18. ΚΑΛΟΚΑΙΡΙΝΑ PROJECTS Ι ΚΑΙ ΜΕΤΑΤΡΟΠΗ PY-TO-EXE

18.0 Εισαγωγή

Το καλοκαιράκι έχει μπει για τα καλά και είναι η ώρα να ασχοληθούμε με κάτι πιο ευχάριστο και πιο καλοκαιρινό.

Ας φτιάξουμε λοιπόν 2 projects μαζί:

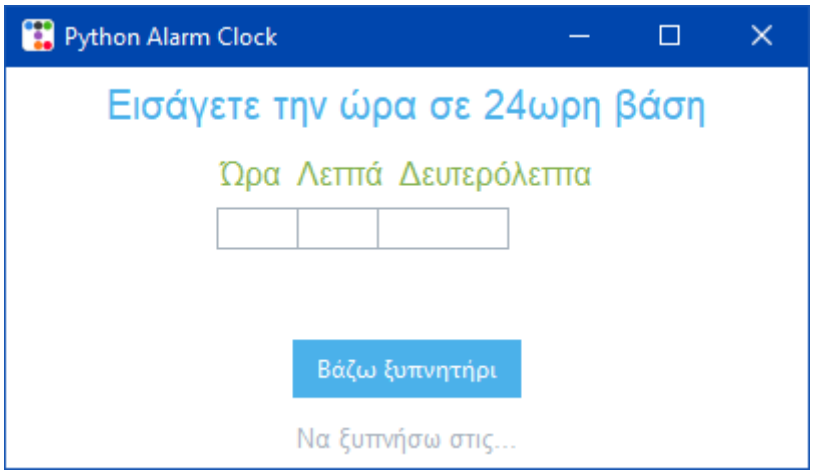
- **Python alarm clock** (ένα ξυπνητήρι με το ttkbootstrap)
- **Story game** (ένα αστείο παιχνίδι κειμένου, δημιουργίας ιστοριών)

18.1.0 Python Alarm Clock

Σαν νέοι προγραμματιστές, είναι σημαντικό να εργαζόμαστε με έναν τρόπο πρακτικό κι ευχάριστο, εμπλουτίζοντας έτσι πιο γρήγορα, πιο πρακτικά και πιο ευχάριστα τις γνώσεις μας.

Για να κατακτήσουμε μια γλώσσα προγραμματισμού, θα πρέπει να εργαστούμε σε έργα. Η δημιουργία έργων μας δίνει πρώτα απ' όλα τη χαρά της δημιουργίας, αλλά και μας βάζει σε μια νοοτροπία προγραμματιστική, στην οποία καταλαβαίνουμε ότι ο προγραμματισμός είναι μια δεξιότητα που απαιτεί συγκεκριμένο κι έξυπνο τρόπο σκέψης, γνώση των βιβλιοθηκών και ολοκλήρωση σε

συγκεκριμένο χρόνο. Πάμε λοιπόν να φτιάξουμε ένα ξυπνητήρι. Το τελικό μας αποτέλεσμα θα είναι το παρακάτω:



18.1.1 Στόχος

Στόχος μας είναι να φτιάξουμε ένα ξυπνητήρι σαν το παραπάνω. Βέβαια, όταν ξεκινάμε κάτι, δεν έχουμε την απεικόνιση του τελικού αποτελέσματος, παρά μόνο στο νου μας. Ήδη υπάρχει δηλαδή ένα νοητικό κατασκεύασμα, το οποίο πρέπει να υλοποιήσουμε γράφοντας τον αντίστοιχο κώδικα και δίνοντάς του την αντίστοιχη μορφή.

18.1.2 Προαπαιτούμενα

Αυτό το έργο απαιτεί καλή γνώση της Python καθώς και της δημιουργίας GUI (Graphic User Interface). Η Python όταν συνδυάζεται με το Tkinter και ακόμα καλύτερα με το ttkbootstrap που μάθαμε τελευταία, παρέχει έναν γρήγορο και εύκολο τρόπο δημιουργίας εφαρμογών GUI.

Το Tkinter με το ttkbootstrap παρέχουν όμορφες και εύκολα παραμετροποιήσιμες γραφικές διεπαφές χρήστη.

Δεν χρειάζεται να κατεβάσουμε ή να χρησιμοποιήσουμε κάτι που δεν έχει ενσωματωμένο η Python, όμως, πρέπει να αφιερώσουμε ελάχιστο χρόνο για να δούμε βιβλιοθήκες συνεργασίας της γλώσσας

με το λειτουργικό μας σύστημα, όπως τις “date, datetime και winsound” τις οποίες και εισάγουμε στην αρχή του project.

18.1.3 Δομή έργου

Κάθε έργο μας, από αυτό το μέγεθος και πάνω, θα πρέπει να έχει μια δομή, ώστε το «νοητικό μας κατασκεύασμα» που αναφέραμε παραπάνω, να αρχίσει να παίρνει μορφή.

Μια υλοποίηση του ξυπνητηριού, θα μπορούσε να χρειάζεται τα παρακάτω βήματα για να υλοποιηθεί:

Αρχικά, ας ελέγξουμε τα βήματα για τη δημιουργία ενός προγράμματος Ξυπνητήρι στην Python:

- Εισαγωγή όλων των βιβλιοθηκών και ενοτήτων που απαιτούνται
- Ένας βρόχος while που παίρνει το όρισμα της ώρας, στην οποία ο χρήστης θέλει να βάλει το ξυπνητήρι να χτυπήσει και διακόπτεται αυτόματα όταν αυτή η ώρα φτάσει, με ήχο
- Γραφικό περιβάλλον με το ttkbootstrap.

18.1.4 Κατασκευή – Περιγραφή – Σχόλια

Ξεκινάμε την κατασκευή του προγράμματός μας με την εισαγωγή των απαραίτητων βιβλιοθηκών:

```
from tkinter import *
import ttkbootstrap as ttk
from ttkbootstrap.constants import *

# themes = cosmo, flatly, journal, litera, lumen, minty, pulse, sandstone,
# united, yeti, morph, simplex, cerculean

import datetime
```

import time

import winsound

Εισάγουμε τα πάντα από το Tkinter, καθώς και το ttkbootstrap με τις σταθερές του (SUCCESS, LEFT κ.λ.π.).

Τα modules ***datetime*** και ***time*** μάς βοηθούν να εργαστούμε με τις ημερομηνίες και την ώρα της τρέχουσας ημέρας στην οποία ο χρήστης θα χρειαστεί το ξυπνητήρι.
Το module winsound παρέχει πρόσβαση στη χρήση βιβλιοθηκών ήχου των Windows. Αυτό θα μας είναι χρήσιμο για τη «χτύπημα του ξυπνητηριού» μόλις ικανοποιηθεί η συνθήκη και κληθεί η συνάρτησή μας.
Πληροφορίες για την datetime, μπορείτε να βρείτε εδώ:
https://docs.python.org/3/library/datetime.html

18.1.5 Βρόχος while – Η «μηχανή» του ξυπνητηριού

Πάμε να δούμε πως θα υλοποιήσουμε το ξυπνητήρι μας. Μπορούμε να σκεφτούμε: Όσο ο χρόνος μετράει και πλησιάζει τον χρόνο της αφύπνισης, μην κάνεις τίποτα. Όταν ο χρόνος της αφύπνισης, γίνει ίσος με τον χρόνο του τώρα, (της στιγμής), χτύπα το ξυπνητήρι.

Δηλαδή, τύπωσε ένα μήνυμα και κάλεσε τον ήχο του ξυπνητηριού.

Για να τα κάνουμε όλα αυτά, χρειάζεται να παίρνουμε τον χρόνο του τώρα και μπορούμε επίσης, για να έχουμε έναν έλεγχο του τι συμβαίνει, να τυπώνουμε τον χρόνο, παρατηρώντας ότι πλησιάζει τον χρόνο αφύπνισης. Έτσι, μπορούμε και εύκολα να αποσφαλματώσουμε το προγραμματάκι μας:

def alarm(set_alarm_timer): # Η παράμετρος εισάγεται από την actual_time()

while True: # Όσο δεν συμβαινει τίποτα, τύπωνε στην κονσόλα την ημ/νία και την ώρα

```
        time.sleep(1) # Ο χρόνος θα μετράει με καθυστέρηση ενός δευτ/του
        current_time = datetime.datetime.now()
        now = current_time.strftime("%H:%M:%S")
        date = current_time.strftime("%d/%m/%Y")
        print("Η ημερομηνία είναι:",date) # Εκτύπωση στην κονσόλα (IDLE)
        print(now) # Εκτύπωση στην κονσόλα (IDLE)
        if now == set_alarm_timer: # Όταν η ώρα γίνει ίση με την ώρα που έχει ορίσει ο χρήστης
            print("Ώρα να ξυπνήσεις!") # τύπωσε μήνυμα στην κονσόλα
            # winsound.PlaySound("sound.wav",winsound.SND_ASYNC) # Εδώ έχουμε έναν κλασικό ήχο των Windows
            winsound.PlaySound("D://Cookoo_Home/Λήψεις/mixkit-retro-game-emergency-alarm-1000.wav",winsound.SND_FILENAME) # και κάλεσε τον ήχο του ξυπνητηριού (ήχος που έχουμε κατεβάσει)
            break # βγες από τον βρόχο (η λειτουργία του ξυπνητηριού ολοκληρώθηκε)
```

Βέβαια, παρατηρούμε ότι η παράμετρος ***set_alarm_timer*** εισήχθη από την ***activate_alarm*** την οποία και πρέπει να δημιουργήσουμε:

```
def activate_alarm(): # Παίρνει τον χρόνο αφύπνισης και "καλεί" τη "μηχανή" που θα χτυπήσει το ξυπνητήρι
    set_alarm_timer = f"{hour.get()}:{min.get()}:{sec.get()}"
    alarm(set_alarm_timer) # Κλήση της μηχανής του ξυπνητηριού
```

18.1.6 Δημιουργία γραφικού περιβάλλοντος

Έχουμε αναφερθεί αρκετά σ' αυτό το κομμάτι στα προηγούμενα μαθήματα στα οποία και μπορείτε να ανατρέξετε. Μια εύκολη

παραλλαγή του GUI, μπορούμε να δημιουργήσουμε απλά αλλάζοντας το themename (όλες οι ονομασίες που μπορούμε να χρησιμοποιήσουμε, σας παρέχονται για ευκολία σας στα σχόλια).

 # Δημιουργία γραφικού περιβάλλοντος

clock = ttk.Window(themename="cerculean")

clock.title("Python Alarm Clock")

#clock.iconbitmap(r"python.ico") # Μπορούμε να βάλουμε το δικό μας εικονίδιο στο παράθυρο

clock.geometry("400x200")

time_format=ttk.Label(clock, text= "Εισάγετε την ώρα σε 24ωρη βάση", font=('Helvetica', 15), bootstyle = PRIMARY).pack(side = TOP, padx=5, pady=5)

addTime = ttk.Label(clock, text = "Ώρα Λεπτά Δευτερόλεπτα", font=('Helvetica', 12), bootstyle = SUCCESS).pack(side = TOP, padx=5, pady=5)

setYourAlarm = ttk.Label(clock, text = "Να ξυπνήσω στις...", font=('Helvetica', 10), bootstyle = SECONDARY).pack(side = BOTTOM, padx=5, pady=5)

18.1.7 Μεταβλητές και ολοκλήρωση του προγράμματος

Χρειαζόμαστε κάποιες μεταβλητές οι οποίες θα αποθηκεύουν τον χρονο που θα μας δίνει ο χρήστης. Αυτές πρέπει να είναι τύπου StringVar(), καθότι μ’ αυτό τον τύπο λειτουργούν και τα modules datetime και date, παραπάνω.

 # Οι μεταβλητές που χρειαζόμαστε για το ξυνητήρι (αρχικοποίηση):

hour = StringVar()

min = StringVar()

sec = StringVar()

Κατόπιν, φτιάχνουμε 3 πεδία (Entry) στο παράθυρο, ώστε ο χρήστης να μπορει να ορίζει την ώρα αφύπνισης. Μπορούμε εδώ, να πειραματιστούμε και πάλι με τα ttk widgets, ή να αφήσουμε την κλασική έκδοση του Tk().

 # Ώρα στην οποία βάζουμε το ξυπνητήρι να χτυπήσει

hourTime= Entry(clock,textvariable = hour, width = 10).place(x=105,y=70)

minTime= Entry(clock,textvariable = min, width = 10).place(x=145,y=70)

secTime = Entry(clock,textvariable = sec, width = 10).place(x=185,y=70)

Ολοκληρώνουμε με τη δημιουργία του αντίστοιχου κουμπιού και το απαραίτητο συνεχές ***mainloop():***

 # Το ξυπνητήρι μπαίνει σε λειτουργία με το πάτημα του κουμπιού

submit = ttk.Button(clock,text = "Βάζω ξυπνητήρι", width = 15, command = activate_alarm).pack(side = BOTTOM, padx=5, pady=5)

clock.mainloop()

ΤΟ ΞΥΠΝΗΤΗΡΙ ΜΑΣ ΕΙΝΑΙ ΕΤΟΙΜΟ!

18.1.8 Τι μάθαμε

- *Μάθαμε πως να χρησιμοποιούμε την ώρα του συστήματος στα προγράμματά μας.*
- *Μάθαμε πώς να αναζητούμε και να βάζουμε δικούς μας ήχου και εικονίδια στο πρόγραμμα μας.*

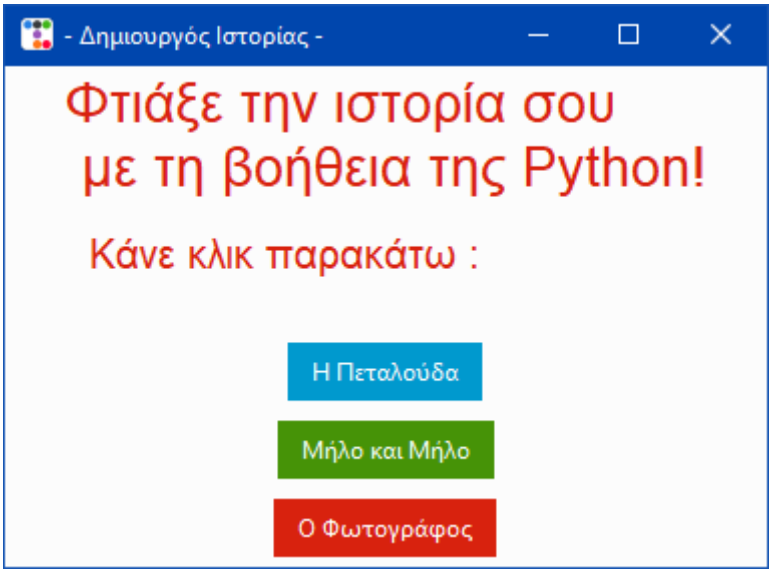
- *Τέλος, μάθαμε πώς να δομούμε ένα πρόγραμμα. Αυτό φυσικά αποκτάται όλο και καλύτερα με την εμπειρία.*

18.2.0 Story Game

Το παιχνίδι μας θα είναι ένα παιχνίδι κειμένου. Αυτός ο τύπος παιχνιδιών, ήταν δημοφιλής πριν από πολλά χρόνια. Τότε που τα γραφικά και η «δράση», δεν ήταν δυνατόν να υποστηριχτούν από το hardware.

Πώς λειτουργεί το παιχνίδι: Υπάρχει ένα μικρό κείμενο με κενά και ο χρήστης καλείται να συμπληρώσει κάποια στοιχεία, τα οποία αργότερα θα συμπληρώσουν τα κενά του προϋπάρχοντος κειμένου. Ο χρήστης δεν γνωρίζει που ακριβώς θα μπουν τα στοιχεία που δίνει. Αυτό άλλωστε είναι και το διασκεδαστικό όφελος του παιχνιδιού.

Το παράθυρο μετά την ολοκλήρωση του παιχνιδιού θα είναι το παρακάτω:



18.2.1 Στόχος

Στόχος μας είναι να φτιάξουμε 3 μικρά σενάρια και να ορίσουμε «κενά» σημεία, στα οποία εμείς να ζητάμε από τον χρήστη να μας δίνει στοιχεία και κατόπιν όλα αυτά τα στοιχεία, να τυπώνονται στην κονσόλα σαν μια ενιαία ιστορία.

Είναι ένα project σχετικά εύκολο στην υλοποίηση και περισσότερο θα πρέπει να σκεφτούμε τις ιστοριούλες, καθώς και τα «κενά» που θα δημιουργήσουμε, ώστε η ιστορία να έχει ενδιαφέρον.

Ο χρήστης επιλέγει την ιστορία που θέλει να συμπληρώσει, πατώντας στο αντίστοιχο κουμπί και κατόπιν, του ζητούνται κάποια δεδομένα, όπως ένα «όνομα» ένα «ρήμα» κ.λ.π.

Η ελληνική γλώσσα δεν προσφέρεται για τέτοιου είδους παιχνίδια καθότι έχει πολλές πτώσεις, πρόσωπα και ιδιαιτερότητες, με αποτέλεσμα στην τελική ιστορία να υπάρχουν γραμματικά λάθη, όμως αυτό μας βάζει να γινόμαστε ακόμα πιο λεπτομερείς και ακριβείς σ' αυτό που ζητάμε από τον χρήστη (πχ «συμπληρώστε α' πληθυντικό πρόσωπο»), αλλά και στο πως συντάσσουμε την ιστορία.

18.2.2 Προαπαιτούμενα

Δεν χρειαζόμαστε κάτι περισσότερο απ' αυτό που ήδη έχουμε εγκαταστήσει μέχρι τώρα. Όλα τα modules που χρειαζόμαστε είναι ήδη εγκατεστημένα και χρειάζεται απλά να τα εισάγουμε

18.2.3 Δομή έργου

Το έργο μας θα έχει την παρακάτω απλή δομή:

- **Εισαγωγή modules**
- **Δημιουργία παραθύρου**
- **Καθορισμός συναρτήσεων**
- **Δημιουργία γραφικού περιβάλλοντος**

18.2.4 Κατασκευή έργου – Περιγραφή – Σχόλια

Πρώτα απ’ όλα, καθορίζουμε τις ιστοριούλες μια-μια. Ονομάζουμε τη συνάρτηση που θα χρειαστούμε με το όνομα της ιστορίας, και φτιάχνουμε ένα print() με το οποίο θα τυπώνουμε την ιστορία.

Τώρα, μπορούμε να καθορίσουμε κάποια κενά, στα οποία θα ζητάμε από τον χρήστη να συμπληρώσει τα στοιχεία.

Έτσι, καθορίζουμε τις αντίστοιχες μεταβλητές, ζητάμε και παίρνουμε τα δεδομένα από τον χρήστη και με το τελευταίο στοιχείο τυπώνουμε όλη την ιστορία μας. Επαναλαμβάνουμε τα βήματα για κάθε ιστορία.

18.2.5 Εισαγωγή των modules

Μιας και το παράθυρό μας είναι αρκετά απλό, δεν χρειάζεται καν να εισάγουμε το tkinter:

import ttkbootstrap as ttk

from ttkbootstrap.constants import *

18.2.6 Δημιουργία του παραθύρου

Δημιουργούμε και αρχικοποιούμε το παράθυρο, καθώς και κάποια βασικά widgets του. Περιλαμβάνουμε τα themes του ttkbootstrap για ευκολία πειραματισμών:

 # themes = cosmo, flatly, journal, litera, lumen, minty, pulse, sandstone,

 # united, yeti, morph, simplex, cerculean

root = ttk.Window(themename="simplex") # root είναι το όνομα του παραθύρου μας

root.geometry('380x250') # Διαστάσεις παραθύρου

root.title('- Δημιουργός Ιστορίας -') # Ο τίτλος του παιχνιδιού μας

label1 = ttk.Label(root, text= 'Φτιάξε την ιστορία σου \n με τη βοήθεια της Python!', font = ('Helvetica', 20), bootstyle = PRIMARY).pack() # Κείμενο μέσα στο παράθυρο

label2 = ttk.Label(root, text = 'Κάνε κλικ παρακάτω :', font=('Helvetica', 15), bootstyle = PRIMARY).place(x=40, y=80) # Κείμενο μέσα στο παράθυρο

18.2.7 Καθορισμός συναρτήσεων

Κάθε συνάρτηση είναι μια πλήρης ιστορία και συνδέεται με ένα κουμπί. Θα μπορούσαμε να έχουμε περισσότερες ιστορίες ή και μόνο μια. Σε κάθε συνάρτηση βρίσκεται το κύριο μέρος του προγράμματός μας. Δίνουμε σαφείς οδηγίες στο τι ακριβώς ζητάμε από τον χρήστη.

Πάμε να δούμε τις συναρτήσεις:

##################- Ιστορίες -###############

def photographer():

animals= input('Εισάγετε όνομα ζώου : ')

profession = input('Εισάγετε ένα επάγγελμα (στην αιτιατική): ')

cloth = input('Εισάγετε όνομα υφάσματος: ')

things = input('Εισάγετε όνομα αντικειμένων: ')

name= input('Εισάγετε όνομα με άρθρο: ')

place = input('Εισάγετε τόπο: ')

verb = input('Εισάγετε ρήμα σε ενεστώτα, γ πληθυντικό: ')

food = input('Εισάγετε όνομα τροφής: ')

print('Πες ' + food + ', είπε ο φωτογράφος καθώς το flash άναψε! ' + name + ' κι εγώ πήγαμε στο/στη ' + place +' σήμερα, για να πάρουμε τις φωτογραφίες μας. Η πρώτη φωτογραφία την οποία θέλαμε να δούμε, ήταν αυτή που είμαστε ντυμένοι ' + animals + ' παριστάνοντας έναν/μια ' + profession + ' .Όταν είδαμε τη δεύτερη φωτογραφία, ήταν ακριβώς αυτό που θέλαμε. Μοιάζαμε και οι δύο με ' + things + ' που φοράνε ' + cloth + ' και ' + verb + ' --ακριβώς ότι είχα στο μυαλό μου')

```
def butterfly():

    adjective = input('Εισάγετε ένα επίθετο (θηλυκό) : ')
    color = input('Εισάγετε ένα χρώμα : ')
    thing = input('Εισάγετε ένα αντικείμενο :')
    place = input('Εισάγετε μια πόλη : ')
    person= input('Εισάγετε ένα όνομα (αιτιατική): ')
    adjective1 = input('Εισάγετε ένα επίθετο : ')
    insect= input('Εισάγετε ένα έντομο : ')
    food = input('Εισάγετε μια τροφή : ')
    verb = input('Εισάγετε ένα ρήμα (α πληθυντικό): ')

    print('Χθες το βράδυ ονειρεύτηκα ότι ήμουν μια ' +adjective+ '
πεταλούδα με ' + color+ ' φτερά που έμοιαζαν με '+thing+ '. Πετούσα
για τον/την' + place+ ' με τον καλύτερό μου φίλο και τον/την '+person+
' που ήταν ένας/μια '+adjective1+ ' ' +insect +' . Φάγαμε λίγο/η '
+food+ ' μόλις φτάσαμε και αποφασίσαμε να '+verb+ ' και το όνειρο
τέλειωσε όταν είπα "Ας ' +verb+ '"')



def appleandapple():
    person = input('Εισάγετε ένα όνομα (σε γενική πτώση): ')
    color = input('Εισάγετε ένα χρώμα (πληθυντικός) : ')
    foods = input('Εισάγετε μια τροφή (γενική) : ')
    adjective = input('Εισάγετε ένα επίθετο (ουδέτερο ενικός): ')
    thing = input('Εισάγετε ένα αντικείμενο : ')
```

```
place = input('Εισάγετε μια πόλη : ')
verb = input('Εισάγετε ένα ρήμα (ενεστώτας, α πληθυντικό): ')
adverb = input('Εισάγετε ένα επίρρημα : ')
food = input('Εισάγετε μια τροφή: ')
things = input('Εισάγετε ένα αντικείμενο : ')

print('Σήμερα πήραμε μήλα από το περιβόλι του/της '+person+ '. Δεν είχα ιδέα πως υπάρχουν πολλές ποικιλίες μήλων. Έφαγα ' +color+ ' μήλα κατευθείαν από το δέντρο και είχαν γεύση '+foods+ '. Μετά είδαμε ένα '+adjective+ ' μήλο που έμοιαζε με ' + thing + '. Όταν η τσάντα μας γέμισε, πήγαμε μια βόλτα στον/στην '+place+ ' και ξαναγυρίσαμε πίσω. Τελειώσαμε τη βόλτα μας αφού πέσαμε (ευτυχώς) σε μια στοίβα άχυρα και μετά έπρεπε να ' +verb+ ' ' +adverb+ '. Δεν κρατιέμαι να φτάσω σπίτι και να μαγειρέψω με τα μήλα. Θα κάνουμε μηλο- '+food+ ' και '+things+' -πιτες!.')
```

18.2.8 Καθορισμός υπόλοιπων γραφικών στοιχείων (κουμπιών)

Το μόνο που μας λείπει τώρα είναι τα κουμπιά. Όπως είπαμε, κάθε κουμπί είναι η αρχή μιας ιστορίας την οποία με το πάτημά του, επιλέγει ο χρήστης. Περιλαμβάνουμε το σύνολο των στυλ του ttkbootstrap για πειραματισμό. Πάμε λοιπόν να τα φτιάξουμε:

```
# Για πιο όμορφα κουμπιά, χρησιμοποιούμε το ttkbootstrap
# ttkbootstrap styles - PRIMARY, SECONDARY, SUCCESS, INFO, WARNING, DANGER, LIGHT, DARK
b1 = ttk.Button(root, text= 'Ο Φωτογράφος', bootstyle = PRIMARY, command = photographer)
```

b1.pack(side = BOTTOM, padx=5, pady=5)

b2 = ttk.Button(root, text= 'Μήλο και Μήλο', bootstyle = SUCCESS, command = appleandapple)

b2.pack(side = BOTTOM, padx=5, pady=5)

b3 = ttk.Button(root, text= 'Η Πεταλούδα', bootstyle = INFO, command = butterfly)

b3.pack(side = BOTTOM, padx=5, pady=5)

root.mainloop()

ΤΟ ΠΑΙΧΝΙΔΙ ΙΣΤΟΡΙΩΝ ΜΑΣ ΕΙΝΑΙ ΕΤΟΙΜΟ!

18.2.9 Τι μάθαμε

Χρησιμοποιήσαμε το Tkinter καθώς και το ttkbootstrap, αποκτώντας όλο και περισσότερη εμπειρία.

Ξεσκονίσαμε τις γνώσεις μας για την εισαγωγή κειμένου από τον χρήστη καθώς και για εκτύπωση

Σκεφτήκαμε πώς να δομήσουμε το πρόγραμμα μας καθώς και ποιες ιστορίες να δημιουργήσουμε.

18.3.0 Μετατροπή αρχείων py-to-exe

Είναι επόμενο πως σαν χρήστες Windows, θα μας εξυπηρετούσε κάποιες φορές να έχουμε διαθέσιμα αρχεία executables, δηλαδή με κατάληξη .exe, που όπως γνωρίζετε είναι η κατάληξη των εκτελέσιμων αρχείων των Windows.

18.3.1 Προσθήκη της Python στο Windows path

Αν δεν το έχουμε κάνει ήδη, μπορούμε να το κάνουμε τώρα, είτε επανεγκαθιστώντας την Python, (ή εγκαθιστώντας μια νεότερη έκδοση) και προσέχοντας κατά την εγκατάσταση να κάνουμε τικ στο ειδικό κουτάκι το οποίο μας ζητάει την άδεια για να «Add Python to Windows path", είτε χειροκίνητα, ακολουθώντας τα παρακάτω βήματα:

Πατάμε WinKey + R

Πληκτρολογούμε **sysdm.cpl**

Πάμε στην καρτέλα «Για προχωρημένους»

Επιλέγουμε «Μεταβλητές περιβάλλοντος»

Κάτω από το πεδίο Μεταβλητές χρήστη, πατάμε «Δημιουργία» και μας παρουσιάζεται ένα παραθυράκι στο οποίο πρέπει να συμπληρώσουμε 2 πεδία.

Πρέπει να γνωρίζουμε το μονοπάτι της Python, αν όχι, μπορούμε από τον IDLE, να κάνουμε κλικ στο «Αρχείο» → «Άνοιγμα» και να κάνουμε αντιγραφή το path από το παραθυράκι που μας ανοίγει, ακυρώνοντας κατόπιν το άνοιγμα αρχείου.

Θα χρειαστούμε 2 paths: Το application path και το scripts path. Το πρώτο θα είναι όπως παρακάτω, δηλαδή:

C:\Users\User\AppData\Local\Programs\Python\Python311

ενώ το δεύτερο:

C:\Users\User\AppData\Local\Programs\Python\Python311\scripts

Στο πεδίο Όνομα μεταβλητής πληκτρολογούμε "**Path**" και στο πεδίο «**Τιμή μεταβλητής**», βάζουμε και τα δύο paths, χωρισμένα με ελληνικό ερωτηματικό, δηλαδή:

C:\Users\User\AppData\Local\Programs\Python\Python311;
C:\Users\User\AppData\Local\Programs\Python\Python311\scripts

Πατάμε ΟΚ και ξανά ΟΚ κι έχουμε ολοκληρώσει.

18.3.2 Εγκατάσταση του πακέτου PyInstaller

Ανοίγουμε το command prompt και πληκτρολογούμε:

pip install pyinstaller

18.3.3 Δημιουργία εκτελέσιμου αρχείου

Πάμε τώρα να δημιουργήσουμε το εκτελέσιμο αρχείο:

Πλοηγούμαστε στον φάκελο όπου βρίσκεται το αρχείο .py, το οποίο θέλουμε να μετατρέψουμε επίσης σε .exe. Αυτό το κάνουμε στο command prompt των Windows, πληκτρολογώντας cd και το path, δηλαδή:

cd C:\Users\User\AppData\Local\Programs\Python\Python311

Τώρα, δίνουμε την εντολή:

pyinstaller –onefile onomaarxeiou.py

και πατάμε το Enter.

Αφού η διαδικασία ολοκληρωθεί, θα βρούμε το εκτελέσιμό μας στον φάκελο dist με το όνομα ***onomaarxeiou.exe***.

ΚΑΛΟ ΚΑΛΟΚΑΙΡΙ ΠΑΙΖΟΝΤΑΣ ΜΕ ΤΗΝ PYTHON!