



ΠΕΡΙΕΧΟΜΕΝΑ:

1. Συναρτήσεις
 1. Ορίσματα με Λέξεις-Κλειδιά
 2. Προκαθορισμένες Τιμές Παραμέτρων
 3. Αυθαίρετο Πλήθος Ορισμάτων
2. Algorithm: Selection Sort
3. Game Project: Yahtzee
4. Data Project: CRUD – Delete

Θάνος Θ.

Χρυσός Χορηγός Μαθήματος

Νίκος Θ.

Ασημένιος Χορηγός Μαθήματος

ΜΑΘΗΜΑ 13: Περισσότερα για τις Συναρτήσεις

1. 1. Ορίσματα με λέξεις - κλειδιά

- Τα ορίσματα που έχουμε μελετήσει καλούνται και θεσιακά ορίσματα (positional arguments) :
 - Διότι γίνεται 1-1 αντιστοίχιση παραμέτρου – ορίσματος από αριστερά προς τα δεξιά.
- Υπάρχουν ωστόσο και τα **ορίσματα με λέξεις – κλειδιά (keyword arguments)**:
 - Θέτουμε τα ορίσματα ως ζεύγη κλειδιού – τιμής, όπου κλειδί είναι το όνομα της παραμέτρου και τιμή είναι το όρισμα που περνάμε.
 - Η σύνταξη είναι **κλειδί = τιμή**.

Παράδειγμα 1: keyword.arguments.py

```
def print_full_name(first_name, surname, fathers_name):  
    print(f"{first_name} {fathers_name[0]}. {surname}")  
  
print_full_name("Jack", "Bauer", "Philip")  
print_full_name("Jack", fathers_name="Philip", surname="Bauer")
```

Οι δύο κλήσεις είναι ισοδύναμες.

Παρατήρηση:

- Τα θεσιακά ορίσματα πρέπει να προηγούνται των ορισμάτων με λέξεις κλειδιά.
- Σε αντίθετη περίπτωση, το πρόγραμμα θα βγάλει λάθος:

```
# positional.after.keyword.args.py  
def print_full_name(first_name, surname, fathers_name):  
    print(f"{first_name} {fathers_name[0]}. {surname}")  
  
print_full_name(surname="Bauer", first_name="Jack", "Philip")
```

Άσκηση 1:

Ορίστε τη συνάρτηση που υπολογίζει τη συνάρτηση: $f(x, y, z) = 2x^3 + 3x^2y + y^2z + 1$

Υπολογίστε με τέσσερις τρόπους την ποσότητα $f(5, 2, 3)$ χρησιμοποιώντας θεσιακά ορίσματα και ορίσματα με λέξεις – κλειδιά, ή συνδυασμό αυτών.

ΜΑΘΗΜΑ 13: Περισσότερα για τις Συναρτήσεις 1.2. Προκαθορισμένες τιμές παραμέτρων python 3

- Μας δίνεται η δυνατότητα να θέσουμε **προκαθορισμένες τιμές σε παραμέτρους (default value)**:
 - Αν δεν έχει οριστεί τιμή στην παράμετρο, κατά την κλήση της συνάρτησης, θα αποδίδεται αυτόματα στην παράμετρο, η προκαθορισμένη τιμή.
 - Το συντακτικό είναι: **parameter=default_value**, στον ορισμό της παραμέτρου της συνάρτησης.

Παράδειγμα 2: default.value.py

```
def print_name(first_name, surname, details=False):
    if details:
        print("First Name: " + first_name)
        print("Surname: " + surname)
    else:
        print(f"{first_name} {surname}")

print_name("Charles", "Kane")
print_name("Charles", "Kane", True)
```

Παρατηρήσεις:

- Οι παράμετροι με προκαθορισμένες τιμές, πρέπει να έπονται των παραμέτρων χωρίς προκαθορισμένη τιμή.

```
# default.values.order.py
def func(par1, par2, par3 = 0, par4 = 0):
    print(f"par1={par1} par2={par2} par3={par3} par4={par4}")

func(1,2,3,4)
func(1,2,3)
func(1,2)
```

- Ενώ συνηθισμένη χρήση σε συναρτήσεις είναι να γίνονται κλήσεις που να ορίζεται τιμή σε παραμέτρους με προκαθορισμένη τιμή, μέσω ορίσματος με λέξη κλειδί

```
# default.values.keyword.arguments.py
def func(par1, par2 = 0, par3 = 0, par4 = 0):
    print(f"par1={par1} par2={par2} par3={par3} par4={par4}")

func(1, par3="5")
```

Άσκηση 2:

Επεκτείνετε τη συνάρτηση print_name του παραδείγματος 2 με:

- Το όρισμα second_name με δεύτερο όνομα του ατόμου. Η εκτύπωση του πλήρους ονόματος να γίνει π.χ. ως "Charles-Bob"
- Το όρισμα second_surname με δεύτερο επώνυμο του ατόμου. Η εκτύπωση του πλήρους επωνύμου να γίνει με ένα ενδιάμεσο κενό μεταξύ των δύο επωνύμων.

ΜΑΘΗΜΑ 13: Περισσότερα για τις Συναρτήσεις

1.3. Αυθαίρετο Πλήθος Ορισμάτων

- Μας δίνεται η δυνατότητα να θέσουμε όσα ορίσματα θέλουμε σε μία συνάρτηση (arbitrary number of arguments – αυθαίρετος αριθμός ορισμάτων)
 - Θέτουμε `*args` ως παράμετρο
 - Σε αυτήν “μαζεύονται” όλα τα ορίσματα που διοχετεύουμε.

Παράδειγμα 3: args.py

```
def my_sum(*numbers):  
    print(numbers)  
    s = 0  
    for number in numbers:  
        print(number)  
        s += int(number)  
    return s  
  
print(f"sum={my_sum(1,2,3,4,5)}")
```

- Μας δίνεται η δυνατότητα να διοχετεύσουμε ζεύγη κλειδιού – τιμής (keyword arguments) ως ορίσματα με λέξεις-κλειδιά.
 - Θέτουμε `**kwargs` ως παράμετρο
 - Σε αυτήν “μαζεύονται” όλα τα ορίσματα που διοχετεύουμε στη μορφή ενός λεξικού.

Παράδειγμα 4: kwargs.py

```
def create_person(**kwargs):  
    print(kwargs)  
    return kwargs  
create_person(name="Jim", occupation="teacher")  
create_person(name="Bruce", ability="fly")
```

Σημείωση: Η Python ορίζει αυστηρή σειρά: Πρώτα τυχόν θεσιακά ορίσματα, έπειτα το `*args` και έπειτα το `**kwargs` σε περίπτωση που θέλουμε και τα τρία σε μία συνάρτηση.

Άσκηση 3:

Κατασκευάστε μία συνάρτηση με όνομα `float_average` η οποία δέχεται αυθαίρετο πλήθος πραγματικών αριθμών και υπολογίζει και επιστρέφει το μέσο όρο τους.

Άσκηση 4:

Ορίστε μία συνάρτηση η οποία δέχεται αυθαίρετο πλήθος ορισμάτων από:

- βαθμούς (ακέραιοι αριθμοί)
- Ζεύγη κλειδιού-τιμής που απεικονίζουν Επώνυμο – Μάθημα
- Να υπολογίζει το μέσο όρο όλων των βαθμών και να το τυπώνει
- Να υπολογίζει και να τυπώνει ανά μάθημα ποιοι μαθητές έχουν συσχετιστεί με το μάθημα αυτό.

ΜΑΘΗΜΑ 13: .. Συναρτήσεις Algorithm Project: Selection Sort (Ταξινόμηση με Επιλογή)

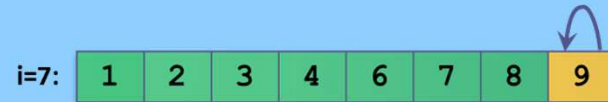
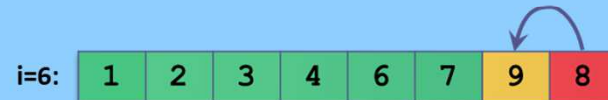
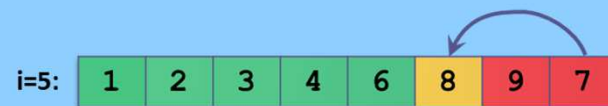
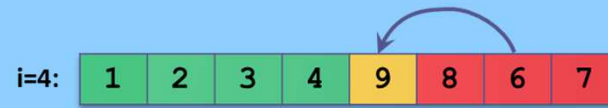
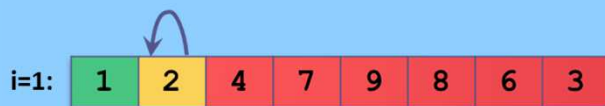
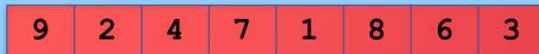
Συνεχίζουμε με το πρόβλημα της ταξινόμησης, με έναν ακόμη αλγόριθμο που το επιλύει.

Το **σκεπτικό της ταξινόμησης με επιλογή (selection sort)** είναι:

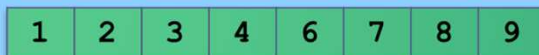
- Έχοντας ταξινομήσει ήδη τις θέσεις $0..i-1$
- Επιλέγουμε το ελάχιστο από τα στοιχεία στις θέσεις $i \dots N-1$ και το θέτουμε στη θέση i .

Παράδειγμα Εκτέλεσης:

Αρχικός Πίνακας:



Τελικός Πίνακας:



Άσκηση 5:

Υλοποιήστε τον αλγόριθμο Selection Sort (σε μία συνάρτηση)

Άσκηση 6:

Υλοποιήστε τον αλγόριθμο Selection Sort ώστε να αποφασίζεται μέσω παραμέτρου αν θα κάνει ταξινόμηση σε αύξουσα ή σε φθίνουσα σειρά. Παραδείγματα κλήσης που θέλουμε να δουλεύουν σωστά:

```
selection_sort([3, 5, 2])  
selection_sort([3, 5, 2], reverse = True)
```

Άσκηση 7.1: Ζαριές

Θα κατασκευάσουμε μια απλοποιημένη έκδοση του Yahtzee (βλ. βίντεο για επεξήγηση του παιχνιδιού), όπου οι δύο παίκτες προσπαθούν να “μαζέψουν” όσο μεγαλύτερα αθροίσματα από ίδια αποτελέσματα ζαριών (δηλαδή μόνο τις επιλογές ones, twos, ..., sixes)

- Ξεκινήστε την κατασκευή, ορίζοντας μία συνάρτηση με όνομα `roll_dice` η οποία παίρνει ως όρισμα έναν ακέραιο (1-5) και επιστρέφει αποτελέσματα ίσου πλήθους τυχαίων ρίψεων ζαριών, οργανωμένων σε μία ταξινομημένη λίστα.

Άσκηση 7.2: Γύρος Παίκτη

Κατασκευάστε μία συνάρτηση (`player_turn`) η οποία

- Δεν παίρνει ορίσματα
- Ο παίκτης παίζει τρεις φορές.
- Την πρώτη φορά ρίχνει 5 ζάρια. Η ζαριά εκτυπώνεται στον παίκτη και αυτός επιλέγει ποια ζάρια θέλει να κρατήσει (από κανένα έως όλα)
- Τις επόμενες φορές, αν ο παίκτης δεν έχει κρατήσει όλα τα ζάρια, τότε ρίχνει τα υπόλοιπα και επιλέγει πάλι ποια θέλει να κρατήσει.
- Επιστρέφεται μία λίστα με τα αποτελέσματα των ζαριών.

Άσκηση 7.3: Κράτημα Παίκτη

Σε μία global λίστα 2 θέσεων, θα κρατάμε το σκορ των δύο παικτών (λεξικό με έξι κλειδιά: `ones`, `twos`, ..., `sixes` με αρχική τιμή -1 σε όλα)

Κατασκευάστε την επιλογή του παίκτη ως προς τη ζαριά που έχει φέρει (`player_picks`) με ορίσματα τον παίκτη (0 ή 1) και τη ζαριά

- Να τυπώνονται οι διαθέσιμες επιλογές για να κρατήσει τη ζαριά του (π.χ. `ones`, `threes`, `fives`)
- Ο παίκτης επιλέγει σε ποιο κλειδί θέλει να αποθηκεύσει τη ζαριά του (να γίνεται έλεγχος ότι έχει κάνει έγκυρη επιλογή)

Άσκηση 7.4: Παιχνίδι

Ορίστε στη `main` μια επαναληπτική διαδικασία 6 γύρων:

- Σε κάθε επανάληψη θα παίζουν ο κάθε ένας μία φορά.
- Στο τέλος θα τυπώνεται αναλυτικά η κάρτα κάθε παίκτη, το συνολικό του σκορ και θα αναδεικνύεται ο νικητής.

Άσκηση 8.1: Διαγραφή με βάση το id

Δημιουργήστε μία νέα συνάρτηση:

- **delete_pupil_by_id**(pupil_id): Παίρνει ένα pupil_id και διαγράφει το μαθητή με το συγκεκριμένο id.

Άσκηση 8.2: Μενού Διαγραφών

Υλοποιήστε τη διαγραφή μαθητή:

- Αρχικά να προτρέπει το χρήστη να επιλέξει αν θέλει να διαγράψει με βάση το επώνυμο ή με id.
- Αν ψάξει με επώνυμο και υπάρχουν παραπάνω του ενός μαθητές, τότε να τυπώνει τα πλήρη στοιχεία των μαθητών και να προτρέπει το χρήστη να επιλέξει το id του μαθητή που επιθυμεί.
- Σε κάθε περίπτωση αναζήτησης αν δεν υπάρχουν μαθητές που να ταιριάζουν με τα κριτήρια αναζήτησης, τότε να τυπώνεται μήνυμα λάθους και να επιστρέφεται ο έλεγχος στο αρχικό μενού.
- Έχοντας το id ένα μαθητή πλέον, να γίνεται η διαγραφή του μαθητή.

“If the implementation is hard to explain, it's a bad idea.”

Zen of Python #17