

## Βασικές Εντολές

Για εμφάνιση στην οθόνη χρησιμοποιούμε την εντολή **print** με σύνταξη:

```
print "μήνυμα" ή  
print τιμή μεταβλητής
```

Για είσοδο αριθμητικών δεδομένων από το πληκτρολόγιο χρησιμοποιούμε την εντολή **input** με σύνταξη:

```
μεταβλητή=input("μήνυμα")
```

Για είσοδο δεδομένων συμβολοσειράς από το πληκτρολόγιο χρησιμοποιούμε την εντολή **raw\_input** με σύνταξη:

```
μεταβλητή=raw_input("μήνυμα")
```



```
Python 3.11.4 (v3.11.4:d2340ef257, Jun 6 2023, 19:15:51) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin  
Type "help", "copyright", "credits" or "license()" f  
or more information.  
>>> print("Hello World")  
Hello World  
>>>
```

Ας δούμε την πρώτη μας εντολή - προσέξτε πως το περιβάλλον προγραμματισμού χρωματίζει τις εντολές.

Η εντολή `print()` δεν εκτυπώνει κάτι σε χαρτί. Εμφανίζει όμως μηνύματα στην οθόνη μας. Αυτό που εμφανίζει είναι το μήνυμα **"Hello World"**. Με λίγα λόγια, σιδηποτε βάλουμε στην παρένθεση, όταν πατήσουμε το πλήκτρο ENTER στο πληκτρολόγιό μας, θα εμφανιστεί στην οθόνη.

Θα παρατηρήσατε πως το **Hello World** δεν εμφανίζεται με εισαγωγικά. Και πολύ σωστά δεν εμφανίζεται, καθώς τα εισαγωγικά χρησιμεύουν μόνο για να "πουν" στην Python ποιο κείμενο να εμφανίσει στην οθόνη.

Θα δοκιμάσουμε τώρα το ίδιο μήνυμα, αλλά δεν θα κλείσουμε τα εισαγωγικά. Για να δούμε τι θα συμβεί...

Προσέξτε τις διαφορές: η παρένθεση, στη δεύτερη περίπτωση, εμφανίζεται με πράσινο και όχι μαύρο χρώμα.



```
Python 3.11.4 (v3.11.4:d2340ef257, Jun 6 2023, 19:15:51) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin  
Type "help", "copyright", "credits" or "license()" f  
or more information.  
>>> print("Hello World")  
Hello World  
>>> print("Hello World"  
...  
SyntaxError: incomplete input  
>>>
```

Επειδή δεν κλείσαμε τα εισαγωγικά, η Python θεωρεί πως συνεχίζουμε να πληκτρολογούμε κείμενο για εμφάνιση στην οθόνη. Όταν πατήσαμε το πλήκτρο ENTER στο πληκτρολόγιο, η εντολή δεν εκτελέστηκε επειδή υπάρχει πρόβλημα στη σύνταξη της!

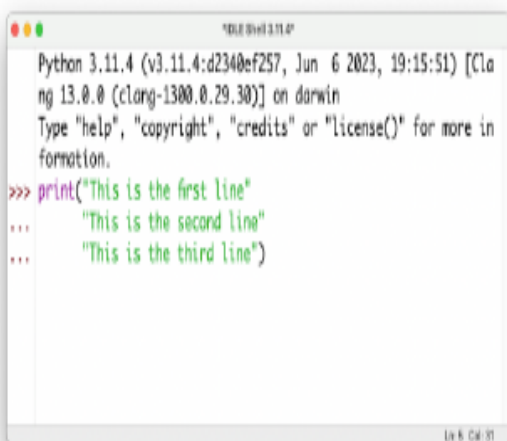
### Η `print()` μέχρι τώρα:

Με την `print()` μπορούμε να εμφανίσουμε κείμενο (ή σειρές χαρακτήρων) πάνω στην οθόνη. Για να εκτελεστεί η εντολή, θα πρέπει, μέσα στην παρένθεση, το κείμενο μας να είναι σε εισαγωγικά.



```
>>> print("This is just a text")
```

Όπως έχουμε δει, κάθε εντολή εκτελείται με το πάτημα του ENTER στο πληκτρολόγιο. Αν θέλουμε να εμφανίσουμε περισσότερες γραμμές κειμένου, ο απλούστερος τρόπος είναι να μην κλείσουμε την παρένθεση. Πατάμε το ENTER στο τέλος της γραμμής και πληκτρολογούμε την επόμενη φράση/γραμμή. Κάθε φορά που πατάμε το ENTER (χωρίς να έχουμε κλείσει την παρένθεση), απλά αλλάζει η γραμμή χωρίς να εκτελεστεί η εντολή.



```
Python 3.11.4 (v3.11.4:d2340ef257, Jun 6 2023, 19:15:51) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more in
formation.
>>> print("This is the first line"
...       "This is the second line"
...       "This is the third line")
```

Στην εικόνα πιο πάνω, προσέξτε τις 3 τελείες στο πλαίσιο - αυτό δείχνει πως η εντολή δεν έχει ακόμη εκτελεστεί.

Όταν όμως κλείσουμε την παρένθεση, η εντολή έχει ολοκληρωθεί και -με το πάτημα του ENTER- θα εκτελεστεί.

Στην εκτέλεση, δε θα εμφανιστεί η μια γραμμή κάτω από την άλλη. Θα εμφανιστεί η μια μετά την άλλη:

This is the first line  
This is the second line  
This is the third line

Αργότερα θα δούμε πώς μπορούμε να εμφανίσουμε κείμενο σε διαφορετικές γραμμές.

## Εργασία με αριθμούς

Μέχρι τώρα χρησιμοποιήσαμε την εντολή `print()` για να εμφανίσουμε στην οθόνη κείμενο. Ας δούμε τι θα συμβεί αν, αντί για κείμενο, χρησιμοποιήσουμε αριθμούς.

```
>>> print(1234)
1234
>>> print("1234")
1234
```

Δείτε το παράδειγμα πιο πάνω: στην πρώτη γραμμή, μέσα στην παρένθεση είχαμε τον αριθμό 1234 χωρίς εισαγωγικά. Στην εκτέλεση, εμφανίστηκε ο αριθμός της παρένθεσης.

Στη δεύτερη εντολή, το "1234" ήταν σε εισαγωγικά. Αμέσως εμφανίστηκε με πράσινο χρώμα. Όταν εκτελέστηκε η εντολή, εμφανίστηκε και πάλι ο αριθμός 1234. Το αποτέλεσμα φαίνεται να είναι το ίδιο, όμως υπάρχει μεγάλη διαφορά, κάτι που θα δούμε στη συνέχεια!

Ας δούμε ξανά το παράδειγμα, αλλά αυτή τη φορά ας χρησιμοποιήσουμε το σύμβολο της πρόσθεσης.

Στην πρώτη γραμμή, όπως φαίνεται και στην εικόνα πιο κάτω, έχουμε την αριθμητική πρόταση σε εισαγωγικά. Η Python εμφανίζει στην οθόνη οτιδήποτε υπάρχει μέσα στα εισαγωγικά, όπως ακριβώς το έχουμε πληκτρολογήσει.

```
>>> print("12+5")
12+5
>>> print(12+5)
17
```

Στη δεύτερη εντολή, που δεν έχουμε χρησιμοποιήσει εισαγωγικά, η Python θεωρεί πως θέλουμε να εμφανίσουμε στην οθόνη, όχι την πράξη 12+5, αλλά το αποτέλεσμα της πράξης. Έτσι, μας εμφανίζει το άθροισμα του 12 συν 5, δηλαδή το 17.

### Προτεραιότητα πράξεων:



Η Python εκτελεί τις εντολές με τη σωστή σειρά (προτεραιότητα πράξεων). Πρώτα πράξεις σε παρενθέσεις, στη συνέχεια πολλαπλασιασμοί και διαιρέσεις και τέλος προσθέσεις και αφαιρέσεις.

```
>>> print((10-2)*2)
```

16

Ας δούμε κάποια παραδείγματα πράξεων:

**Πρόσθεση:** `print(100 + 5)`

**Αφαίρεση:** `print(100 - 5)`

**Πολλαπλασιασμός:** `print(100 * 5)`

**Διαίρεση:** `print(100 / 5)`

**Δυνάμεις:** `print(5**100)`

Αν θέλουμε να κάνουμε πιο ενδιαφέρον το αποτέλεσμα που εμφανίζεται στην οθόνη, μπορούμε να συνδυάσουμε τη μαθηματική πρόταση με το αποτέλεσμα της πράξης.

```
>>> print("12+5=", 12+5)
```

Όταν εκτελεστεί η πιο πάνω εντολή, θα εμφανιστεί στην οθόνη το 12+5= που βρίσκεται μέσα στα εισαγωγικά. Στη συνέχεια, υπάρχει το κόμμα (,) που δηλώνει πως υπάρχει και άλλο τμήμα εντός της παρένθεσης που πρέπει να εκτελεστεί. Επειδή το μέρος που ακολουθεί είναι αριθμοί εκτός εισαγωγικών, η Python θα κάνει την πράξη και θα εμφανίσει το αποτέλεσμα:

```
12+5= 17
```

## Βασικές Συναρτήσεις

Η Python διαθέτει ένα σύνολο ενσωματωμένων συναρτήσεων που μπορούμε να καλούμε άμεσα, όπως οι παρακάτω:

### **float()**

Μετατρέπει ακεραίους και συμβολοσειρές σε δεκαδικούς αριθμούς.

### **int()**

Δέχεται οποιαδήποτε τιμή και τη μετατρέπει σε ακέραιο, κόβοντας τα δεκαδικά ψηφία, αν υπάρχουν.

### **abs()**

Επιστρέφει την απόλυτη τιμή ενός αριθμού.

### **pow(a,b)**

Επιστρέφει τη δύναμη του a υψωμένη στο b.

### **divmod(x,y)**

Επιστρέφει το ηλίκο και το υπόλοιπο της διαίρεσης x/y.

## Αναφορές

- Ευσταθίου Χρήστος, *Αρχές Προγραμματισμού Υπολογιστώ σε γλώσσα Python*, ΕΠΑΛ Τρικάλων, 2021
- Κοφτερός Αλέξανδρος, *Η Python με Απλά Λόγια: Σύντομος οδηγός για αρχάριους*, Λευκωσία, 2023