



[7η Εβδομάδα]

Γλώσσα 'Προγραμματισμού':

{

[Python]

< Εισηγητής: Νίκος Κούκος >

}



```
1 7.0.0 { [Προηγούμενες ασκήσεις]
2
3
4
5
6      <
7      • Ενδεικτικές λύσεις
8      • Άλλες λύσεις και σχολιασμός
9      • Πρακτική και εμπέδωση
10     • Ασκήσεις: 1, 2, 3, 4, 5
11     >
12 }
13
14
```


7.1.0 { [Προαιρετικά - ορίσματα (Optional arguments)]

Παρέχονται κατ' επιλογή (γι' αυτό
μπαίνουν πάντα στο τέλος)

<

Ορισμός:

```
def όνομα_συνάρτησης(πρώτο_όρισμα,  
δεύτερο_όρισμα, προαιρετικό_όρισμα=' '):
```

... ..

>

7.2.0 { [Επιστροφή λεξικού – (dictionary)]

Μια συνάρτηση μπορεί να επιστρέψει
πολλά είδη δομών δεδομένων

Παράδειγμα:

```
def build_person(first_name, last_name):  
    """Επίστρεψε το σύνολο πληροφοριών ενός ατόμου σε μορφή  
    λεξικού."""  
    person = {'first': first_name, 'last': last_name}  
    return person  
musician = build_person('jimi', 'hendrix')  
print(musician)
```

Έξοδος:

```
{'first': 'jimi', 'last': 'hendrix'}
```

7.3.0 { [Συναρτήσεις + βρόχος while]

- Οι συναρτήσεις “συνεργάζονται” παντού μέσα στον κώδικα της Python και τον κάνουν πιο επαρκή
- Πώς δίνουμε στον χρήστη δικαίωμα να “βγει” όταν μπορεί μόνο να εισάγει στοιχεία σε ένα πρόγραμμα;

}

7.4.0 { [Περνώντας ορίσματα σε συναρτήσεις (Λίστες)]

- Τα ορίσματα είναι οι είσοδοι της κάθε συνάρτησης
- Η εργασία με τις λίστες είναι πιο κομψή και πιο αποδοτική. Επίσης πιο ευκολοσυντήρητη
- Παράδειγμα

}

7.5.0 { [Τροποποίηση ορισμάτων σε συναρτήσεις (Λίστες)]

- Οι αλλαγές στη λίστα είναι μόνιμες
- Αποτελεσματικότητα ακόμα και με μεγάλο όγκο δεδομένων
- Παράδειγμα του πραγματικού κόσμου


```
1
2 7.6.0 { [Αντίγραφο ασφαλείας
3         ορίσματος (Λίστας)]
4
5
6
7
8
9
10
11
12
13
14 }
```

- Αποτροπή αλλοίωσης του αρχικού αντιγράφου
- Αντιγραφή της πρωτότυπης λίστας
- Τροποποίηση του προηγούμενου παραδείγματος