Programmes Python3 de tests

Programme python proposé par Xavier

menu_prenom_option_devinette_calculette_0000.py.zip

Modification du programme python proposé par Xavier modifié par GL

menu_prenom_option_devinette_calculette_0003.py.zip

python_prog-test_mouv_lum_son.zip

rectangleentourcercle.py.zip

python:neopixel

bonjour en python dans rectangle

Balle rebondit avec pygame

Cercle en mode blocs en python

Programme Bonjour

bonjour001.py

```
import time

print('Bonjour à tous !')
time.sleep(2)

message = "Bonjour à tous!"
print(message)
time.sleep(2)


salutation = "Bonjour"
nom = "à tous  "
print(salutation + ", " + nom + "!")
time.sleep(2)

nom = "à tous "
print(f"Bonjour, {nom}!")
time.sleep(2)

print("Bonjour,")
print("à tous")
```

```
time.sleep(2)

print("Bonjour,\nà tous!")
time.sleep(2)
```

Menu 1 en mode texte tres simple Python3

py menu002.py

```
menu_options = {
    1: 'Option 1',
    2: 'Option 2',
    3: 'Option 3',
    4: 'Exit',
}

def print_menu():
    for key in menu_options.keys():
        print (key, '--', menu_options[key] )

def option1():
     print('Handle option \'Option 1\'')

def option2():
     print('Handle option \'Option 2\'')

def option3():
     print('Handle option \'Option 3\'')

if __name__=='__main__':
    while(True):
        print_menu()
        option = ''
        try:
            option = int(input('Enter your choice: '))
        except:
            print('Wrong input. Please enter a number ...')
        #Check what choice was entered and act accordingly
        if option == 1:
           option1()
        elif option == 2:
            option2()
        elif option == 3:
            option3()
        elif option == 4:
            print('Thanks message before exiting')
```

```
            exit()
        else:
            print('Invalid option. Please enter a number between 1 and
4.')
```

Menu en mode texte python3

doc console-menu

py menu001.py

```
# Import the necessary packages
# pip install console-menu
# Modifier par GL le 15/01/2022
###############################

from consolemenu import *
from consolemenu.items import *

# Create the menu
menu = ConsoleMenu("Essais de menu", "sous menu")

# Create some items

# MenuItem is the base class for all items, it doesn't do anything when
selected
menu_item = MenuItem("Menu 1")

# A FunctionItem runs a Python function when selected
function_item = FunctionItem("Menu 2", input, ["Entrer une donnee "])

# A CommandItem runs a console command
command_item = CommandItem("Demarrer une console de commande",  "touch
hello.txt")

# A SelectionMenu constructs a menu from a list of strings
selection_menu = SelectionMenu(["item1", "item2", "item3"])
```

```
# A SubmenuItem lets you add a menu (the selection_menu above, for
example)
# as a submenu of another menu
submenu_item = SubmenuItem("Sous Menu item", selection_menu, menu)

# Once we're done creating them, we just add the items to the menu
menu.append_item(menu_item)
menu.append_item(function_item)
menu.append_item(command_item)
menu.append_item(submenu_item)

# Finally, we call show to show the menu and allow the user to interact
menu.show()
```

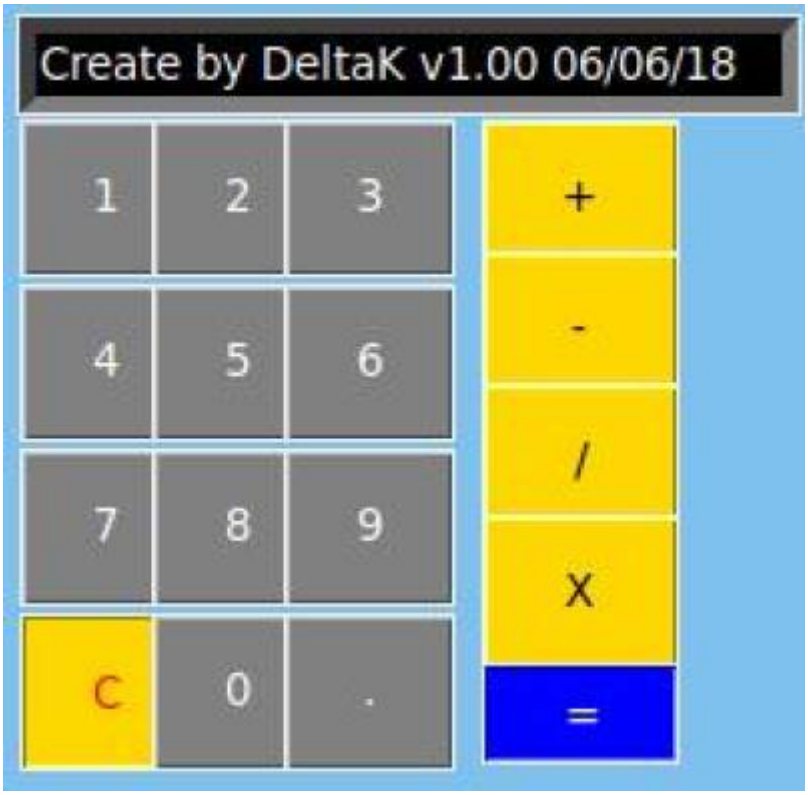
calculatrice python3 menu texte

py calc3.py

```
## Programme de calculatrice tres simple pour demarrer en python
## Modifier par GL 01/2022
#############

def calculate():
   operation = input('''
SVP , Entrez le type d operation desire:
+ pour l addition
- pour la soustraction
* pour la  multiplication
/ pour la  division
''')

   number_1 = int(input('SVP entrer le premier nombre: '))
   number_2 = int(input('SVP entrer le deuxieme nombre: '))

   if operation == '+':
       print('{} + {} = '.format(number_1, number_2))
       print(number_1 + number_2)

   elif operation == '-':
       print('{} - {} = '.format(number_1, number_2))
       print(number_1 - number_2)

   elif operation == '*':
       print('{} * {} = '.format(number_1, number_2))
       print(number_1 * number_2)
```

```
    elif operation == '/':
        print('{} / {} = '.format(number_1, number_2))
        print(number_1 / number_2)

    else:
        print('Vous n avez pas chois une operation valide , veuillez
recommencer .')

    # Add again() function to calculate() function


def again():
    calc_again = input('''
Voulez vous calculer a nouveau?
SVP taper O pour Oui , et N pour Non.
''')

    if calc_again.upper() == 'O':
        calculate()
    elif calc_again.upper() == 'N':
        print('Merci, au revoir...')
    else:
        again()

calculate()
```

Une calculatrice en python3 mode graphique



py calc10.py

```
#////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
//////////////
#
CALCULATRICR (GRAPHIQUE)
#\\\\\\\\\\\\\\\\\\\

#--------------
#IMMPORTATION :
#--------------
from tkinter import * # Tkinter

#---------
# CLASSE :
#---------
class Calculator():
    def __init__(self): # Construction
        self.phase1 = 0 # Premier nombre
        self.phase2 = 0 # Deuxième nombre
        self.final = 0 # Valeur finale
        self.entry = StringVar() # Capte les valeurs écrit
        self.text = "" # Nombre écrir par l'utilisateur
        self.signe = "" # Type d'opération
        self.entry.set("Create by DeltaK v1.00 06/06/18")

    def init(self): # Initialisation
```

```
        self.phase1 = 0 # Premier nombre
        self.phase2 = 0 # Deuxième nombre
        self.final = 0 # Valeur finale
        self.text = "" # Nombre écrir par l'utilisateur
        self.signe = "" # Type d'opération

    def afficher_Nb(self): # Afficher les nombre sur écran
        self.entry.set(self.text)

    def operation(self): # Vérification du type d'opération
        try :
            if "+" in self.text:
                self.Plus()
            elif "-" in self.text:
                self.Sous()
            elif "/" in self.text:
                self.Div()
            elif "X" in self.text:
                self.Mult()
        except:
            self.entry.set("ERROR")
            self.init()

    def Plus(self): # Addition
        nb = self.text.split("+")
        self.phase1 = float(nb[0])
        self.phase2 = float(nb[1])
        self.final = self.phase1 + self.phase2
        self.entry.set(str(self.final))
        self.init()

    def Sous(self): # Soustraction
        nb = self.text.split("-")
        self.phase1 = float(nb[0])
        self.phase2 = float(nb[1])
        self.final = self.phase1 - self.phase2
        self.entry.set(str(self.final))
        self.init()

    def Div(self): # Division
        nb = self.text.split("/")
        self.phase1 = float(nb[0])
        self.phase2 = float(nb[1])
        self.final = self.phase1 / self.phase2
        self.entry.set(str(self.final))
        self.init()

    def Mult(self): # Multiplication
        nb = self.text.split("X")
        self.phase1 = float(nb[0])
        self.phase2 = float(nb[1])
```

```
        self.final = self.phase1 * self.phase2
        self.entry.set(str(self.final))
        self.init()

#------------
# FONCTIONS :
#------------
def Button1 (): # Actionnerle bouton 1
    calculatrice.text += "1"
    calculatrice.entry.set(calculatrice.text)

def Button2 (): # Actionnerle bouton 2
    calculatrice.text += "2"
    calculatrice.entry.set(calculatrice.text)

def Button3 (): # Actionnerle bouton 3
    calculatrice.text += "3"
    calculatrice.entry.set(calculatrice.text)

def Button4 (): # Actionnerle bouton 4
    calculatrice.text += "4"
    calculatrice.entry.set(calculatrice.text)

def Button5 (): # Actionnerle bouton 5
    calculatrice.text += "5"
    calculatrice.entry.set(calculatrice.text)

def Button6 (): # Actionnerle bouton 6
    calculatrice.text += "6"
    calculatrice.entry.set(calculatrice.text)

def Button7 (): # Actionnerle bouton 7
    calculatrice.text += "7"
    calculatrice.entry.set(calculatrice.text)

def Button8 (): # Actionnerle bouton 8
    calculatrice.text += "8"
    calculatrice.entry.set(calculatrice.text)

def Button9 (): # Actionnerle bouton 9
    calculatrice.text += "9"
    calculatrice.entry.set(calculatrice.text)

def Button0 (): # Actionnerle bouton 0
    calculatrice.text += "0"
    calculatrice.entry.set(calculatrice.text)

def ButtonF(): # Actionnerle bouton F
    calculatrice.text += "."
```

```
    calculatrice.entry.set(calculatrice.text)

def ButtonP (): # Actionnerle bouton P
    calculatrice.text += "+"
    calculatrice.entry.set(calculatrice.text)

def ButtonS (): # Actionnerle bouton S
    calculatrice.text += "-"
    calculatrice.entry.set(calculatrice.text)

def ButtonD (): # Actionnerle bouton D
    calculatrice.text += "/"
    calculatrice.entry.set(calculatrice.text)

def ButtonM (): # Actionnerle bouton M
    calculatrice.text += "X"
    calculatrice.entry.set(calculatrice.text)

def ButtonE (): # Actionnerle bouton E
    calculatrice.operation()

def ButtonC (): # Actionnerle bouton c
    calculatrice.entry.set("")
    calculatrice.init()

#----------
# FENETRE :
#----------
fen = Tk() # Création de a fenêtr 1e
fen.geometry("200x240") # Définition de la fenêtre
fen.title("Calculatrice v1.0") # Titre de la calculatrice
fen["bg"]= "SkyBlue2" # Coleur de la fenêtre
fen["relief"] = "raised" # Profondeur de la fenêtre
#------------
# PROGRAMME :
#------------
# Création instance
calculatrice = Calculator()

# ATTRIBUTS DE LA FENETRE
#########################
# // Ecran calculatrice //
ECRAN = Entry(fen, width=28, textvariable=calculatrice.entry, bg
="black", fg="white", relief=SUNKEN, bd=5).place(x=9, y=8)

# // Bouttons //
B1 = Button(fen, text="1", command=Button1, width=3, height=2,
bg="grey", fg="white").place(x=10, y=40) # Boutton 1
B2 = Button(fen, text="2", command=Button2, width=3, height=2,
bg="grey", fg="white").place(x=50, y=40) # Boutton 2
B3 = Button(fen, text="3", command=Button3, width=3, height=2,
```

```
bg="grey", fg="white").place(x=90, y=40) # Boutton 3
B4 = Button(fen, text="4", command=Button4, width=3, height=2,
bg="grey", fg="white").place(x=10, y=90) # Boutton 4
B5 = Button(fen, text="5", command=Button5, width=3, height=2,
bg="grey", fg="white").place(x=50, y=90) # Boutton 5
B6 = Button(fen, text="6", command=Button6, width=3, height=2,
bg="grey", fg="white").place(x=90, y=90) # Boutton 6
B7 = Button(fen, text="7", command=Button7, width=3, height=2,
bg="grey", fg="white").place(x=10, y=140) # Boutton 7
B8 = Button(fen, text="8", command=Button8, width=3, height=2,
bg="grey", fg="white").place(x=50, y=140) # Boutton 8
B9 = Button(fen, text="9", command=Button9, width=3, height=2,
bg="grey", fg="white").place(x=90, y=140) # Boutton 9
BC = Button(fen, text="C", command=ButtonC, width=3, height=2,
bg="gold", fg="red", relief=RIDGE).place(x=10, y=190) # Boutton C
(Clear)
B0 = Button(fen, text="0", command=Button0, width=3, height=2,
bg="grey", fg="white").place(x=50, y=190) # Boutton 0
BF = Button(fen, text=".", command=ButtonF, width=3, height=2,
bg="grey", fg="white").place(x=90, y=190) # Boutton = (égale)

BP = Button(fen, text="+", command=ButtonP, width=4, height=2,
bg="gold", fg="black", relief=GROOVE).place(x=150, y=40) # Boutton +
(addition)
BS = Button(fen, text="-", command=ButtonS, width=4, height=2,
bg="gold", fg="black", relief=GROOVE).place(x=150, y=80) # Boutton -
(soustacrtion)
BD = Button(fen, text="/", command=ButtonD, width=4, height=2,
bg="gold", fg="black", relief=GROOVE).place(x=150, y=120) # Boutton /
(division)
BM = Button(fen, text="X", command=ButtonM, width=4, height=2,
bg="gold", fg="black", relief=GROOVE).place(x=150, y=160) # Boutton X
(multiplication)
BE = Button(fen, text="=", command=ButtonE, width=4, height=1,
bg="blue", fg="white", relief=RIDGE).place(x=150, y=205) # Button =
(égale)

fen.mainloop() # Gestion de la fenêtre
```

lire un fichier texte

py testfich001.py

```
#! /usr/bin/env python
# coding: utf-8
```

```
menu_options = {
    1: 'Option 1',
    2: 'Option 2',
    3: 'Aide -- 3',
    4: 'Exit',
}

def print_menu():
    for key in menu_options.keys():
        print (key, '--', menu_options[key] )

def option1():
     print('Handle option \'Option 1\'')

def option2():
     print('Handle option \'Option 2\'')

def option3():
    f = open('aide001.txt', 'r')
    data = f.read()
    f.close
    print(data)

if __name__=='__main__':
    while(True):
        print_menu()
        option = ''
        try:
            option = int(input('Enter your choice: '))
        except:
            print('Wrong input. Please enter a number ...')
        #Check what choice was entered and act accordingly
        if option == 1:
           option1()
        elif option == 2:
            option2()
        elif option == 3:
            option3()
        elif option == 4:
            print('Thanks message before exiting')
            exit()
        else:
            print('Invalid option. Please enter a number between 1 and
4.')
```

Inserer le fichier texte “aide001.txt” dans le meme repertoire que le programme ci-dessus

aide001.txt

```
Ceci est un fichier d aide à completer suivant les besoins :
```

```
une introduction  :

Python (prononcé /pi.tɔ̃/) est un langage de programmation interprété,
multi-paradigme et multiplateformes.

Il favorise la programmation impérative structurée, fonctionnelle et
orientée objet.

Il est doté d'un typage dynamique fort, d'une gestion automatique de la
mémoire par ramasse-miettes et d'un système de gestion d'exceptions ;

il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl.

Le langage Python est placé sous une licence libre proche de la licence
BSD3 et fonctionne sur la plupart des plates-formes informatiques, des
smartphones aux ordinateurs centraux4,
de Windows à Unix avec notamment GNU/Linux en passant par macOS,  ou
encore Android, iOS, et peut aussi être traduit en Java ou .NET. Il est
conçu pour optimiser la productivité
des programmeurs en offrant des outils de haut niveau et une syntaxe
simple à utiliser.

Il est également apprécié par certains pédagogues qui y trouvent un
langage où la syntaxe, clairement séparée des mécanismes de bas niveau,
permet une initiation aisée aux concepts de base de la programmation5.
```

Balle qui rebondit avec pygame

ballerebondit001.mp4

exemple001.py

```python
import sys, pygame
pygame.init()
import time

size = width, height = 1280, 960
speed = [5, 5]
black = 0, 0, 0

screen = pygame.display.set_mode(size)

ball = pygame.image.load("intro_ball.gif")
#ball = pygame.image.load("bird-animated-gif-26-665736874.gif")
#ball = pygame.image.load("oiseau001.gif")
```

```
ballrect = ball.get_rect()

while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT: sys.exit()

    ballrect = ballrect.move(speed)
    if ballrect.left < 0 or ballrect.right > width:
        speed[0] = -speed[0]
        time.sleep(0.2)
    if ballrect.top < 0 or ballrect.bottom > height:
        speed[1] = -speed[1]
        time.sleep(0.2)

    screen.fill(black)
    screen.blit(ball, ballrect)
    pygame.display.flip()
```

Images à telecharger dans le meme repertoire que le fichier python



Installer Pygame

Avant d'installer Pygame, nous devons avoir Python sur notre machine. Sur quelques systèmes d'exploitation, Python est installé par défaut. Ainsi, Pygame devrait être compatible avec toutes les versions de Python et il est importons à mentionner que nous aurons également besoin de la lirairie NumPy.

Installation sur linux

- Sur linux, installez pygame en utilisant cette commande:

```
sudo apt-get install python-pygame
```

- Pygame peut être trouvé dans les archives Debian

```
https://installati.one/install-python3-pygame-sdl2-debian-12/
```

Nous pouvons installer NumPy avec la commande suivante:

```
sudo apt-get install python-numpy
```

Installation sur Windows

Le programme d'installation de Windows Python se trouve sur www.python.org/download. Sur ce site, nous pouvons également trouver un outil d'installation pour Mac OS X et archives tar sources pour Linux, Unix et Mac OS X.

- Depuis le site Web de Pygame :

Accédez à http://www.pygame.org/download.shtml, vous pouvez télécharger l'installateur binaire approprié pour la version Python que nous utilisons.

- Téléchargez un programme d'installation NumPy pour Windows à partir du site web SourceForge http://sourceforge.net/projects/numpy/files/.

Depuis cmd

```
Pip install pygame
Pip install numpy
```

Python 3.9 possède 36 mots réservés.

On ne peut pas nommer une variable avec un mot réservé (levée d'une erreur).

ma_var = 2300 print("contenu de la variable ma_var = ", ma_var)

global ma_var ma_var = 2300 print("contenu de la variable ma_var = ", ma_var)

help("keywords")

import keyword print("nombre de mots-clés = ", len(keyword.kwlist)) print(keyword.kwlist)

ch = "" for nn in range(len(keyword.kwlist)):

```
ch += keyword.kwlist[nn]
if nn<len(keyword.kwlist) - 1:
    ch += ", "
```

print(ch)

nombre de mots-clés = 35

['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']

Comment créer un jeu de devinettes de nombres en Python

Dans ce projet, vous allez créer un simple jeu de devinettes qui permet à l'utilisateur de deviner un nombre aléatoire compris entre 1 et 100. Le programme donnera des indices à l'utilisateur après chaque supposition, indiquant si sa supposition était trop élevée ou trop faible, jusqu'à ce que l'utilisateur devine le bon numéro.

exe0001.py

```
import random

secret_number = random.randint(1, 100)

while True:
    guess = int(input("Deviner un nombre entre  1 et 100: "))

    if guess == secret_number:
        print("Bravo vous avez trouvez le nombre!")
        break
    elif guess < secret_number:
        print("Trop bas recommencez !.")
    else:
        print("Trop haut recommencez.")
```

Explication :

- Commencez par importer le module random, qui vous permettra de générer un nombre aléatoire.
- Générez un nombre aléatoire entre 1 et 100 à l'aide de la fonction randint() du module random, et affectez-le à une variable.
- Créez une boucle qui permet à l'utilisateur de deviner le nombre jusqu'à ce qu'il devine correctement. À l'intérieur de la boucle, invitez l'utilisateur à saisir sa supposition à l'aide de la fonction input() et convertissez son entrée en un entier à l'aide de la fonction int().
- Ajoutez une instruction conditionnelle à l'intérieur de la boucle qui vérifie si la supposition de l'utilisateur est correcte, trop élevée ou trop faible. Si la supposition est correcte, imprimez un message de félicitations et sortez de la boucle. Si la supposition est trop élevée ou trop faible, imprimez un message d'indice pour aider l'utilisateur à deviner correctement le nombre.
- Exécutez le programme et jouez au jeu de devinettes !

Comment créer un générateur de mot de passe simple en Python

Un générateur de mot de passe, comme son nom l'indique, génère un mot de passe aléatoire d'une longueur particulière en utilisant différentes combinaisons de caractères et de caractères spéciaux.

exe002.py

```
import random
import string

def generate_password(length):
    """Cette fonction génère un mot de passe aléatoire
    d'une longueur donnée en utilisant une combinaison de
    lettres majuscules, de lettres minuscules,
    de chiffres et de caractères spéciaux"""

    #Define a string containing all possible characters
    all_chars = string.ascii_letters + string.digits +
string.punctuation

    # Generate a password using a random selection of characters
    password = "".join(random.choice(all_chars) for i in range(length))

    return password

# Test the function by generating a password of length 16
password = generate_password(16)
print(password)
```

Explication :

- Nous importons les modules random et string que nous utilisons respectivement pour générer des valeurs aléatoires et travailler avec des chaînes.
- Ensuite, nous définissons une fonction appelée generate_password qui prend un seul paramètre length, qui spécifie la longueur du mot de passe qui doit être généré.
- À l'intérieur de la fonction, nous définissons une chaîne appelée all_chars qui contient tous les caractères possibles pouvant être utilisés pour générer le mot de passe. Nous utilisons les constantes string.ascii_letters, string.digits et string.uccion pour créer cette chaîne.
- Nous utilisons ensuite la compréhension de liste pour générer une liste de caractères aléatoires length à partir de la chaîne all_chars en utilisant la fonction random.choice(). Enfin, nous joignons ces caractères en une seule chaîne à l'aide de la fonction “”.join() et renvoyons le résultat.
- Pour tester la fonction, nous l'appelons avec un argument de 16 pour générer un mot de passe de longueur 16 et imprimer le résultat.

Notez qu'il s'agit d'un générateur de mot de passe très simple et qu'il peut ne pas convenir à une utilisation dans des scénarios réels où la sécurité est une préoccupation.