

Αντικειμενοστρεφής Προγραμματισμός - Python

Κ.Π. Γιαλούρης

Στόχοι του σημερινού μαθήματος

- Εξοικείωση με τα περιβάλλοντα της Python
- Κατανόηση βασικών εννοιών & ΤΕΧΝΙΚΩΝ
 - Τύπος δεδομένων
 - Μεταβλητή
 - Εντολή ανάθεση τιμής /εντολή αντικατάστασης
 - Εισαγωγή δεδομένων
 - Εμφάνιση αποτελεσμάτων

Στόχοι του σημερινού μαθήματος

- Γραφή και εκτέλεση προγράμματος
 - Υπολογισμός μαθηματικών εκφράσεων
 - Είσοδος δεδομένων στον υπολογιστή
 - Εκτύπωση αποτελεσμάτων στην οθόνη
 - Ανάλυση προγραμματισμός ενός απλού προβλήματος

Τύπος Δεδομένων

- ❑ Ο Τύπος Δεδομένων το είδος του δεδομένου που θα αποθηκευτεί σε μία περιοχή της μνήμης.
- ❑ Βασικοί τύποι είναι οι τύποι που τους παρέχει η γλώσσα προγραμματισμού.
- ❑ Ο προγραμματιστής μπορεί να ορίσει δικούς του τύπους δεδομένων οι οποίοι βασίζονται στους βασικούς τύπους.

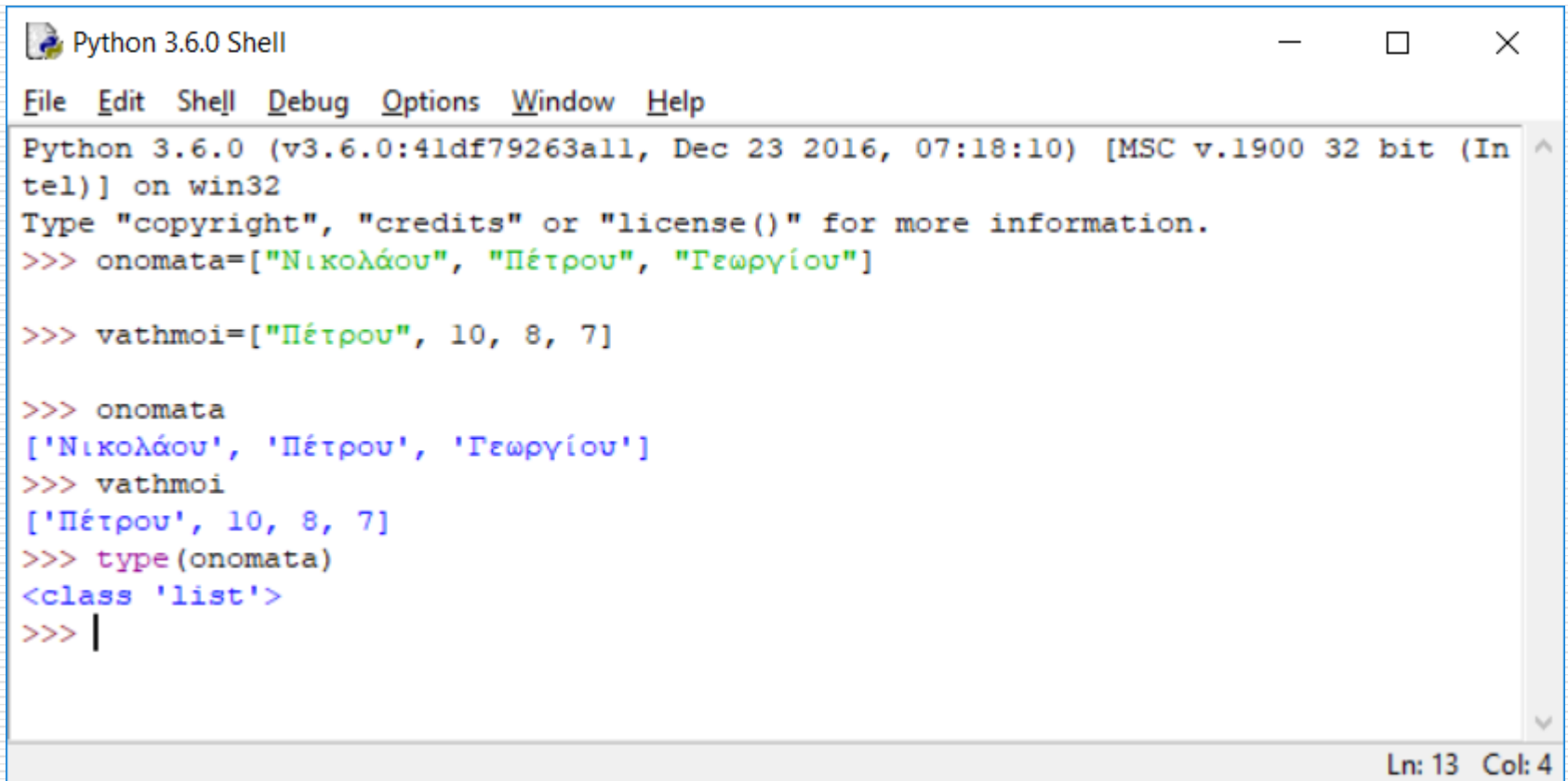
Οι κυριότεροι τύποι /classes δεδομένων στη Python

Τύπος	Περιγραφή
<u>bool</u>	Λογικός τύπος με διακριτές τιμές True ή False
<u>float</u>	Πραγματικοί αριθμοί
<u>int</u>	Ακέρατοι αριθμοί
<u>str</u>	Χαρακτήρες
<u>list</u>	Λίστα
<u>dict</u>	Λεξικό
<u>set</u>	Σύνολο
<u>tuple</u>	Πλειάδα

list -Λίστα

- ❑ Μία λίστα είναι ένα αντικείμενο το οποίο περιέχει πολλαπλά δεδομένα.
- ❑ Μία λίστα είναι ένας τύπος μεταβλητής
- ❑ Μία λίστα είναι ένας δυναμικός τύπος δεδομένων. Αυτό σημαίνει ότι μπορούμε να προσθέσουμε ή να αφαιρέσουμε στοιχεία σε μία λίστα.

list -Λίστα



```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> onomata=["Νικολάου", "Πέτρου", "Γεωργίου"]

>>> vathmoi=["Πέτρου", 10, 8, 7]

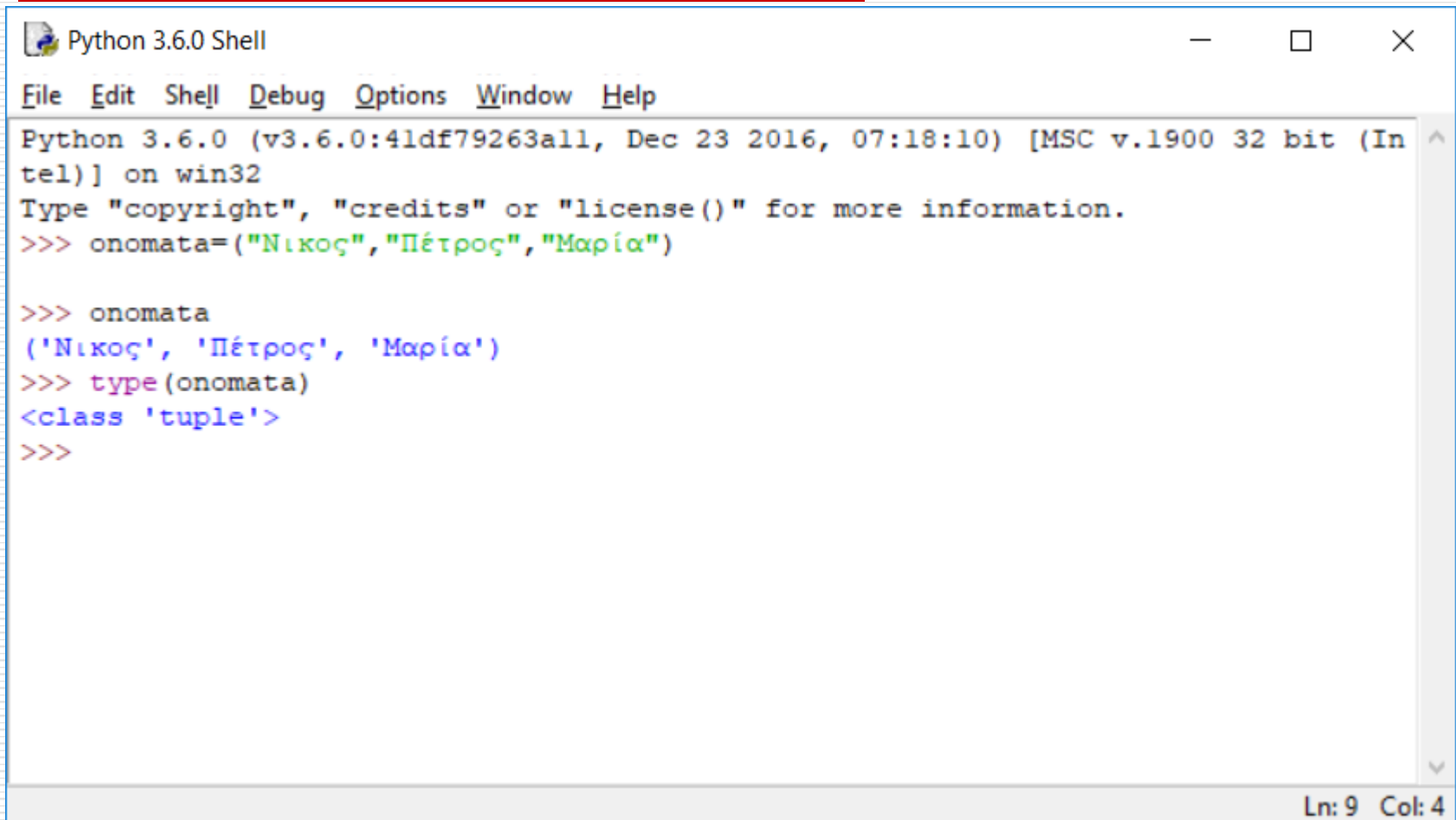
>>> onomata
['Νικολάου', 'Πέτρου', 'Γεωργίου']
>>> vathmoi
['Πέτρου', 10, 8, 7]
>>> type(onomata)
<class 'list'>
>>> |
```

Ln: 13 Col: 4

tuple

- Μία **tuple(πλειάδα)** είναι μία αλληλουχία όπως και η λίστα. Διαφέρει ως προς τη λίστα όσον αφορά στο γεγονός ότι το περιεχόμενό της δεν μπορεί να αλλάξει.
- Δηλώνεται ως εξής:
 - `onomata = ("Νικος", "Πέτρος", "Μαρία")`
 - `b = (4, 3, 5, 6, 2, -3)`
 - `c = (1,)`

tuple - Πλειάδα



```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> onomata=("Νίκος","Πέτρος","Μαρία")

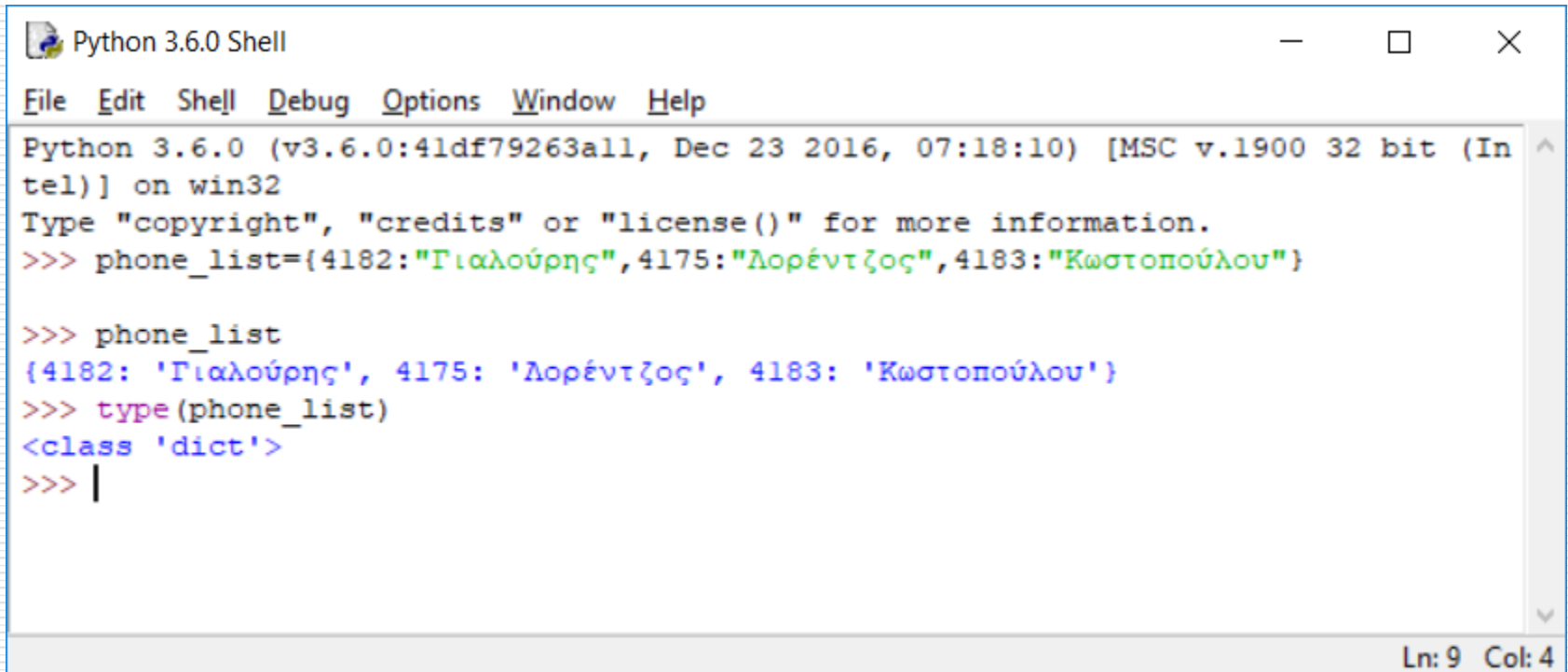
>>> onomata
('Νίκος', 'Πέτρος', 'Μαρία')
>>> type(onomata)
<class 'tuple'>
>>>
```

Ln: 9 Col: 4

dictionary -Λεξικό

- Μία μεταβλητή τύπου `dictionary` (λεξικό) είναι μία συλλογή δεδομένων. Κάθε στοιχείο της συλλογής αυτής αποτελείται από δύο μέρη. Το πρώτο μέρος αναφέρεται ως κλειδί (`key`) και το δεύτερο μέρος ως τιμή(`value`)

Dictionary -Λεξικό



```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> phone_list={4182:"Γιαλούρης",4175:"Λορέντζος",4183:"Κωστοπούλου"}

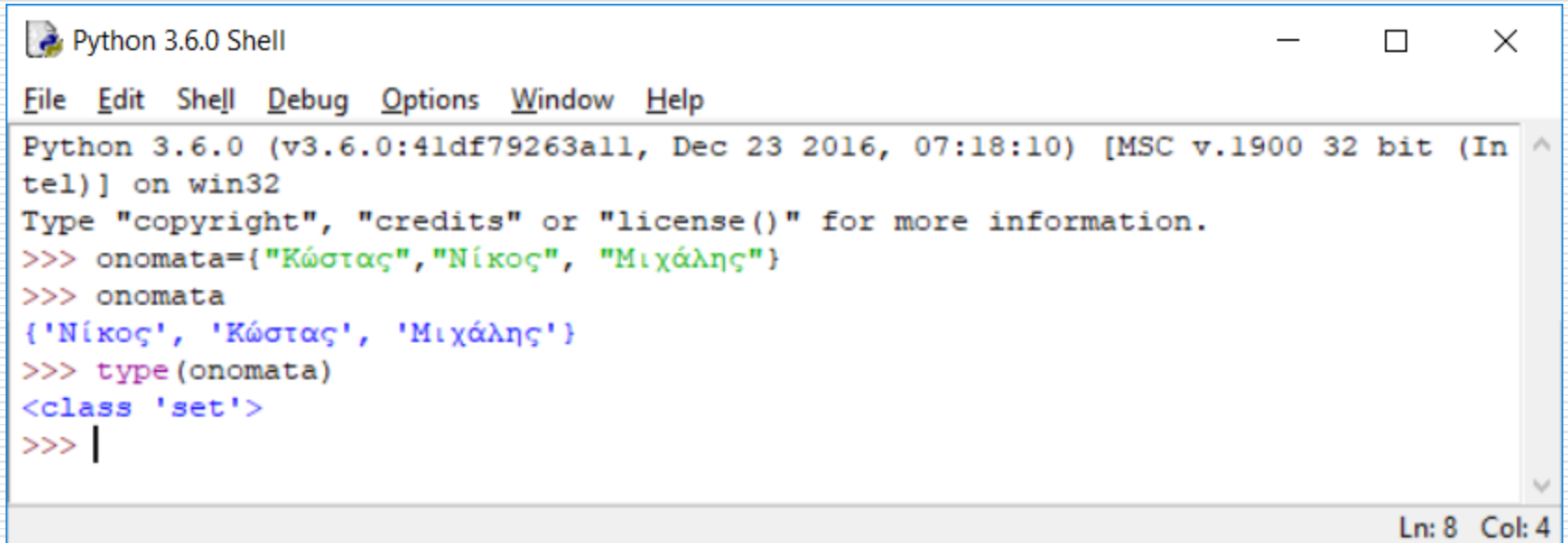
>>> phone_list
{4182: 'Γιαλούρης', 4175: 'Λορέντζος', 4183: 'Κωστοπούλου'}
>>> type(phone_list)
<class 'dict'>
>>> |
```

Ln: 9 Col: 4

set -Σύνολο

- Ένα `set` είναι μία συλλογή δεδομένων από μοναδικές τιμές δηλ. δεν υπάρχουν δύο ίδια στοιχεία σε ένα σύνολο
- Τα στοιχεία ενός συνόλου **δεν είναι διατεταγμένα**
- Τα στοιχεία ενός συνόλου **μπορεί να είναι διαφορετικού τύπου**

set -Σύνολο



```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> onomata={"Κώστας","Νίκος", "Μιχάλης"}
>>> onomata
{'Νίκος', 'Κώστας', 'Μιχάλης'}
>>> type(onomata)
<class 'set'>
>>> |
```

Ln: 8 Col: 4

Η έννοια της μεταβλητής

- Είναι μια προσωρινή θέση αποθήκευσης δεδομένων στο πρόγραμμά μας.
- Μπορούμε να χρησιμοποιούμε μια ή περισσότερες μεταβλητές.
- Αποθηκεύονται λέξεις, αριθμοί, ημερομηνίες χαρακτήρες κλπ.

Η μεταβλητή



Όνομα Μεταβλητής

- ❑ Αρχίζει με λατινικό γράμμα ή την κάτω παύλα (`_`) και αποτελείται από λατινικά γράμματα, αριθμούς, την κάτω παύλα.
- ❑ Δεν πρέπει να περιέχει κενά ή άλλα σύμβολα (π.χ. τελεία).

Όνομα Μεταβλητής

- ❑ Δεν μπορεί να είναι κάποια από τις δεσμευμένες λέξεις ή ονόματα αντικειμένων ή ιδιοτήτων της Python (π.χ. `def`, `for`, `if`, `else` κλπ).
- ❑ Μπορεί να αποτελείται από συνδυασμό να λέξεων που εκφράζουν το περιεχόμενο της μεταβλητής (π.χ. `totalCost`).

Αριθμητικοί Τελεστές

Αριθμητικές πράξεις

- Πρόσθεση
- Αφαίρεση
- Πολλαπλασιασμός
- Διαίρεση
- Πηλίκο διαίρεσης
- Υπόλοιπο διαίρεσης
- Ύψωση σε δύναμη

Τελεστές

+

-

*

/

//

%

**

Εντολη ανάθεσης τιμής

- Στις μεταβλητές καταχωρούνται τιμές με τη χρήση του τελεστή απόδοσης τιμής (=) και εκφράσεων.
- ΌνομαΜεταβλητής = **έκφραση**

Εντολη ανάθεσης τιμής

- Το αριστερό μέλος της εντολής αντιστοιχεί σε μεταβλητή.
- Το δεξιό μέλος της εντολής (έκφραση) μπορεί να είναι:
 - Σταθερά
 - Μεταβλητή
 - Συνδυασμός σταθερών/μεταβλητών/συναρτήσεων με χρήση τελεστών και παρενθέσεων

Αριθμητικοί Τελεστές

Λειτουργία Τελεστή	Τελεστής	Παράδειγμα
Πρόσθεση	+	<code>cost + tax</code>
Αφαίρεση	-	<code>totalCost - cost</code>
Πολλαπλασιασμός	*	<code>unitCost*numProducts</code>
Διαίρεση	/	<code>17/5</code> είναι 3,4
Ακέραια Διαίρεση (πηλίκο)	//	<code>17//5</code> είναι 3
Υπόλοιπο Διαίρεσης	%	<code>17 % 5</code> είναι 2
Υψωση σε Δύναμη	**	<code>x ** 2</code>

Εντολή ανάθεσης τιμής

- `x=3`
- `y=a+b`
- `z=z+3*(a+1)`
- `ονομα="ΓΙΩΡΓΟΣ"`
- `passed=True`

Εντολή ανάθεσης τιμής

- Σε περίπτωση αριθμητικών μεταβλητών μπορούμε να χρησιμοποιούμε την παρακάτω σύνταξη:
- `όνομαΜεταβλητής Θ = Έκφραση`
- Όπου **Θ** ένας οποιοσδήποτε τελεστής αριθμητικής πράξης (+, -, *, /, //, %, **)

Εντολή ανάθεσης τιμής

Εντολή	Ισοδύναμη εντολή	επεξήγηση
$x += a$	$x = x+a$	Αυξάνει το περιεχόμενο της μεταβλητής x κατά a
$x /= 5$	$x=x/5$	Διαιρεί το περιεχόμενο της μεταβλητής x δια 5
$x-=3*(y+a)$	$x=x-3*(y+a)$	Μειώνει το περιεχόμενο της μεταβλητής x κατά το αποτέλεσμα της παράστασης $3*(y+a)$

Εντολη ανάθεσης τιμής

$$y = \frac{2(x+3)^3 + 3(x-1)}{(3x-5)}$$

$$y = (2 * (x+3) ** 3 + 3 * (x-1)) / (3 * x - 5)$$

Εισαγωγή δεδομένων

- Για την εισαγωγή δεδομένων σε ένα πρόγραμμα της Python μπορεί να χρησιμοποιηθεί η συνάρτηση `input`.
- Η `input` είναι μια ειδική συνάρτηση η οποία δίνει δυνατότητα στον χρήστη να εισάγει δεδομένα στο πρόγραμμα κατά την διάρκεια της εκτέλεσης του προγράμματος.

input

Η βασική σύνταξή της είναι:

```
variableName =input ("Μήνυμα")
```

Κατά την εκτέλεση του προγράμματος η συνάρτηση **input** εμφανίζει το *Μήνυμα*, που βρίσκεται εντός εισαγωγικών, στην οθόνη και περιμένει τον χρήστη να πληκτρολογήσει το δεδομένο εισόδου. Η διαδικασία εισόδου ολοκληρώνεται με το πάτημα του κουμπιού **Enter** του πληκτρολογίου. Τότε αναθέτει στην μεταβλητή μία τιμή που αντιστοιχεί σε **string**

input

- Η συνάρτηση `input` αναθέτει στην μεταβλητή μία τιμή που αντιστοιχεί σε `string`.
- Σε περίπτωση που μας ενδιαφέρει ανάγνωση αριθμητικής τιμής τότε θα πρέπει να γίνει μετατροπή σε αριθμητικό τύπο με χρήση συνάρτησης μετατροπής `int` ή `float`

input

π.χ. Είσοδος βαθμολογίας
(πραγματικός αριθμός)

```
vathmos=input("δώσε τη βαθμολογία")  
vathmos=float(vathmos)
```

Ή

```
vathmos=float(input("δώσε τη βαθμολογία"))
```

input

π.χ. Είσοδος πλήθους φοιτητών
(ακέραιος αριθμός)

```
plithos= input("δώσε πλήθος των φοιτητών")  
plithos=int(plithos)
```

Ή

```
plithos=int(input("δώσε πλήθος των φοιτητών"))
```

Εμφάνιση αποτελεσμάτων

`print (δεδομένα_εξόδου)`

Όπου **`δεδομένα_εξόδου`** είναι μία η περισσότερες μεταβλητές, σταθερές ή εκφράσεις διαχωριζόμενες με (,)

Εμφάνιση αποτελεσμάτων

Η ολοκληρωμένη σύνταξη της `print` είναι:

```
print (δεδομένα_εξόδου, sep=string1, end=string2)
```

όπου:

- ❑ `string1`: Ένας ή περισσότεροι χαρακτήρες οι οποίοι θα εμφανίζονται μεταξύ των εκτυπούμενων τιμών. Η προκαθορισμένη τιμή είναι ένα κενό διάστημα.
- ❑ `string2`: Ένας ή περισσότεροι χαρακτήρες οι οποίοι θα εμφανίζονται στο τέλος της γραμμής εκτύπωσης. Η προκαθορισμένη τιμή είναι ο χαρακτήρας διαφυγής `\n` ώστε η εκτύπωση να γίνει στην επόμενη γραμμή εκτύπωσης. Εάν δοθεί οποιοσδήποτε άλλος χαρακτήρας εκτός του χαρακτήρα διαφυγής `\n`, τότε η επόμενη εκτύπωση θα γίνει στην ίδια γραμμή.

Χαρακτήρες Διαφυγής

Χαρακτήρες διαφυγής	Λειτουργία
<code>\n</code>	Η εκτύπωση που θα ακολουθήσει θα γίνει στην επόμενη γραμμή
<code>\t</code>	Η εκτύπωση που θα ακολουθήσει θα γίνει στην tab θέση
<code>\'</code>	Θα γίνει η εκτύπωση του μονό εισαγωγικού <code>'</code>
<code>\"</code>	Θα γίνει η εκτύπωση του διπλού εισαγωγικού <code>"</code>
<code>\\</code>	Θα γίνει η εκτύπωση της ανάποδη μπάρας <code>\</code>

Πρόβλημα

Να γραφεί πρόγραμμα σε Python που

- να διαβάζει από την οθόνη τις ώρες εργασίας και το ωρομίσθιο ενός εργαζόμενου

■ ***Να υπολογίζει***

- τις ακαθάριστες αποδοχές του*
- Τις κρατήσεις για ασφάλιση*
- Τις κρατήσεις φόρου*
- Πληρωτεο ποσό*

Δεδομένα

- *Εισόδου*
 - *Ώρες Εργασίας.*
 - *Ωριαία Αποζημίωση.*
- *Ποσοστά κρατήσεων*
 - *Ποσοστά ασφαλισής*
 - *Ποσοστό Φόρου.*

Ζητούμενα

- *Πληρωτέο Ποσό*

Σχέσεις που συνδέουν Δεδομένα - Ζητούμενα

- **Μισθός** =
(*Ώρες Έργασίας*) * (*Ωριαία Αποζημίωση*)
- **Κρατήσεις Ασφάλισης** =
(*Μισθός*) * (*Ποσοστό Ασφάλισης*)
- **Φορολογητέο ποσό** =
(*Μισθός*) - (*Κρατήσεις Ασφάλισης*)
- **Φόρος** =
(*Φορολογητέο Ποσό*) * (*Ποσοστό Φόρου*)
- **Πληρωτέο Ποσό** =
(*Φορολογητέο Ποσό*) - (*Φόρος*)