Η Γλώσσα Προγραμματισμού Python

Αρχές Γλωσσών Προγραμματισμού και Μεταφραστών

Γ. Γαροφαλάκης, Σ. Σιούτας, Π. Χατζηδούκας

Διαδικαστικά μαθήματος (1)

- Εισαγωγή στη Γλώσσα Προγραμματισμού **Python**
  - Κατανόηση των βασικών εννοιών της Python: Μεταβλητές, τύποι δεδομένων και δομές ελέγχου.
  - Εισαγωγή στη συνάρτηση και τη δομή του κώδικα: Σύνταξη, αναγνώριση σφαλμάτων και debugging.
- Προχωρημένα Θέματα στη Γλώσσα Προγραμματισμού **Python**
  - Χρήση προηγμένων βιβλιοθηκών, για τη διαχείριση δεδομένων.
  - Κατανόηση της έννοιας της κληρονομικότητας και των αντικειμενοστραφών προγραμματιστικών αρχών στην python.
- Εισαγωγή στην Επιστήμη των Δεδομένων με **Python** (φροντιστήριο)
  - Εισαγωγή στη NumPy
  - Εισαγωγή στη χρήση Pandas
  - Εισαγωγή στη Matplotlib

Διαδικαστικά μαθήματος (2)

- Αξιολόγηση **(Νέα διαδικασία)**:
  - **60%** Τελική γραπτή εξέταση
  - **20%** Project flex/bison (υποχρεωτικό), 1 – 3 ατόμων
  - **20%** Εργασία Python (υποχρεωτική), 1 – 3 ατόμων
- Προϋποθέσεις:
  - **Τουλάχιστον 3,5** και στα τρία (γραπτό, flex/bison, Python)
  - Φυσικά, μέσος όρος των τριών, τουλάχιστον 5.
  - Μπορεί να δοθεί οποιοδήποτε/α από τα τρία τον Ιούνιο και το/τα υπόλοιπο/α το Σεπτέμβριο (-> βαθμός Σεπτεμβρίου).
  - Οι βαθμοί Project flex/bison και Εργασίας Python **παραμένουν** και για το επόμενο ακαδημαϊκό έτος. **Όχι** ο βαθμός του Γραπτού.
  - Οι φοιτητές/ριες μεγαλύτερων από το 2ο έτος, ακολουθούν την παλιά διαδικασία (70% γραπτή, 30% flex/bison, τουλ. 5 και στα 2).
  - Οι τελευταίοι/ες μπορούν να ακολουθήσουν τη νέα διαδικασία, εφόσον παραδώσουν Εργασία Python (έμμεση δήλωση προτίμησης διαδικασίας…)

Η Γλώσσα Προγραμματισμού Python

Το 1989 ο Guido Van Rossum δημιούργησε τη γλώσσα Python.

Η ονομασία της δεν προέρχεται από το φίδι πύθωνα αλλά από το γεγονός ότι ο Guido λάτρευε το Monty Pythons Flying Circus.

- Το 1994 παρουσιάστηκε η έκδοση 1.0
- Το 2000 παρουσιάστηκε η έκδοση 2.0
- Το 2008 παρουσιάστηκε η έκδοση 3.0
- Σήμερα χρησιμοποιούμε την έκδοση 3.13.2

Χαρακτηριστικά της γλώσσας Python

- Η Python είναι μια γλώσσα με απλή και καθαρή σύνταξη, εύκολη στην ανάγνωση και τη συγγραφή, ιδανική για αρχάριους και προχωρημένους.
- Ο κώδικάς της είναι ανοικτού λογισμικού και διαθέτει μια μεγάλη, ενεργή παγκόσμια κοινότητα υποστήριξης.
- Υποστηρίζει πολλαπλά προγραμματιστικά πρότυπα, όπως:
  - Διαδικαστικός προγραμματισμός
  - Αντικειμενοστρεφής προγραμματισμός
- Είναι cross-platform: λειτουργεί σε Windows, macOS και Linux.

Εφαρμογές της Python και διαθέσιμες βιβλιοθήκες

- Διαθέτει χιλιάδες βιβλιοθήκες για διάφορους τομείς, όπως:
  - Ανάλυση δεδομένων (π.χ. pandas, numpy)
  - Μηχανική μάθηση και τεχνητή νοημοσύνη (π.χ. scikit-learn, tensorflow)
  - Ανάπτυξη ιστοσελίδων (π.χ. Django, Flask)
  - Κατασκευή παιχνιδιών (π.χ. pygame)
- Χρησιμοποιείται ευρέως:
  - Στην έρευνα και την εκπαίδευση
  - Στην επιχειρηματική ανάλυση
  - Στην αυτοματοποίηση διαδικασιών (scripts, bots)
  - Στην ανάπτυξη λογισμικού και εφαρμογών

Χρήσιμες Ιστοσελίδες

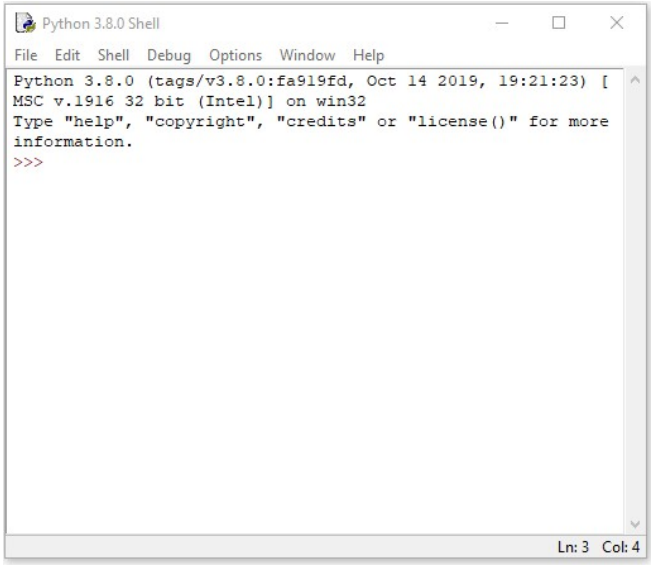
- Επίσημη ιστοσελίδα της Python
- Κεντρική βιβλιογραφία της Python
- Πληροφορίες για τη βασική βιβλιοθήκη της γλώσσας
- Λίστα με βιβλιοθήκες της Python

Εγκατάσταση

Στο python.org επιλέγουμε την έκδοση 3.13.2 της Python

Το πακέτο που θα γίνει download περιλαμβάνει:

Το περιβάλλον ανάπτυξης **IDLE**

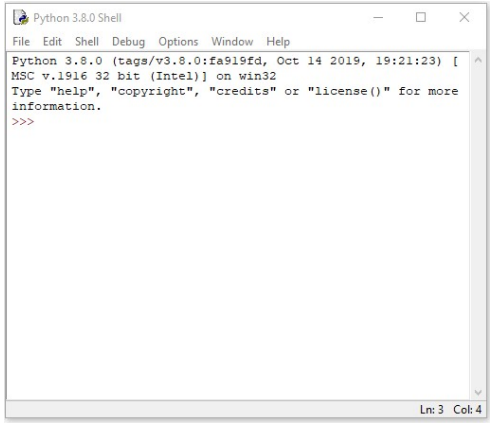


Σημαντικό: επιλέγουμε το κουτάκι "Add Python to PATH" κατά την εγκατάσταση.

Εργαλεία & Περιβάλλοντα για Python (1/2)

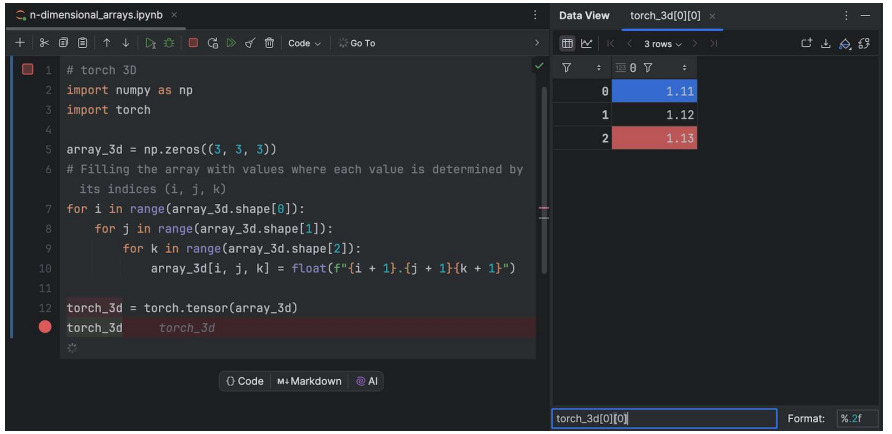
Python Idle

Επίσημος compiler από το python.org



PyCharm

IDE για έργα και debugging

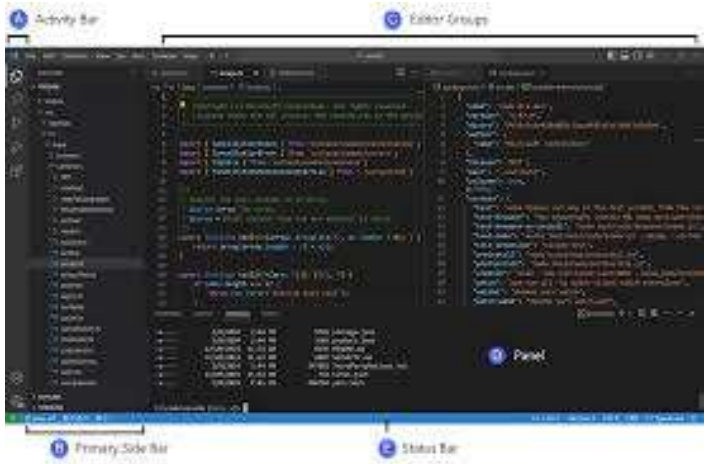


Εργαλεία & Περιβάλλοντα για Python (2/2)

Visual Studio Code

VS Code

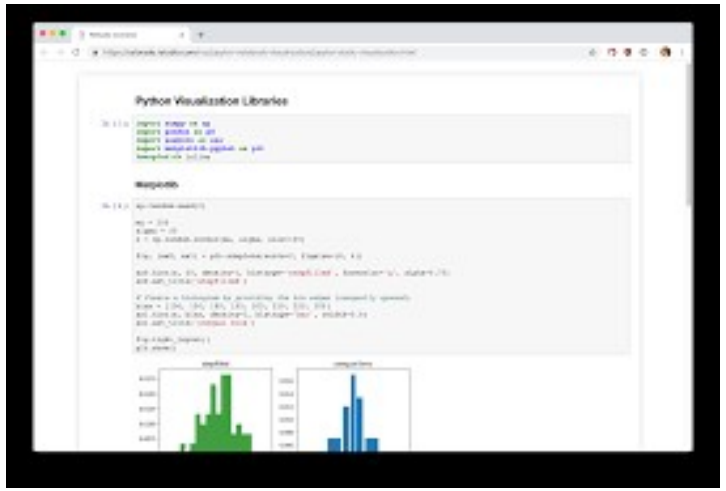
Ελαφρύ, γρήγορο, με υποστήριξη Python μέσω επέκτασης



jupyter

Jupyter Notebook

Ιδανικό για ανάλυση δεδομένων, παρουσιάσεις και εκπαίδευση



pip

Διαχειριστής πακέτων για εγκατάσταση βιβλιοθηκών

Χρήση της Python σε άλλες Επιστήμες

- Η Python χρησιμοποιείται ευρέως και εκτός του πεδίου των επιστημών πληροφορικής:
- **Βιολογία**: Ανάλυση γονιδιώματος, bioinformatics, προσομοιώσεις
- **Χημεία**: Μοριακή μοντελοποίηση, φασματοσκοπία
- **Γεωεπιστήμες**: Ανάλυση χωρικών δεδομένων (GIS), μετεωρολογία
- **Οικονομικά / Κοινωνικές επιστήμες**: Στατιστική ανάλυση, οικονομετρία
- **Φυσική**: Προσομοίωση φαινομένων, αριθμητική επίλυση εξισώσεων

Γιατί Python για μηχανικούς & επιστήμονες;

- Εύκολη καμπύλη εκμάθησης για ερευνητές χωρίς background προγραμματισμού
- Πολλά διαθέσιμα παραδείγματα και notebooks
- Δυνατότητα αυτοματοποίησης πειραμάτων και αναλύσεων
- Υποστήριξη διεθνών επιστημονικών κοινοτήτων
- Ισχυρά εργαλεία για visualisation, στατιστικά και machine learning

Python & Data Science

- Η Python είναι η πιο δημοφιλής γλώσσα για Data Science λόγω:
  - Απλής σύνταξης
  - Πληθώρας βιβλιοθηκών
  - Μεγάλης κοινότητας
  - Ενσωμάτωσης με εργαλεία visualisation και machine learning

Βασικές Βιβλιοθήκες στην Python για Data Science

- NumPy: αριθμητικοί υπολογισμοί και πίνακες
- Pandas: ανάλυση και επεξεργασία δεδομένων (DataFrames)
- Matplotlib / Seaborn: οπτικοποίηση δεδομένων
- Scikit-learn: μηχανική μάθηση
- Statsmodels: στατιστική ανάλυση
- TensorFlow / PyTorch: deep learning

Ενδεικτικές Εφαρμογές της Python στη Data Science

- Καθαρισμός και μετασχηματισμός δεδομένων
- Δημιουργία γραφημάτων και dashboards
- Πρόβλεψη μέσω αλγορίθμων μηχανικής μάθησης
- Επεξεργασία φυσικής γλώσσας (NLP)
- Ανάλυση εικόνας και video

Anaconda – Ολοκληρωμένη Διανομή για Python & Data Science

- Περιλαμβάνει:
  - Τον Python interpreter
  - Το conda (διαχειριστής πακέτων & περιβαλλόντων)
  - Πλήθος έτοιμων βιβλιοθηκών (π.χ. numpy, pandas, matplotlib)
  - Εργαλεία όπως Jupyter Notebook, Spyder
- Πλεονεκτήματα:
  - Ευκολία στην εγκατάσταση όλων των εργαλείων μαζί
  - Ιδανικό για επιστημονικό υπολογισμό, στατιστική, machine learning
  - Κατάλληλο για εκπαιδευτικά εργαστήρια και αρχάριους

Εναλλακτικά & Συμπληρωματικά Εργαλεία

- **Miniconda**
  - Ελαφριά έκδοση του Anaconda – εγκαθιστάς μόνο τα πακέτα που χρειάζεσαι
- **JupyterLab**
  - Νέο, πιο ισχυρό interface για notebooks
  - Υποστηρίζει αρχεία, terminals, live code
- **Google Colab**
  - Online περιβάλλον Jupyter notebook
  - Υποστήριξη GPU/TPU χωρίς εγκατάσταση
  - Ιδανικό για συνεργασία και machine learning
- **Spyder IDE**
  - Περιβάλλον τύπου MATLAB, κατάλληλο για επιστημονικό υπολογισμό

Μεταβλητές (1/2)

Το όνομα μιας μεταβλητής:

- μπορεί να περιλαμβάνει λατινικούς χαρακτήρες ή αριθμούς (0…9) καθώς και το _
- αλλά **δεν** μπορεί να αρχίζει με αριθμητικό ψηφίο.

Παραδείγματα:

```
y = 7
firstname = “Νίκος“
pi        = 3.14
```

Μεταβλητές (2/2)

Τύποι αριθμητικών μεταβλητών:

- **<int>** ακέραιος (integer)
- **<float>** δεκαδικός (floating point)
- **<str>** συμβολοσειρά (string)

Με την εντολή type(x) εμφανίζεται ο τύπος μιας μεταβλητής

Εκχώρηση τιμής σε μεταβλητή (1/2)

Παραδείγματα:

```
b = 5
type(a)

z = z**3 + 10
x = “Κώστας ” - ”Νίκος”
```

Σύμβολα πράξεων:

+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
/	Διαίρεση
//	Ακέραια διαίρεση (πηλίκο)
%	modulo (υπόλοιπο διαίρεσης)
**	Ύψωση σε δύναμη

Εκχώρηση τιμής σε μεταβλητή (2/2)

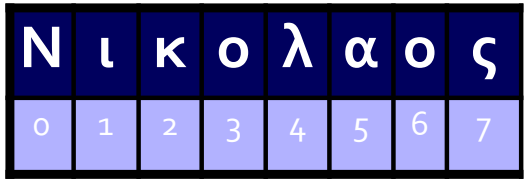
Συντομογραφίες πράξεων:

- **x += y** Πρόσθεση του y στην τιμή του x (x = x + y)
- **x -= y** Αφαίρεση του y από την τιμή του x (x = x – y)
- **x *= y** Πολλαπλασιασμός του y με την τιμή του x (x = x * y)
- **x /= y** Διαίρεση του y με την τιμή του x (x = x / y)
- **x //= y** Ακέραια διαίρεση του y με x (x = x // y)
- **x %= y.** Υπόλοιπο της διαίρεσης του y με x (x = x % y)
- **x **= y** Ύψωση του y στο x (x = x **y)

Δημιουργία συμβολοσειράς (string)

Μια συμβολοσειρά είναι μια ακολουθία χαρακτήρων που περικλείεται από μονά ή διπλά εισαγωγικά
Κάθε χαρακτήρας έχει τη θέση του στο αλφαριθμητικό οι οποίες ξεκινούν από τη θέση 0

Παραδείγματα:



name="Νικόλαος" (8 χαρακτήρες)
city= 'Πάτρα'

name[0] : N

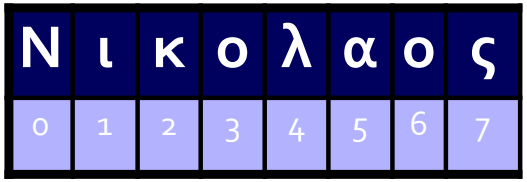
Οι χαρακτήρες αποθηκεύονται ως ακέραιοι μήκους από 1 έως 4 bytes
Η συνάρτηση ord(character) επιστρέφει τον κωδικό του χαρακτήρα
Η συνάρτηση chr(integer) επιστρέφει το χαρακτήρα που αντιστοιχεί ο κωδικός.

Τεμαχισμός συμβολοσειράς (slice)

Ένα τμήμα μιας συμβολοσειράς αποτελείται από το χαρακτήρα της αρχής μέχρι το χαρακτήρα τέλους, χωρίς να περιλαμβάνεται ο τελευταίος.

Παράδειγμα:

s=“Νικόλαος” (8 χαρακτήρες)



s[2:6]
‘κολα’

Πράξεις με συμβολοσειρές

Οι συμβολοσειρές αποτελούν αμετάβλητες ακολουθίες χαρακτήρων και **δεν επιτρέπεται η αλλαγή τους.**

Επιτρέπεται όμως η **πρόσθεση** συμβολοσειρών όπως και ο **πολλαπλασιασμός** συμβολοσειράς με ακέραιο.

τελεστής	**αποτέλεσμα**
<seq> + <seq>	συνένωση
<seq> * <int>	επανάληψη
<seq>[]	δείκτης
len(<seq>)	μήκος ακολουθίας
<seq>[:]	τεμαχισμός
for <var> in <seq>:	επανάληψη
<expr> in <seq>	συμμετοχή (Boolean)

Επιπλέον η συνάρτηση len(s) μας δίνει το πλήθος χαρακτήρων της συμβολοσειράς

Μέθοδοι συμβολοσειρών

Υπάρχουν πολλές μέθοδοι που μπορούν να εφαρμοστούν σε μια συμβολοσειρά string για αναζήτηση, μετατροπή, εξαγωγή λέξεων, έλεγχο αν το κείμενο αποτελείται από αριθμούς ή γράμματα κ.α.

Μερικές από αυτές είναι οι παρακάτω:

islower() - αληθές αν το str έχει μόνο πεζά
isdigit () - αληθές αν το str έχει μόνο αριθμούς
strip([chars]) – αφαιρεί τους χαρακτήρες από την αρχή και το τέλος του str
upper() - μετατρέπει τα κεφαλαία σε μικρά
lower() – μετατρέπει τα πεζά σε κεφαλαία
count() – μέτρηση του αριθμού εμφάνισης ενός χαρακτήρα σε ένα str
isalpha() – αληθές αν περιέχονται μόνο χαρακτήρες

Εντολές output

print()

print(ορίσματα, sep =‘δ’, end = ‘ε’)

δ : διαχωριστής

ε : τερματικός χαρακτήρας

Εντολές output

```
>>> x = 124.456

>>> print("η τιμή του χ είναι:", x)

η τιμή του χ είναι: 124.456

>>> print("η τιμή του χ είναι %f" % (x))

η τιμή του χ είναι 124.456000

>>> print("η τιμή του χ είναι %1.2f" % (x))

η τιμή του χ είναι 124.46
```

Εντολές input

input()

x=input(“Δώσε μια τιμή:”)

Προσοχή: Η input() επιστρέφει συμβολοσειρά

Τι είναι λάθος στο παρακάτω παράδειγμα:

x = input(“Δωσε 1ο αριθμό”)

y = input(“Δωσε 2ο αριθμό”)

s=x+y

Δομή δεδομένων Λεξικού (dictionary)

Ορισμός λεξικού ακολουθία από ζευγάρια τύπου:
dict={κλειδί: τιμή, …} μέσα σε { }

katalogos = {“Κώστας”:123456,”Σπύρος”:122344}

Τα κλειδιά πρέπει να είναι **μοναδικά** - επιτρέπονται αμετάβλητοι τύποι δεδομένων όπως αριθμοί, string, tuples, αλλά όχι list, dictionaries

Μέθοδοι λεξικών:
keys() – επιστρέφει τα κλειδιά
values() – επιστρέφει τις τιμές των στοιχείων
items() – επιστρέφει μια λίστα από τα αντικείμενα

Δομή επανάληψης for

for μέλος in αντικείμενο :

ομάδα εντολών 1

else:

ομάδα εντολών 2

(η ομάδα εντολών 2 δεν εκτελείται αν υπάρχει break)

for/break/continue

```
for μέλος in αντικείμενο :
        ομάδα εντολών 1

        if συνθήκη :
                continue
        if συνθήκη :
                break

else:

        ομάδα εντολών 2
```

range: επανάληψη for με απαρίθμηση

for i **in** range(start, end, step) : ομάδα εντολών 1

Εμβέλεια μεταβλητών

- Εμβέλεια ορίζεται η περιοχή του προγράμματος στην οποία είναι γνωστή μια μεταβλητή
- Οι **τοπικές** μεταβλητές έχουν εμβέλεια μόνο στη συνάρτηση ορισμού τους και ορίζονται μέσα σε συναρτήσεις
- Οι μεταβλητές λέγονται καθολικές (**global**): ορίζονται εκτός συναρτήσεων και έχουν εμβέλεια σε ολόκληρο το πρόγραμμα
- Ορισμός τοπικής μεταβλητής ως **global**: οι αλλαγές της επηρεάζουν την συνώνυμη καθολική μεταβλητή

Βιβλιοθήκες / modules

- **import** module_name
- **from** module_name **import** *

Η βιβλιοθήκη module_name είναι ένα αρχείο python, με όνομα module_name.py το οποίο φορτώνεται ώστε τα στοιχεία που περιέχει να μπορούν να χρησιμοποιηθούν στο πρόγραμμά μας.

Επιπλέον μπορούμε να δημιουργήσουμε τα δικά μας modules και να τα καλέσουμε από το πρόγραμμά μας.

Python modules (ενδεικτικά)

random	δημιουργία τυχαίων αριθμών
math	μαθηματικές συναρτήσεις
os os.path	διεπαφή λειτουργικού συστήματος
urllib.request	ανάκτηση ιστοσελίδων
sqlite3	βάση δεδομένων SQLite
csv	διαχείριση αρχείων csv
socket	διαδικτυακή επικοινωνία διεργασιών
cgi	Common Gateway Interface
wsgiref	υλοποίηση διεπαφής WSGI

try / except / finally

Exception : η κατάσταση σφάλματος στην οποία εισέρχεται ένα πρόγραμμα.

Η δομή **try / except** μας επιτρέπει να αντιμετωπίσουμε τα σφάλματα έτσι ώστε να μην διακοπεί η λειτουργία του προγράμματος.

```
try:
εντολές
…
except <τύπος σφάλματος 1> :
εντολές
except <τύπος σφάλματος 2> :
εντολές
finally :
εντολές
```

Διαχείριση αρχείων

Στα αρχεία αποθηκεύονται δεδομένα που μπορούμε να διαβάσουμε. Επιπλέον μπορούμε να γράψουμε δεδομένα σε αρχείο που υπάρχει ήδη, αλλά και να το δημιουργήσουμε (αν δεν υπάρχει).

Υπάρχουν δυο είδη αρχείων: κειμένου ή αρχεία bytes

Αρχεία κειμένου

fobj = **open**(όνομα, κατάσταση, encoding= 'utf-8')
Κατάσταση = '**w**' (write), '**r**' (read), '**a**' (append)
fobj.**name** # the name of the file
fobj.**read**() # read content of file
fobj.**seek**(0) # set current position at start
fobj.**close**() # close file

Σύνδεση με λειτουργικό σύστημα

Οι βιβλιοθήκες **os**, **os.path**, επιτρέπουν τη διεπαφή με το λειτουργικό σύστημα, τη δημιουργία και τη διαγραφή φακέλων, καθώς και την προβολή του περιεχομένου τους.

Επίσης η συνάρτηση **os.system()** μας επιτρέπει να εκτελούμε εντολές της γραμμής εντολών.

Η βιβλιοθήκη os

os.**getcwd**() # current folder
os.**chdir**(d) # change folder
os.**listdir**(d) # content of folder
os.**rename**("t1.txt", "t2.txt") # rename file
os.**remove**(file_name) # delete file
os.**mkdir**("newdir") # create folder
os.**rmdir**(dirname) # delete folder
os.**system**('mkdir newdir') # command line
os.**walk**(d) # walk the generated file tree
os.**sep** # separate character for folders (/)

Η βιβλιοθήκη os.path

os.path.**isdir**(filename) # ελέγχει αν είναι φάκελος
os.path.**split**(filename) # χωρίζει φάκελο-αρχείο
os.path.**splitext**(filename) # χωρίζει επέκταση
os.path.**dirname**(filename) # επιστρέφει φάκελο
os.path.**basename**(filename) # επιστρέφει αρχείο
os.path.**join**(dir, f) # ενώνει φάκελο+αρχείο

Regular expressions (1)

Η βιβλιοθήκη re επιτρέπει τη χρήση της γλώσσας αναγνώρισης προτύπων σε κείμενο Regular Expressions (regex).

Μέσω της re μπορεί να γίνει αναζήτηση προτύπων σε κείμενα, καθώς και εκτέλεση πράξεων όπως ανάκτηση, μετατροπή κ.α. βάσει αυτών των προτύπων.

Regular expressions (2)

Η βιβλιοθήκη re επιτρέπει τη χρήση Regular Expressions (regex), μια ισχυρή γλώσσα για τον εντοπισμό και τη διαχείριση προτύπων κειμένου. Μέσω της re μπορούμε να:

- Αναζητούμε πρότυπα σε κείμενα: Εντοπισμός συγκεκριμένων χαρακτήρων, λέξεων ή μοτίβων.
- Αντικαθιστούμε/Μετατρέπουμε κείμενο: Χρήση regex για αναγνώριση και αντικατάσταση κειμένων με άλλες τιμές ή μοτίβα.
- Διαχωρίζουμε κείμενο: Χρήση regex για διαχωρισμό κειμένου σε βάση προτύπων (π.χ. χώρους, κόμματα).
- Εξαγωγή πληροφοριών: Εξαγωγή συγκεκριμένων κομματιών πληροφορίας από κείμενα, όπως email ή αριθμούς τηλεφώνου.

Regular expressions - Συναρτήσεις

re.**compile**(pattern): αντικείμενο pattern* (μετάφραση προτύπου)

re.match(), re.search(): Αναζητούν πρότυπα στην αρχή του κειμένου ή οποιοδήποτε μέρος του.

re.findall(), re.finditer(): Επιστρέφουν όλα τα ευρήματα ως λίστα ή iterator.

re.sub(): Αντικαθιστά τα ευρήματα με νέο κείμενο.

re.split(): Χρησιμοποιεί πρότυπα για να διαχωρίσει το κείμενο.

Regular expressions – Κλάσεις χαρακτήρων

\w Οποιοσδήποτε αλφαριθμητικός χαρακτήρας [a-zA-Z0-9_]
\W Οτιδήποτε εκτός αλφαριθμητικών χαρακτήρων [^a-zA-Z0-9_]
\d Αριθμός, [0-9]
\D Οτιδήποτε εκτός αριθμού, [^0-9]
\s Κενό [\n\f\t\v\r]
\S Οτιδήποτε εκτός του κενού [^[\n\f\t\v\r]

\b Όριο μεταξύ αλφαριθμητικού και μη-αλφαριθμητικού (αρχή ή τέλος λέξης)

Ειδικοί χαρακτήρες: . ^ $ * + ? { } [ ] \ | ( )

Ανάκτηση δεδομένων από το διαδίκτυο

Βιβλιοθήκη urllib.request

```
req_object = urllib.request.Request(διαδ. πόρος)
mypage = urllib.request.urlopen(req_object)
html = mypage.read().decode()
```