touche)  canvas = Canvas(fenetre, width=500, height=500)  canvas.focus\_set()  canvas.bind("<Key>", clavier)  canvas.pack() | Fonction event touche clavier  Touche prend la valeur de la touche appuyé  Affiche la touche  Cree une canvas de 500/500 |

SQL pour PYTHON

|  |  |
| --- | --- |
| import mysql.connector  conn = mysql.connector.connect(host="localhost",user="root",password="XXX", database="test1")  cursor = conn.cursor()  conn.close() | Connection à la base |
|  |  |
| cursor.execute("""  CODE SQL  """) | Pour cree /modifier/inséré |
| 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])) | Recupere des données de la table |

|  |  |
| --- | --- |
| SOURIS  1  2  3  4  5 | **-1 pour le MOUSEMOTION**  Bouton gauche  Bouton molette ou gauche/droite en même temps  Bouton droit  Molette vers le haut  Molette vers le bas |
| Lettres: K\_a ... K\_z  Nombres: K\_0 ... K\_9  Controles: K\_TAB K\_RETURN K\_ESCAPE K\_SCROLLOCK K\_SYSREQ K\_BREAK K\_DELETE K\_BACKSPACE K\_CAPSLOCK K\_CLEAR K\_NUMLOCK  Ponctuation: K\_SPACE 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  Touches d'édition: K\_HELP K\_HOME K\_END K\_INSERT K\_PRINT K\_PAGEUP, K\_PAGEDOWN K\_FIRST, K\_LAST  Clavier numérique: K\_KP0 ... K\_KP9 K\_KP\_DIVIDE K\_KP\_ENTER K\_KP\_EQUALS K\_KP\_MINUS K\_KP\_MULTIPLY K\_KP\_PERIOD K\_KP\_PLUS  SHF,CTL,ALT etc: K\_LALT, K\_RALT K\_LCTRL, K\_RCTRL K\_LSUPER, K\_RSUPER K\_LSHIFT, K\_RSHIFT K\_RMETA, K\_LMETA  Flèches: K\_LEFT K\_UP K\_RIGHT K\_DOWN  Autres: K\_MENU K\_MODE K\_PAUSE K\_POWER K\_UNDERSCORE K\_HASH | Abcdefghijklmnopqrstuvwxyz  0123456789  Tab  Entrée  Echap  Arret defil  Imp ecran  pause  Suppr  Arriere  Verouillage majuscule  /////////////////////////  Verrouillage pave numerique  Espace  .  ,  ?  &  \*  @  ^  `  $  =  €  !  / , \  , , ;  ‘ , ‘’  - , +  > , <  [ , ]  ( , )  F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14 F15  ? haut  ///////////////////////  Fin  Inser  Print ecr  Page prec , pag suiv  ////////////////////////  0123456789  / numerique  Entrer numerique  =  -  \*  . numerique  +  Alt gauche , alt droite  Ctrl gauche , ctrl droite  ////////////////////////  Shift gauche , shift droite  ////////////////////////  Fleche gauche  Fleche heure  Fleche droite  Fleche bas  Touche de menu (à côté de ctrl)  /////////////////////////  Touche pause  Touche power (**à ne pas utiliser**)  \_  r" |