

Λίγα απλά μαθηματικά



Ανοίξτε το IDLE (Python GUI) και δοκιμάστε τα εξής:

```
>>> 2 + 2
```

Όπως παρατηρείτε, μπορούμε να χρησιμοποιήσουμε το Python Shell σαν αριθμομηχανή (κομπιουτεράκι).

Δοκιμάστε με δικούς σας αριθμούς τις τέσσερις αριθμητικές πράξεις: +, -, *, /.

Τα σύμβολα των πράξεων ονομάζονται **τελεστές (operators)**.



Πληκτρολογήστε τα παραδείγματα που ακολουθούν:

```
>>> 2 + 2 + 2 + 2
```

```
>>> 8*6
```

```
>>> 10-5+6
```

```
>>> 2 +                2
```

```
>>> 5 +
```

Τι εμφανίζεται ως αποτέλεσμα κάθε φορά; Εξηγήστε τι συνέβη στην τελευταία περίπτωση.

Όταν συνδυάζουμε σωστά αριθμούς και αριθμητικούς τελεστές, δημιουργούμε αριθμητικές **εκφράσεις (expressions)**.

Αποθηκεύοντας τιμές σε μεταβλητές



Όταν προγραμματίζουμε, θέλουμε συχνά να αποθηκεύουμε το αποτέλεσμα ενός υπολογισμού (**την τιμή μίας έκφρασης**) για να την χρησιμοποιήσουμε αργότερα.

Μπορούμε να αποθηκεύουμε τιμές σε **μεταβλητές**.

Φανταστείτε τις μεταβλητές ως κουτιά που έχουν κάποιο όνομα και που μπορούν να χωρέσουν μόνο μια τιμή κάθε χρονική στιγμή.

Στην πραγματικότητα όμως, οι μεταβλητές είναι **θέσεις μνήμης** και τις βρίσκουμε στην **κύρια μνήμη (RAM)**.



Ας δοκιμάσουμε να δώσουμε την τιμή 15 στη μεταβλητή spam:

```
>>> spam = 15
```

Παρατηρούμε ότι δεν εμφανίστηκε τίποτε ως αποτέλεσμα της εντολής αυτής.

Μπορείτε να εξηγήσετε γιατί;

Με την εντολή αυτή **εκχωρούμε (assign)** την τιμή 15 στη μεταβλητή spam.

Θα μπορούσε η μεταβλητή να είχε ονομασθεί διαφορετικά;



Ένας τρόπος να δούμε την **τιμή μιας μεταβλητής** (το περιεχόμενό της) είναι να πληκτρολογήσουμε το όνομά της. Δοκιμάστε τα εξής:

```
>>> spam = 3
```

```
>>> spam
```

Τι εμφανίζεται;

Τι περιμένετε να εμφανιστεί ως αποτέλεσμα των ακόλουθων εντολών; Εξηγήστε.

```
>>> spam = 3
```

```
>>> spam + 7
```



Δοκιμάστε τις ακόλουθες εντολές. Παρατηρήστε το αποτέλεσμα κάθε εντολής και εξηγήστε γιατί το τελικό αποτέλεσμα είναι 8 κι όχι 28.

```
>>> spam = 15
```

```
>>> spam + 5
```

```
>>> spam = 3
```

```
>>> spam + 5
```



Ποιο είναι το αποτέλεσμα που περιμένετε να δείτε μετά την εκτέλεση των ακόλουθων εντολών:

```
>>> spam = 15
```

```
>>> spam + spam
```

```
>>> spam - spam
```

Ποια πιστεύετε ότι ήταν η τιμή της μεταβλητής spam όταν εκτελέστηκε η τρίτη εντολή;



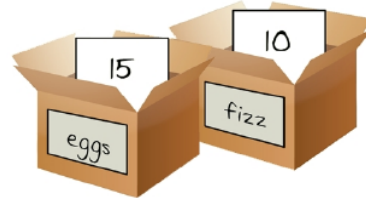
Χρησιμοποιώντας δύο μεταβλητές

Όταν προγραμματίζουμε χρησιμοποιούμε πολλές μεταβλητές. Πρέπει να φροντίσουμε να δώσουμε διαφορετικό όνομα στην καθεμιά.

Είναι σημαντικό κάθε μεταβλητή να έχει έναν **ρόλο** (όχι πάνω από έναν) και το όνομά της να δείχνει αυτό το ρόλο.

Δοκιμάστε τις εξής εντολές:

```
>>> eggs = 15
>>> fizz = 10
>>> spam = eggs + fizz
>>> spam
```



Πως εξηγείτε το τελικό αποτέλεσμα;



Ένα βήμα μπροστά

Με ποιες εντολές θα καταφέρουμε τα εξής:

1. Αρχικά θέλω να δώσω στην μεταβλητή *a* την τιμή 10.
2. Στη συνέχεια, θέλω να αλλάξω την τιμή της μεταβλητής *a* σε 30.
3. Τέλος, θέλω να εμφανίσω τόσο την παλιά όσο και την καινούργια τιμή της μεταβλητής *a*.

Δηλαδή: ποιος τρόπος υπάρχει να αποθηκεύουμε την τιμή μιας μεταβλητής, έτσι ώστε όταν η μεταβλητή για κάποιον λόγο αλλάξει τιμή, να έχουμε ακόμη διαθέσιμη την αρχική;

Αν δυσκολεύεστε δοκιμάστε τα εξής:

```
>>> a = 10
>>> b = a
>>> a = 30
>>> a
>>> b
```



Σκεφθείτε έναν τρόπο να αυξηθεί η τιμή της μεταβλητής a από 10 σε 30, χωρίς να χρησιμοποιηθεί άλλη μεταβλητή από την a και άλλος αριθμός από το 10.

Ξεκινήστε δηλ. με την εντολή

```
>>> a = 10
```

και σκεφθείτε τη συνέχεια... :-)

Αν δυσκολεύεστε, εξετάστε το αποτέλεσμα της εντολής

```
>>> a = a + 10
```



Αντιμετάθεση τιμών: πολύ σημαντικό!



Υποθέστε ότι δίνουμε αρχικά την τιμή 8 στη μεταβλητή a και την τιμή 30 στην μεταβλητή b . Επιχειρήστε να ανταλλάξετε τις τιμές των δύο μεταβλητών, χωρίς να χρησιμοποιήσετε πουθενά στις εντολές σας αριθμούς.

Ξεκινήστε με τις εντολές

```
>>> a = 8
```

```
>>> b = 30
```

και σκεφθείτε τη συνέχεια...



Αν δυσκολεύεστε σκεφθείτε ότι έχετε ένα ποτήρι γεμάτο νερό κι ένα δεύτερο ποτήρι γεμάτο πορτοκαλάδα. Θέλετε να ανταλλάξετε το περιεχόμενό τους. Τι χρειάζεστε;