

ΕΙΣΑΓΩΓΗ

Καλωσορίσατε στο βιβλίο «Εκπαιδευτικές εφαρμογές στη γλώσσα προγραμματισμού Python». Αυτό το βιβλίο έχει σχεδιαστεί ειδικά για όσους θέλουν να εξερευνήσουν την ευρεία γκάμα εφαρμογών που προσφέρει η Python, μία από τις πιο δημοφιλείς και φιλικές προς τον χρήστη γλώσσες προγραμματισμού.

Εάν αναρωτιέστε γιατί θα πρέπει να επιλέξετε αυτή τη γλώσσα προγραμματισμού, το πρώτο κεφάλαιο σας προσφέρει μια απάντηση. Θα ανακαλύψετε τα πλεονεκτήματα της Python σε σχέση με άλλες γλώσσες προγραμματισμού και θα εξετάσετε τον τρόπο εγκατάστασης του περιβάλλοντος ανάπτυξης που σας επιτρέπει να γράψετε κώδικα Python με άνεση, το PyCharm.

Το βιβλίο απαρτίζεται από κεφάλαια που περιέχουν μια ευρεία γκάμα εφαρμογών και θεματολογιών. Στο κεφάλαιο 2, εστιάζουμε στη λειτουργία μέσω εφαρμογών, με παραδείγματα που καλύπτουν την αλληλεπίδραση με διαδικτυακές εφαρμογές, τη δημιουργία και αναγνώριση κωδικών QR και την κρυπτογράφηση δεδομένων.

Στο κεφάλαιο 3, εξερευνούμε τις εφαρμογές που συνδέονται με τη δημιουργία ψηφιακών ημερολογίων, την εκτέλεση μετρήσεων ημερομηνιών, τη χρήση της βιβλιοθήκης Pygame για τη δημιουργία παιχνιδιών και την κρυπτογράφηση και αποκρυπτογράφηση αρχείων.

Στο κεφάλαιο 4, εστιάζουμε στις εφαρμογές που αναλαμβάνουν διάφορες εργασίες όπως μετάφραση, μετατροπή video σε GIF, ψηφιακό ρολόι, ειδοποιήσεις και δημιουργία κωδικών πρόσβασης σε αρχεία PDF.

Στο κεφάλαιο 5, ασχολούμαστε με εφαρμογές τεχνητής νοημοσύνης και ρομποτικής, με έμφαση στην αλληλεπίδραση με drone.

Στο κεφάλαιο 6, εστιάζουμε στον έλεγχο και τη λειτουργία των drone και παρουσιάζουμε εφαρμογές που αφορούν την αυτόματη πτήση, την ανίχνευση αντικειμένων, τον χειρωνακτικό έλεγχο με τη χρήση όρασης υπολογιστών, τον έλεγχο πρόσβασης και τη λήψη φωτογραφιών και βίντεο από τον αέρα.

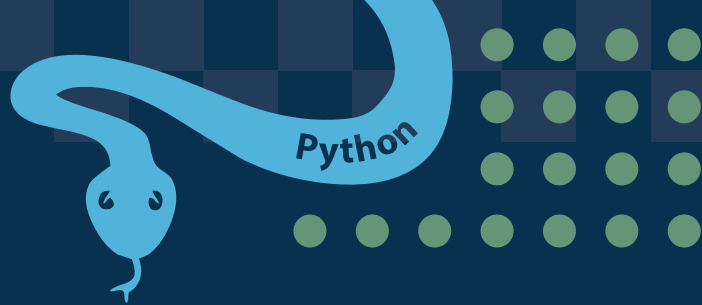
Το βιβλίο συνεχίζεται με τα κεφάλαια 7 έως 13, τα οποία περιέχουν αναλυτικές περιγραφές και ασκήσεις πάνω σε θέματα όπως οι γραφικές αναπαραστάσεις δεδομένων, ο προγραμματισμός δικτυακών εφαρμογών, η δημιουργία ιστοσελίδων με τη χρήση των PyScript, Streamlit και Flask, η οπτικοποίηση δεδομένων, η αναγνώριση χαρακτήρων OCR και πολλά άλλα.

Αυτό το βιβλίο αποτελεί έναν πολύτιμο οδηγό για όσους επιθυμούν να εκμεταλλευτούν τις δυνατότητες της Python για την ανάπτυξη εκπαιδευτικών εφαρμογών. Με κατανοητές εξηγήσεις, παραδείγματα κώδικα και ασκήσεις, θα σας βοηθήσει να ενταχθείτε στον χώρο της προγραμματιστικής κοινότητας και να αναπτύξετε τις δικές σας εφαρμογές με επιτυχία.

Είμαστε ενθουσιασμένοι που μπορούμε να σας καθοδηγήσουμε σε αυτό το ταξίδι προγραμματισμού και ευελπιστούμε ότι αυτό το βιβλίο θα σας εμπνεύσει και θα σας παράσχει τα απαραίτητα εργαλεία για να αναπτύξετε τις δικές σας εκπαιδευτικές εφαρμογές στη γλώσσα προγραμματισμού Python.

Καλή ανάγνωση και καλή διασκέδαση στον κόσμο της Python!

Με εκτίμηση,
Νικόλαος Σηφάκης



*Το παράδοξο της γνώσης:
Όσο περισσότερο μελετάς,
τόσο ανακαλύπτεις την αμάθειά σου.*

- Αριστοτέλης -



Εισαγωγή στη γλώσσα προγραμματισμού Python

Στο κεφάλαιο αυτό, παρουσιάζεται μια εισαγωγή στην πιο δημοφιλή και φιλική στον χρήστη γλώσσα προγραμματισμού, την Python. Ταυτόχρονα, απαντάται η απορία που έχει κάθε αρχάριος όταν θέλει να μάθει μια γλώσσα προγραμματισμού: «Γιατί να μάθω την Python και όχι κάποια άλλη γλώσσα προγραμματισμού;». Στη συνέχεια, αναλύεται ο τρόπος εγκατάστασης της Python σε περιβάλλον Windows, καθώς και η εγκατάσταση του ολοκληρωμένου περιβάλλοντος ανάπτυξης (IDE) κώδικα Python, το PyCharm.

1.1 ΕΙΣΑΓΩΓΗ

Γεια σας και καλώς ήρθατε στο βιβλίο «Εφαρμογές στη Γλώσσα Προγραμματισμού Python». Ελπίζω ότι είστε έτοιμοι να μάθετε γιατί οι περισσότεροι επαγγελματίες και ερασιτέχνες προγραμματιστές ελκύονται από την Python. Το βιβλίο αυτό απευθύνεται σε αρχάριους που ίσως είναι εξοικειωμένοι με τον προγραμματισμό αλλά όχι με τη γλώσσα και το οικοσύστημα της Python, αλλά και σε έμπειρους προγραμματιστές που θέλουν να ανανεώσουν ή να εμπλουτίσουν τις γνώσεις τους. Αν δεν έχετε πτυχίο Πληροφορικής, μην ανησυχείτε! Ο Νικόλαος Σηφάκης θα σας εξηγήσει τις βασικές αρχές της Python αξιοποιώντας θεμελιώδεις έννοιες της Πληροφορικής και παραλείποντας αρχικά τις περιττές λεπτομέρειες.

Όταν κανείς μαθαίνει μια νέα γλώσσα προγραμματισμού, δεν έχει ακόμη την απαραίτητη εμπειρία για να κρίνει αν αυτή θα τον εξυπηρετήσει μακροπρόθεσμα. Αν προσανατολιζέστε προς την Python, επιτρέψτε μου να σας διαβεβαιώσω ότι κάνετε μια καλή επιλογή, καθώς πρόκειται για μια γλώσσα «πλήρους φάσματος». Τι σημαίνει αυτό; Η Python εμπίπτει σε εκείνες τις γλώσσες προγραμματισμού που συνιστώνται στους αρχάριους, επειδή καθιστούν τον προγραμματισμό εξαιρετικά εύκολο. Στον αντίποδα βρίσκονται, για παράδειγμα, οι οπτικές γλώσσες, όπως είναι το Scratch: <https://scratch.mit.edu/>. Το Scratch παρέχει στον χρήστη μπλοκ που αναπαριστούν έννοιες προγραμματισμού, όπως οι μεταβλητές, οι βρόχοι, οι κλήσεις μεθόδων και ούτω καθεξής, τα οποία μετακινεί και τοποθετεί στην επιφάνεια εργασίας του (λειτουργία γνωστή και ως drag and drop). Και μολονότι το Scratch είναι εύχρηστο για τη δημιουργία

απλών προγραμμάτων, δεν είναι δυνατή η κατασκευή επαγγελματικών εφαρμογών με αυτό. Δεν είναι τυχαίο ότι καμία εταιρεία του **Fortune 500** δεν στηρίζει τη βασική επιχειρηματική λογική της στον προγραμματισμό με το Scratch! Υπάρχουν άλλες γλώσσες προγραμματισμού πολύ πιο πρόσφορες για τους έμπειρους προγραμματιστές.

Οι πιο δημοφιλείς σε αυτήν την κατηγορία είναι η C και η C++. Όποιο πρόγραμμα περιήγησης στο διαδίκτυο και αν χρησιμοποιείτε σήμερα, πιθανότατα έχει γραφεί σε μία από αυτές τις δύο, ενώ μάλλον και το λειτουργικό σας σύστημα που τρέχει αυτό το πρόγραμμα περιήγησης έχει δημιουργηθεί με τη C ή τη C++. Το ίδιο ισχύει για το αγαπημένο σας ηλεκτρονικό παιχνίδι ή εφαρμογή γενικά. Αυτές οι γλώσσες προγραμματισμού παρέχουν στον χρήστη τη δυνατότητα να δημιουργεί εκπληκτικές εφαρμογές, αλλά είναι εντελώς ακατάλληλες για τους νεοεισερχόμενους που χρειάζονται ήπια εισαγωγή. Κάποιος που δεν έχει διαβάσει πολύ κώδικα C++ ενδέχεται να νιώσει τα μάτια του να καίει! Ακολουθεί ένα παράδειγμα: https://en.wikipedia.org/wiki/Resource_acquisition_is_initialization και [urlhttps://blog.devgenius.io/7-advanced-c-concepts-you-should-know-78a5f9134be5](https://blog.devgenius.io/7-advanced-c-concepts-you-should-know-78a5f9134be5)

```
import requests import urllib3

urllib3.disable_warnings()
resp = requests.get("https://www.uoc.gr/university/chronology/40years.html",
    ↳ verify=False)
template<class T> class smart_ptr
{
T* m_ptr;
public:
template<typename... Args>
smart_ptr(Args&&... args) : m_ptr(new T(std::forward<Args>(args)...)){}
~smart_ptr() { delete m_ptr; }

smart_ptr(const smart_ptr& rhs) = delete;
smart_ptr& operator=(const smart_ptr& rhs) = delete;

smart_ptr(smart_ptr&& rhs) : m_ptr(exchange(rhs.m_ptr, nullptr)){} smart_
ptr& operator=(smart_ptr&& rhs){
if (&rhs == this) return *this;
delete m_ptr;
m_ptr = exchange(rhs.m_ptr,nullptr);
return *this;
}
T& operator*() const { return *m_ptr; } T* operator->() const { return m_
ptr; }
```

```
};
void func()
{
    auto ptr = smart_ptr<resource>(1, 2); // now ptr guarantee the release of
    resource
    // ...
}
```

Μην κάψετε τα μάτια σας! Το Scratch και η C++ δεν ανήκουν στις προαναφερθείσες γλώσσες «πλήρους φάσματος». Όσον αφορά το Scratch, είναι εύκολο να το χρησιμοποιήσετε στην αρχή προκειμένου να εξοικειωθείτε με τον προγραμματισμό, όμως θα χρειαστεί να αξιοποιήσετε μια δυνατή γλώσσα προγραμματισμού για να δημιουργήσετε πραγματικές εφαρμογές. Η χρήση της C++ οδηγεί στη δημιουργία ολοκληρωμένων ψηφιακών εφαρμογών, ωστόσο η εκμάθηση των πολύπλοκων επιλογών και δυνατοτήτων της που υποστηρίζουν τις σύνθετες εφαρμογές είναι πολύ δύσκολη. Η Python, από την άλλη πλευρά, είναι εύκολη στη χρήση. Συχνά κρίνουμε την απλότητα μιας γλώσσας βάζοντάς τη να τυπώσει τη φράση hello world.

```
print(«Hello world»)
```

Αυτό είναι μόνο. Ας δοκιμάσουμε ένα άλλο παράδειγμα, για να φανεί πόσο εύκολα μπορούμε με την Python να εκτελέσουμε πολύπλοκες ενέργειες με απλό τρόπο. Αν δεν κατανοείτε πλήρως τις ενέργειες, αγνοήστε το γεγονός και απλώς παρακολουθήστε την αλληλουχία —στα επόμενα κεφάλαια καλύπτονται αυτές και άλλες έννοιες. Τι κώδικας θα χρειαζόταν για να γράψετε ένα πρόγραμμα που θα είχε πρόσβαση σε έναν ιστότοπο, θα κατέβαζε το περιεχόμενό του στη μνήμη του υπολογιστή, και στη συνέχεια θα εμφάνιζε ένα κομμάτι αυτού του περιεχομένου στον χρήστη; Ας δούμε αυτό το παράδειγμα της Python με τη βοήθεια του πακέτου (module) requests.

```
import requests
import urllib3

urllib3.disable_warnings()

resp = requests.get("https://www.uoc.gr/university/chronology/40years.html",
, → verify=False)

html = resp.text print(html[859:1322])
```

Όταν εκτελείται ο παραπάνω κώδικας, η έξοδος είναι (προσπέλαση: 04/05/2022, το περιεχόμενο του ιστοτόπου μεταβάλλεται τακτικά):

Το Πανεπιστήμιο Κρήτης, στα σαράντα χρόνια λειτουργίας του, έχει αναδειχθεί σε ένα έντονα εξωστρεφές και καινοτόμο εκπαιδευτικό και ερευνητικό

ίδρυμα με σημαντικές επιστημονικές διακρίσεις και ξεχωριστή παρουσία στις διεθνείς κατατάξεις. Σήμερα, το Πανεπιστήμιο Κρήτης διαθέτει πέντε σχολές (Φιλοσοφική, Κοινωνικών Επιστημών, Επιστημών Αγωγής, Θετικών και Τεχνολογικών Επιστημών, Επιστημών Υγείας) με δεκαέξι τμήματα που καλύπτουν έναν ευρύτατο επιστημονικό χώρο.

Αρκούν λίγες γραμμές απλού κώδικα για να απελευθερωθεί απίστευτη δύναμη. Η πρόσβαση της Python σε πολυάριθμες μεγάλες βιβλιοθήκες επιτρέπει πολλές φορές στον χρήστη να ψάξει μόνο το κατάλληλο module το οποίο στη συνέχεια θα εκτελέσει τις απαιτούμενες ενέργειες χωρίς τη δική του παρέμβαση. Ακολουθούν μερικά παραδείγματα πραγματικών εφαρμογών γραμμένων στην Python:

- Το **YouTube** είναι ιστότοπος ο οποίος επιτρέπει την κοινοποίηση, αποθήκευση, αναζήτηση και αναπαραγωγή ψηφιακών βίντεο και ψηφιακών ταινιών για κάθε άτομο,
- Το **Instagram** είναι εφαρμογή κοινωνικής δικτύωσης που δίνει τη δυνατότητα επεξεργασίας και κοινοποίησης φωτογραφιών και βίντεο στο διαδίκτυο. Οι χρήστες μπορούν να μοιράζονται φωτογραφίες και βίντεο με τους ακολούθους τους (followers) ή με επιλεγμένη ομάδα φίλων, να σχολιάζουν και να δηλώνουν ότι μια δημοσίευση τους αρέσει,
- Το **Pinterest** είναι πλατφόρμα κοινωνικής δικτύωσης και διαδικτυακό εργαλείο δημιουργίας οπτικών σελιδοδεικτών. Οι χρήστες του Pinterest μπορούν να αναρτούν δικές τους φωτογραφίες και βίντεο ή να αποθηκεύουν φωτογραφίες που βρίσκουν στο διαδίκτυο,
- Το **Disqus** είναι μια δικτυακή πλατφόρμα κοινότητας που χρησιμοποιείται από εκατοντάδες χιλιάδες ιστοτόπους. Παρέχει στους ιστοτόπους ένα πλούσιο σε χαρακτηριστικά σύστημα σχολίων με ενσωμάτωση κοινωνικών δικτύων, προηγμένες επιλογές διαχείρισης και συντονισμού, και άλλες εκτεταμένες λειτουργίες κοινότητας,
- Το **Spotify** είναι μια υπηρεσία αναπαραγωγής μουσικής με παγκόσμια διαθεσιμότητα. Επιτρέπει την πλοήγηση και την αναζήτηση μουσικής ανά καλλιτέχνη, άλμπουμ, είδος, λίστα αναπαραγωγής ή δισκογραφικής εταιρείας,
- Το **Dropbox** είναι μια εφαρμογή αποθηκευτικού νέφους, δηλαδή μια υπηρεσία που επιτρέπει την αποθήκευση, τον συγχρονισμό και την κοινή χρήση αρχείων μεταξύ των διαφορετικών συσκευών στις οποίες ο χρήστης έχει επιλέξει να συνδέσει στον λογαριασμό του,
- Η **Uber** είναι μία αμερικανική επιχείρηση παροχής υπηρεσιών με έδρα το Σαν Φρανσίσκο. Προσφέρει online υπηρεσίες μεταφοράς προσώπων σε πολλές πόλεις παγκοσμίως. Η UberTaxi παρέχει ταξί. Η διαμεσολάβηση επιβάτη–οδηγού πραγματοποιείται μέσω μιας εφαρμογής για κινητά ή μιας ιστοσελίδας. Η επιχείρηση χρεώνει προμήθεια έως 20%,
- Το **Reddit** είναι αμερικανική ιστοσελίδα συγκέντρωσης ειδήσεων, περιεχομένου και συζήτησης. Εγγεγραμμένα μέλη ανεβάζουν περιεχόμενο σε αυτήν, όπως εικόνες, συνδέσμους και κείμενα, τα οποία συζητούνται με άλλους χρήστες.

1.2 ΓΙΑΤΙ PYTHON;

Απλή. Η Python είναι απλή και μαθαίνεται εύκολα. Η μελέτη ενός script (προγράμματος) σε Python είναι σαν το διάβασμα ενός βιβλίου αλγορίθμων. Η ομοιότητα της Python με ψευδοκώδικα είναι ένα από τα πιο σημαντικά χαρακτηριστικά της. Μας επιτρέπει να δώσουμε έμφαση στη λύση του προβλήματος και όχι στη σύνταξη της γλώσσας προγραμματισμού. Αυτό το στοιχείο την κάνει ιδανική για την εφαρμογή της στην εκπαίδευση και όχι μόνο. Είναι εύκολη στην εκμάθηση, έχει λιτή σύνταξη και ο χρήστης μπορεί εύκολα να κατανοήσει τη λογική της.

Γλώσσα υψηλού επιπέδου. Όταν προγραμματίζουμε με την Python, δεν χρειάζεται να ανησυχούμε για τη διαχείριση της μνήμης, καθώς όλα γίνονται αυτόματα.

Ελεύθερη και Ανοικτού Κώδικα. Η Python είναι Ελεύθερο Λογισμικό και Λογισμικό Ανοικτού Κώδικα (ΕΛΛΑΚ). Με απλά λόγια, μπορούμε να διανείμουμε αντίγραφα αυτού του λογισμικού, να διαβάζουμε τον πηγαίο κώδικα, να κάνουμε αλλαγές σε αυτόν και να χρησιμοποιούμε κομμάτια του σε νέα ελεύθερα προγράμματα. Το ΕΛΛΑΚ βασίζεται στην ιδέα μιας κοινότητας που μοιράζεται τη γνώση. Αυτός είναι ένας από τους λόγους για τους οποίους η Python είναι τόσο καλή: δημιουργήθηκε και βελτιώνεται συνεχώς από μια κοινότητα που στοχεύει αποκλειστικά στην αναβάθμιση του κώδικα.

Φορητή. Λόγω του ανοικτού της κώδικα, η Python έχει υλοποιηθεί σε πολλές πλατφόρμες. Όλα τα Python προγράμματά σας μπορούν να δουλέψουν σε οποιαδήποτε από αυτές τις πλατφόρμες χωρίς να χρειάζονται καθόλου αλλαγές, αν είστε αρκετά προσεκτικοί ώστε να αποφύγετε να χρησιμοποιήσετε χαρακτηριστικά που εξαρτώνται από κάθε σύστημα. Μπορείτε να χρησιμοποιήσετε την Python στο Linux, στα Windows, στο FreeBSD, σε Macintosh, στο Solaris, στο OS/2, στην Amiga, στο AROS, στο AS/400, στο BeOS, στο OS/390, στο z/OS, στο Palm OS, στο QNX, στο VMS, στο Psion, στο Acorn RISC OS, στο VxWorks, σε PlayStation, στο Sharp Zaurus, στα Windows CE, ακόμα και σε drone.

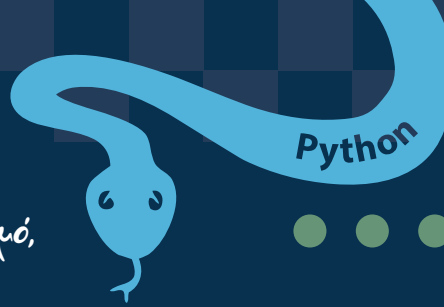
Διερμηνευόμενη. Ένα πρόγραμμα που γράφεται σε μια μεταγλωττιζόμενη γλώσσα όπως η C ή η C++ μετατρέπεται από αυτήν την πηγαία γλώσσα σε μια γλώσσα που μιλάει ο υπολογιστής (δυναμικός κώδικας, δηλαδή 0 και 1) χρησιμοποιώντας έναν μεταγλωττιστή. Για να λειτουργήσει το πρόγραμμα, ο linker το αντιγράφει στη μνήμη και αρχίζει να το τρέχει. Η Python, ωστόσο, δεν χρειάζεται μεταγλώττιση σε δυαδικό αρχείο, αλλά τρέχει το πρόγραμμα απευθείας από τον πηγαίο κώδικα. Εσωτερικά, η Python μετατρέπει τον πηγαίο κώδικα σε μια ενδιάμεση μορφή που ονομάζεται bytecode, στη συνέχεια το μεταφράζει στη γλώσσα του υπολογιστή και, τέλος, το τρέχει. Όλη αυτή η διαδικασία καθιστά τη χρήση της Python πολύ πιο εύκολη, αφού δεν χρειάζεται να φροντίσει ο χρήστης για τη μεταγλώττιση του προγράμματος, τη σύνδεση με τις κατάλληλες βιβλιοθήκες, κ.λπ. Επίσης, καθιστά τα προγράμματα της Python φορητά, εφόσον είναι δυνατή η αντιγραφή τους σε έναν άλλον υπολογιστή στον οποίο θα τρέξουν άμεσα.

Αντικειμενοστρεφής. Η Python υποστηρίζει τον διαδικασιοστρεφή και τον αντικειμενοστρεφή προγραμματισμό object-oriented. Στον διαδικασιοστρεφή προγραμματισμό, το πρόγραμμα δομείται επάνω σε διαδικασίες ή συναρτήσεις που προέκυψαν ως επαναχρησιμοποιήσιμα τμήματα άλλων προγραμμάτων. Στις αντικειμενοστρεφείς γλώσσες, τα προγράμματα δομούνται επάνω σε αντικείμενα τα οποία συνδυάζουν δεδομένα και λειτουργικότητα. Η Python έχει έναν ισχυρότατο αλλά και πολύ απλό τρόπο για αντικειμενοστρεφή προγραμματισμό, ειδικά όταν συγκρίνεται με μεγάλες γλώσσες όπως η C++ ή η Java.

Επεκτάσιμη. Αν είναι αναγκαίο ένα κρίσιμο κομμάτι του κώδικα να τρέχει πολύ γρήγορα ή αν ένα κομμάτι κάποιου αλγορίθμου πρέπει να μην είναι ανοικτό, υπάρχει η δυνατότητα προγραμματισμού εκείνου του τμήματος σε C ή C++ και της μετέπειτα ένταξής του στο υπό κατασκευή πρόγραμμα.

Ενσωματώσιμη. Μπορείτε να ενσωματώσετε την Python μέσα στα προγράμματα σε C ή C++, προκειμένου να τους δώσετε δυνατότητες scripting για τους χρήστες.

Αμέτρητες βιβλιοθήκες. Η πρότυπη βιβλιοθήκη της Python είναι κυριολεκτικώς τεράστια και μπορεί να βοηθήσει στη δημιουργία κανονικών εκφράσεων: δημιουργία τεκμηρίωσης, δοκιμές μονάδων, νημάτωση, βάσεις δεδομένων, περιηγητές ιστού, CGI, FTP, email, XML, XML-RPC, HTML, αρχεία WAV, κρυπτογράφηση, γραφικές διεπαφές χρήστη (GUI graphical user interfaces), Tk και άλλα, τα οποία εξαρτώνται από το υφιστάμενο σύστημα. Αξίζει να σημειωθεί ότι όλα αυτά είναι διαθέσιμα σε κάθε εγκατάσταση της Python. Εκτός από την πρότυπη, υπάρχουν διάφορες άλλες



Επενδύστε στο να μάθετε προγραμματισμό,
γιατί ο προγραμματισμός
σού μαθαίνει να σκέφτεσαι.

- Στιβ Τζομπς, ιδρυτής της Apple -

3

Ψηφιακό ημερολόγιο, games, κρυπτογραφία, τυχερά παίγνια

Στο κεφάλαιο αυτό θα υλοποιήσουμε τις εξής εφαρμογές: ψηφιακό ημερολόγιο, εύρεση μέτρησης διαφοράς ημερομηνιών με GUI, και χρήση του module Pygame για τη δημιουργία των παιχνιδιών Space Attack, Tetris, Flappy Bird. Τέλος, υλοποιούμε μία εφαρμογή κρυπτογράφησης και αποκρυπτογράφησης αρχείων σε GUI και μία εφαρμογή μοντελοποίησης χρήσης τυχερών παιγνίων.

3.1 ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ ΨΗΦΙΑΚΟΥ ΗΜΕΡΟΛΟΓΙΟΥ

GUI Calendar Using Python



Σε αυτήν την ενότητα θα υλοποιήσουμε μια εφαρμογή σε γραφικό περιβάλλον με θέμα την κατασκευή ενός ψηφιακού ημερολογίου. Θα χρησιμοποιήσουμε τις προεγκατεστημένες βιβλιοθήκες tkinter και tkcalendar. Ο κώδικας του προγράμματος εμφανίζεται στο «Πρόγραμμα 3.2».

Tkinter – GUI (Graphical User Interface) – Γραφική διεπαφή χρήστη

Το module Tkinter είναι μία από τις πολλές βιβλιοθήκες για γραφικές εφαρμογές στη γλώσσα προγραμματισμού Python. Η λειτουργία της βασίζεται στον προγραμματισμό γεγονότων (event based programming).

Event-based programming: προγραμματισμός με γεγονότα

- Η tkinter είναι η διεπαφή της Python 3 με τη γραφική βιβλιοθήκη tk (tcl/tk),
- η tkinter περιλαμβάνεται στις βασικές βιβλιοθήκες της Python και δεν χρειάζεται εγκατάσταση,
- εναλλακτικές γραφικές βιβλιοθήκες είναι οι WxPython PyGtk, PyQt και kivy,
- με την tkinter δημιουργούμε παράθυρα που περιέχουν γραφικά στοιχεία widget όπως labels, buttons, entry texts, menu, scroll bars κ.λπ.,
- τα widget δέχονται γεγονότα (event) από ενέργειες του χρήστη και ανταποκρίνονται με κλήση συναρτήσεων: event_handlers.

```
window=tkinter.Tk() window.mainloop()
```

Η συνάρτηση window.mainloop() ξεκινάει έναν βρόχο επεξεργασίας γεγονότων για το αντικείμενο window.

Αλγόριθμος λειτουργίας εφαρμογών γραφικού περιβάλλοντος

- tkinter.ttk themed tk (περισσότερα widget, χρήση στυλ)

```
while True:
    Σχεδίασε γραφικά αντικείμενα
    Περίμενε event (ενέργεια) από χρήστη
    Ανάλογα με το event:
    Κάλυψε συνάρτηση χειριστή (event_handler)
```

Module της Tkinter

- tkinter.ttk themed tk: περισσότερα widget, χρήση στυλ,
- tkinter.scrolledtext: δημιουργία παράθυρου με κύλιση περιεχομένου,
- tkinter.colorchooser: ορίζει την askcolor(initialcolor) για το παράθυρο επιλογής χρώματος,
- tkinter.filedialog: παράθυρο επιλογής αρχείου,
- tkinter.font: κλάση για τους μορφότυπους χαρακτήρων,
- tkinter.messagebox: κλάση για πλαίσια μηνυμάτων,
- tkinter.simpledialog: ορίζει την askinteger, askfloat and askstring για παράθυρα επιλογής τιμών,
- PIL για διαχείριση εικόνων τύπου jpg, png.

Πρόγραμμα 3.1 Κώδικας εφαρμογής ημερολογίου

```
# Εισαγωγή Βιβλιοθηκών from tkinter import * from tkcalendar import *

# Δημιουργός παραθύρου
root = Tk()

# Ο τίτλος του παράθυρου
root.title("GUI Calendar")

# Διαστάσεις.
root.geometry(«320x240»)

# Όχι αλλαγή μεγέθους
root.resizable(width = False, height = False)

# Δημιουργός ημερολογίου στο βασικό παράθυρο
calendar = Calendar(root, year = 2022, month = 10, day = 3) #

# Τοποθέτηση ημερολογίου μέσα στο παράθυρο root
calendar.pack(pady = 25)

# Ξεκινάει ένα βρόχο επεξεργασίας γεγονότων για το αντικείμενο root
root.mainloop()
```

Ανάλυση κώδικα

Εισάγουμε τις απαραίτητες βιβλιοθήκες `tkinter` και `tkcalendar`. Έπειτα, δημιουργούμε ένα κενό παράθυρο με την εντολή `root = Tk()`. Ορίζουμε τον τίτλο του παραθύρου με τη μέθοδο της κλάσης `Tk`, την `.title()`, και ορίζουμε τις διαστάσεις του παραθύρου με τη μέθοδο `.geometry(«320x240»)`. Αν δεν επιθυμούμε να έχει ο χρήστης τη δυνατότητα να τροποποιεί το μέγεθος του παραθύρου, εφαρμόζουμε τη μέθοδο `root.resizable(False, False)`. Με την επιλογή `root.resizable(width=A, height=B)` ορίζουμε αν θα αλλάζει το πλάτος ή το ύψος του παραθύρου (`A` και `B` είναι μεταβλητές `Boolean`). Δημιουργούμε ένα αντικείμενο ημερολογίου `calendar` της κλάσης `Calendar`, ορίζοντας το παράθυρο στο οποίο θα τοποθετηθεί, δηλαδή το `root`. Επίσης, ορίζουμε την ημερομηνία που θα εμφανιστεί στην οθόνη όταν ο χρήστης ξεκινήσει την εφαρμογή (μεταβλητές `year`, `month`, `day`). Με τη μέθοδο `.pack()` τοποθετούμε το ημερολόγιο στο παράθυρο `root` με άνω περιθώριο 25 pixel. Ξεκινάμε τον βρόχο επεξεργασίας γεγονότων για το αντικείμενο `root` απεικονίζοντας γραφικά το ημερολόγιό μας, όπως φαίνεται στο σχήμα 3.1.