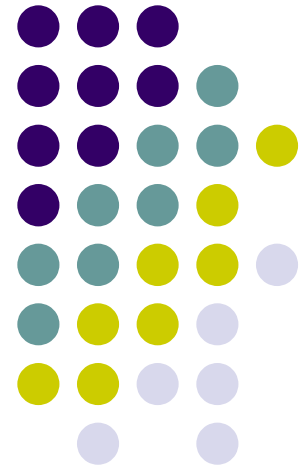
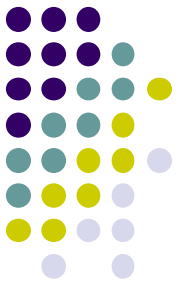


Προγραμματισμός Υπολογιστών - Εισαγωγή

Βασικά στοιχεία γλώσσας
προγραμματισμού
Σχολ. Έτος 2024-25

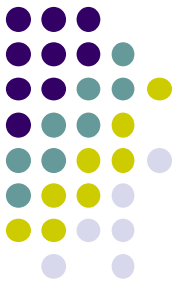


Ιστορία της Python



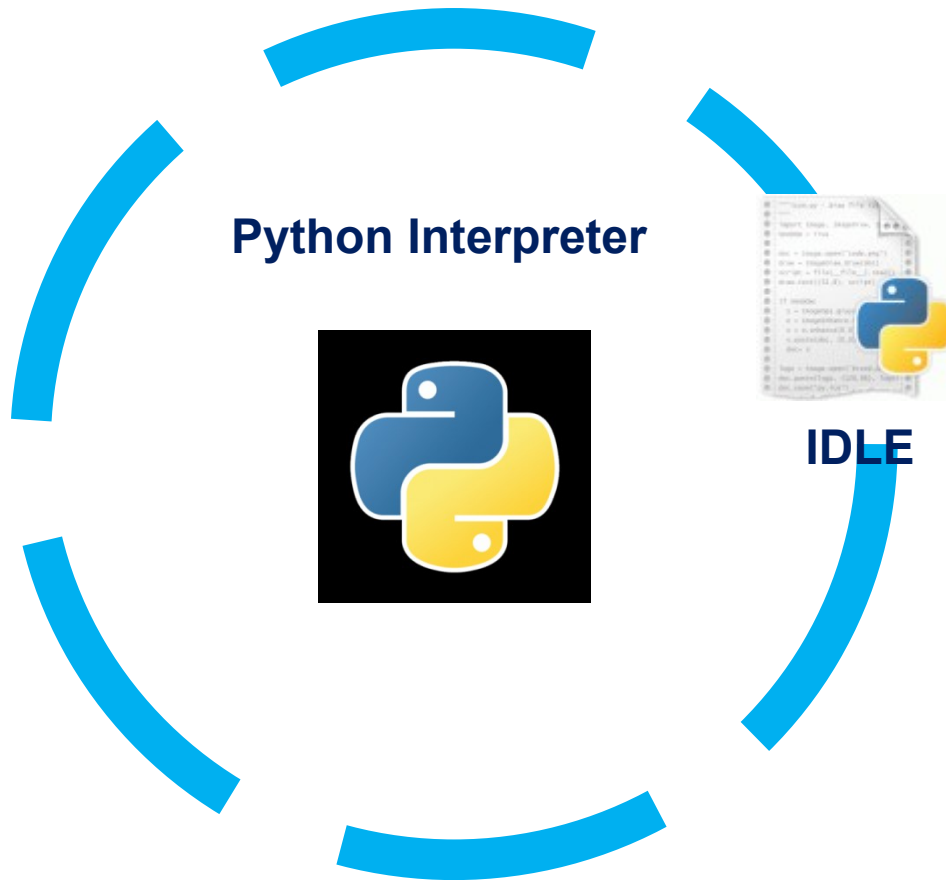
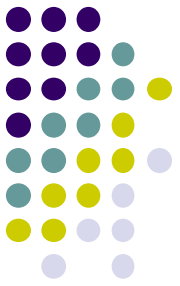
- Δημιουργήθηκε το 1988 από τον Guido Van Rossum
- Το όνομά της από τους Monty Pythons
- Python 1.0 (1994)
- Python 2.0 (2000) - σήμερα 2.7
- Python 3.0 (2008) - σήμερα 3.8
- Η Python 3 τείνει να επικρατήσει, έχει κάποιες μικρές διαφορές από την 2
- Σχολικό βιβλίο και εργαστήριο έκδοση 2.7.10

Χαρακτηριστικά της Python

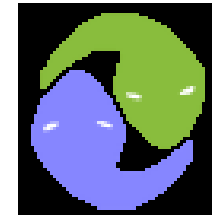


- Απλή στη σύνταξη
- Υποχρεωτική στοίχιση
- Δομές Δεδομένων
- Γλώσσα ανοικτού κώδικα με πλούσιο υλικό υποστήριξης και μεγάλη κοινότητα
- Συνδυασμός προγραμματιστικών παραδειγμάτων (Διαδικαστικός, Συναρτησιακός και Αντικειμενοστραφής προγραμματισμός).

Python Interpreter + IDE

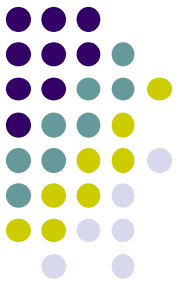


PyCharm

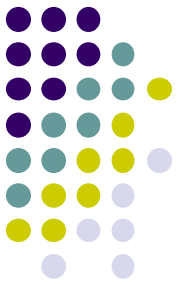


**PyDev
(Eclipse)**

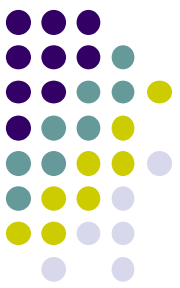
Το περιβάλλον ανάπτυξης προγραμμάτων της Python



- Για τις ανάγκες του μαθήματος θα χρησιμοποιήσουμε την έκδοση 2.7.10 της **Python**.
- Το ολοκληρωμένο περιβάλλον ανάπτυξης προγραμμάτων **IDLE** (*Integrated Development Environment*) της Python.



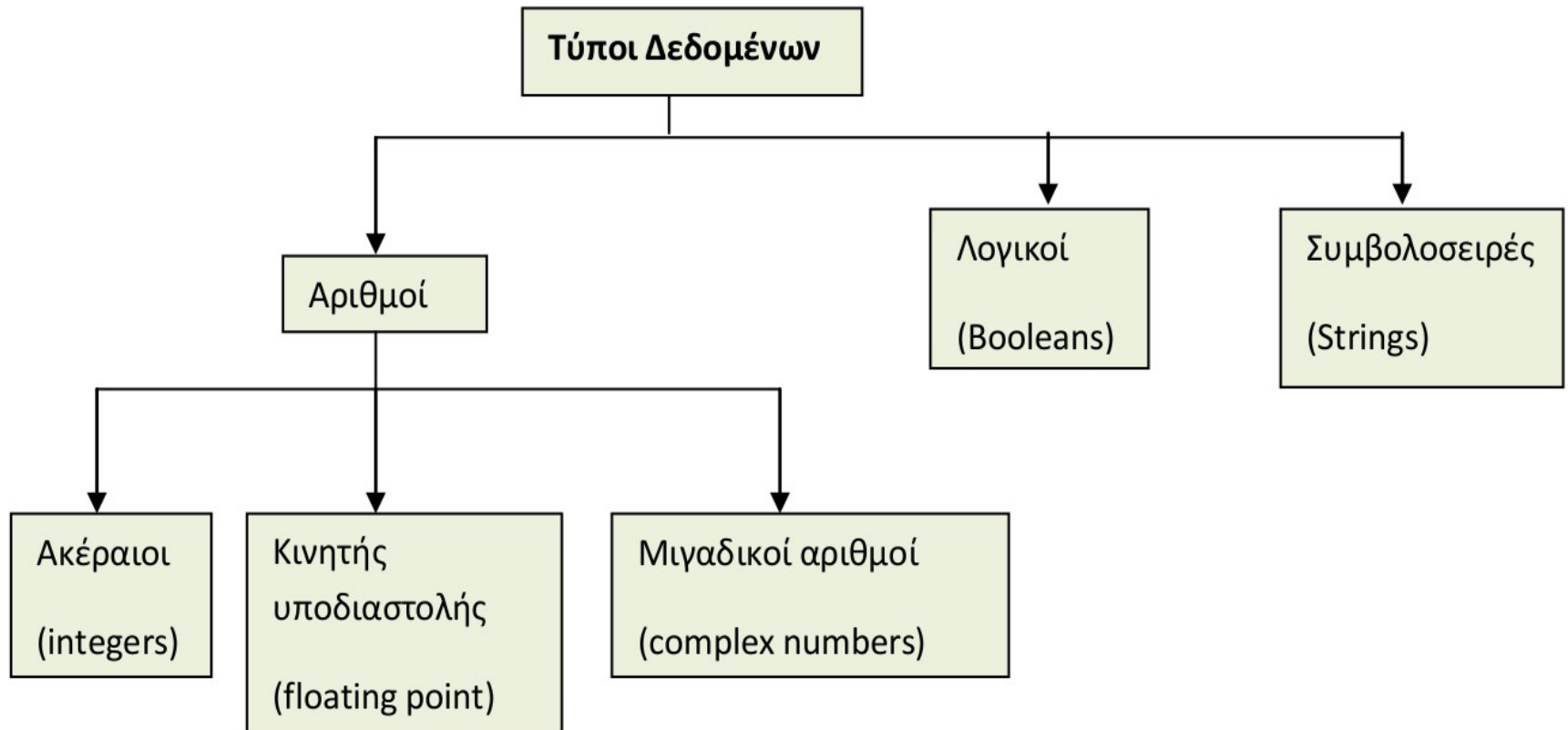
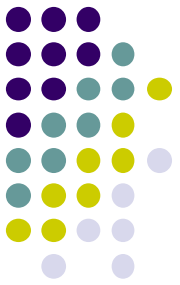
Τύποι Δεδομένων

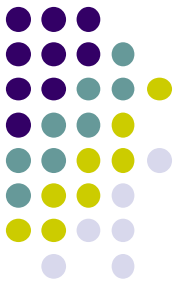


Τύποι δεδομένων

- Ένα πρόγραμμα επεξεργάζεται **δεδομένα** τα οποία μπορεί να είναι αποθηκευμένα στην κύρια μνήμη του υπολογιστή, σε αποθηκευτικό μέσο ή η εισαγωγή τους να γίνεται από κάποια μονάδα.
- Οι **τύποι δεδομένων** προσδιορίζουν τον τρόπο παράστασης των δεδομένων εσωτερικά στον υπολογιστή καθώς και το είδος της επεξεργασίας τους από αυτόν.
- Στην Python δεν δηλώνουμε ρητά τι τύπο δεδομένων χρησιμοποιούμε.
- Στην Python χαρακτηριστικοί τύποι δεδομένων είναι ο **αριθμητικός**, ο **λογικός (boolean)** και **οι συμβολοσειρές** (ή αλφαριθμητικά **strings**).

Τύποι δεδομένων





Ακέραιος τύπος (integer)

- Παράδειγμα ακέραιου αριθμού (**int**) είναι ο 19

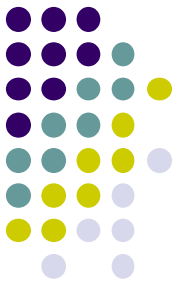
```
>>> type(23)  
<type 'int'>
```

```
>>>>  
type(100000)  
<type 'int'>
```

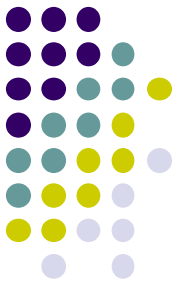
```
>>> type(-17)  
<type 'int'>
```

*Για να ελέγχουμε τον
τύπο δεδομένων
χρησιμοποιούμε την
εντολή **type()***

Αριθμοί κινητής υποδιαστολής

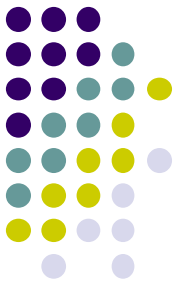


- Παράδειγμα αριθμού κινητής υποδιαστολής (ή **float** για συντομία) είναι ο 256.14
- Παρατήρηση:
- Το δεδαδικό τμήμα διαχωρίζεται αντί για κόμμα ",", " με την τελεία "."
- Άλλο παράδειγμα είναι 28.2E-5 όπου το **E** δηλώνει το **10**, και το **-5** τη δύναμη δηλαδή **28.2 επί 10 εις την -5 (28.2*10⁻⁵)**.



Λογικός τύπος (boolean)

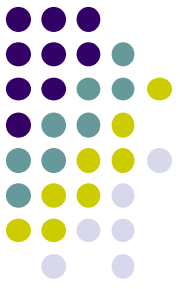
- Ο λογικός τύπος (**bool**) που δέχεται
μόνο δύο τιμές, την τιμή **True** (**Αληθής**)
και την τιμή **False** (**Ψευδής**).



Συμβολοσειρές

- Είναι μια ακολουθία από χαρακτήρες.
- Μια συμβολοσειρά μπορεί να αποτελείται από περισσότερες από μια λέξεις.
- Μπορούμε να ορίσουμε μια γραμματοσειρά με μονά εισαγωγικά ή με διπλά αλλά όχι ανάμικτα.
- Παράδειγμα
- **'Καλημέρα Python'** ή
- **"2210514445"**

Τύποι δεδομένων - Παραδείγματα



- Για να ελέγξουμε τον τύπο δεδομένων χρησιμοποιούμε την εντολή **type()**

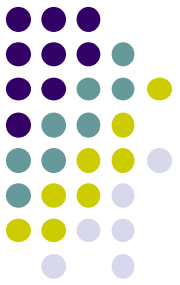
```
>>> type(-3)  
<type 'int'>
```

```
>>>  
type(3.14159)  
<type 'float'>
```

```
>>> type(24.3E7)  
<type 'float'>
```

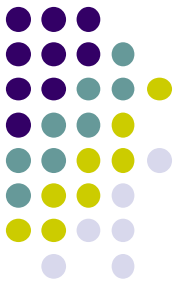
```
>>> type(True)  
<type 'bool'>
```

```
>>>  
type('ΕΠΑΛ')  
<type 'str'>
```

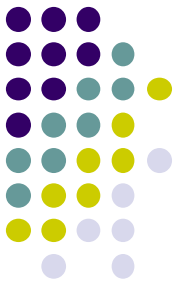


Αριθμητικές και λογικές πράξεις και εκφράσεις

Τελεστές

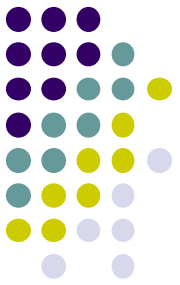


- Οι **τελεστές**(operators) είναι σύμβολα ή λέξεις για τη δημιουργία αριθμητικών και λογικών εκφράσεων.



Αριθμητικοί τελεστές

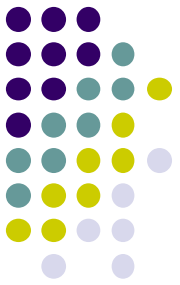
- Είναι τα σύμβολα που χρησιμοποιούμε για να κάνουμε μαθηματικές πράξεις.



Βασικοί αριθμητικοί τελεστές

Πράξη	Σύμβολο
Πρόσθεση	+
Αφαίρεση	-
Πολλαπλασιασμός	*
Διαίρεση	/
Έγξωση σε δύναμη	**
Υπόλοιπο ακέραιας διαίρεσης	%

Αριθμητικοί τελεστές - Παραδείγματα



```
>>> 4*10
```

```
40
```

```
>>> 45/10
```

```
4
```

```
>>> 45.0/10
```

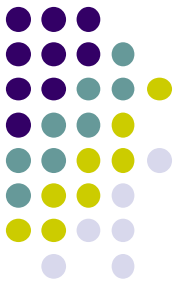
```
4.5
```

```
>>> 45 % 10
```

```
5
```

```
>>> 2**3
```

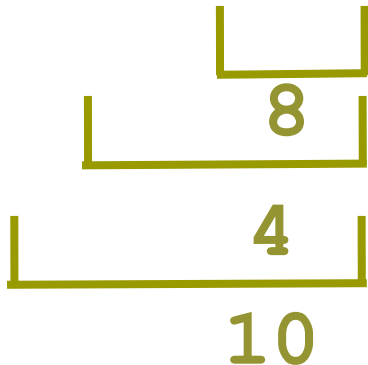
```
8
```

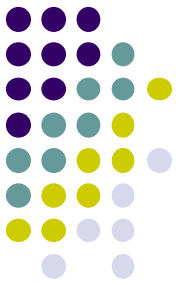


Αριθμητικές εκφράσεις (1)

Προτεραιότητα πράξεων

- Έγωση σε δύναμη
- Πολλαπλασιασμός, διαίρεση, υπόλοιπο ακέραιας διαίρεσης
- Πρόσθεση, αφαίρεση
- $6+32/2^{**}3$





Αριθμητικές εκφράσεις (2)

Αν υπάρχουν παρενθέσεις εκτελούνται πρώτα οι πράξεις μέσα σ' αυτές με την γνωστή προτεραιότητα.

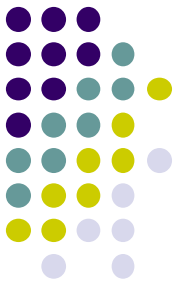
- Παράδειγμα (με παρένθεση)

$$\begin{array}{c} \overline{\quad} \\ (2+3)*5 \\ \quad \underline{5} \\ \quad \quad \underline{25} \end{array}$$

- Παράδειγμα (χωρίς παρένθεση)

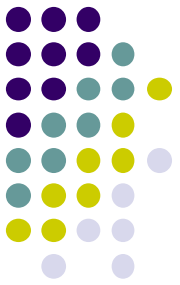
$$\begin{array}{c} \overline{\quad} \\ 2+3*5 \\ \quad \underline{15} \\ \quad \quad \underline{17} \end{array}$$

Σχεσιακοί (ή συγκριτικοί) τελεστές



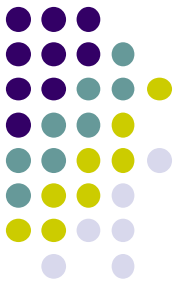
- Χρησιμοποιούνται για τη σύγκριση δύο τιμών ή μεταβλητών, με το αποτέλεσμα μιας σύγκρισης να είναι:
- **True** (Αληθής)
- **False** (Ψευδής)

Βασικοί σχεσιακοί (ή συγκριτικοί) τελεστές



Σύγκριση	Σύμβολο
Μικρότερο από	<
Μικρότερο ή ίσο από	<=
Μεγαλύτερο από	>
Μεγαλύτερο ή ίσο από	>=
Ίσο από	==
Διάφορο από	!=

Σχεσιακοί τελεστές - Παραδείγματα



```
>>> 23==23
```

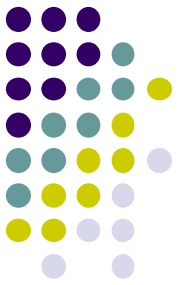
```
True
```

```
>>> 34!=45
```

```
True
```

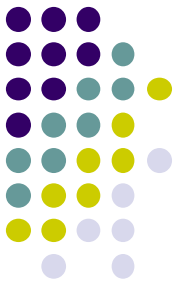
```
>>> 56<=12
```

```
False
```



Τελεστές λογικών πράξεων

- Στις λογικές πράξεις και εκφράσεις χρησιμοποιούνται οι λογικοί τελεστές **not**, **and**, **or** με τις ακόλουθες λογικές λειτουργίες:
- **not (ΟΧΙ)**: πράξη άρνησης
- **and (ΚΑΙ)**: πράξη σύζευξης
- **or (Ή)**: πράξη διάζευξης
- Το αποτέλεσμα μιας λογικής πράξης είναι **True** ή **False**.

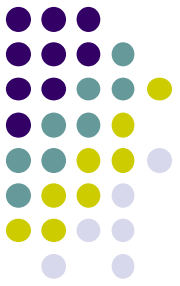


Πίνακες Αληθείας

- Οι Πίνακες αλήθειας των λογικών πράξεων μεταξύ δύο λογικών μεταβλητών **P** και **Q** είναι:

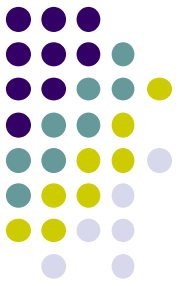
P	Q	P and Q	P or Q	not P
True	True	True	True	False
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True

Λογικοί τελεστές - Προτεραιότητα



Πράξη	Προτεραιότητα
not	1η
and	2η
or	3η

Λογικές εκφράσεις - Παραδείγματα



```
>>> (12<11) and (23>10)
```

False

```
>>> (12<11) or (23>10)
```

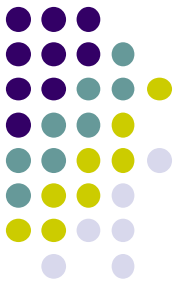
True

```
>>> not(56<=12)
```

True



Μεταβλητές



Μεταβλητές

- Αρκετές φορές σε ένα πρόγραμμα απαιτείται να αποθηκεύσουμε προσωρινά κάποια δεδομένα στη μνήμη του υπολογιστή. Για το σκοπό αυτό χρησιμοποιούμε τις **μεταβλητές**.
- Για τη χρησιμοποίηση μιας μεταβλητής δεν απαιτείται η δήλωσή της, ενώ μπορεί να εκχωρήσουμε διαφορετικούς τύπους τιμών (ακέραιες, κινητής υποδιαστολής, συμβολοσειρές, κ.ά.).

Ονόματα

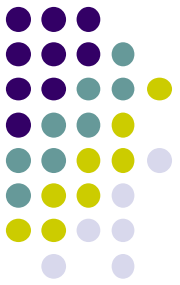


- Για να χρησιμοποιήσουμε μια μεταβλητή απαιτείται να τις δώσουμε ένα όνομα.

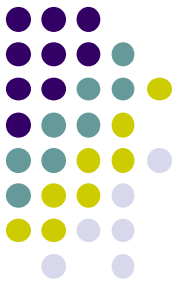
Κανόνες Ονομάτων

- Μπορεί να αποτελούνται από γράμματα του λατινικού αλφαβήτου (**A - Z, a - z**), τον χαρακτήρα **_** και τα ψηφία (**0 - 9**),
- αρκεί ο **πρώτος χαρακτήρας** να **μην είναι ψηφίο** και
- η λέξη να μην είναι **δεσμευμένη**

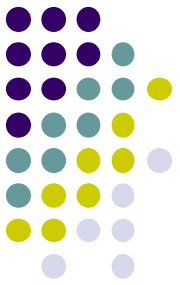
Ονόματα - Παραδείγματα



- onoma1
- onoma_2
- mesi_timi
- timi
- embado
- num_3
- eponymo_a
- eponymo_b

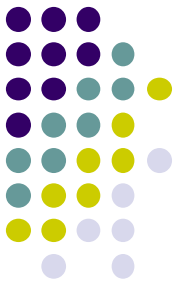


Βασικές Εντολές



Εκχώρηση τιμής

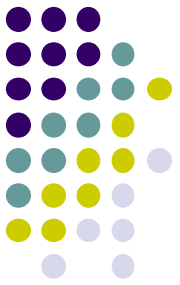
- Για την **εκχώρηση τιμής σε μία μεταβλητή** χρησιμοποιούμε το «=», με μορφή:
ονομα_μεταβλητής=τιμή μεταβλητής
- *Λειτουργία:*
- Καταχωρείται η τιμή του δεξιού μέλους, στη μεταβλητή με το όνομα που έχουμε ορίσει στο αριστερό μέλος.



Πολλαπλή εκχώρηση

- *μορφή:*
 όνομα μεταβλητών = τιμές
- Παραδείγματα
- $a = b = c = 0$ #τα a, b, c λαμβάνουν τις τιμή 0
- $\text{name, age} = \text{"Άλαν", 13}$ #το name λαμβάνει την τιμή Άλαν και το age την τιμή 13

Εκχώρηση τιμής - Παραδείγματα



$x=1$ #το x λαμβάνει την τιμή 1

$x=x+10$ #το x αυξάνει την τιμή του κατά 10

$onoma='Λία'$ #το $onoma$ λαμβάνει την τιμή *Λία*

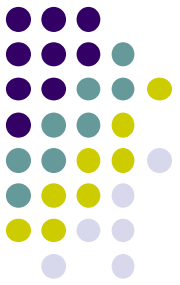
$m1=m2=m3=21$ #τα $m1, m2, m3$ λαμβάνουν την
τιμή 21

$x, y=10, 'Τούλα'$ #το x λαμβάνει την τιμή 10 και το y
την τιμή *Τούλα*

$x, y, z=2, 9, 13$ #τα x, y, z λαμβάνουν τις τιμές 2, 9, 13
αντίστοιχα

Παράδειγμα

Η εντολή $x += 1$

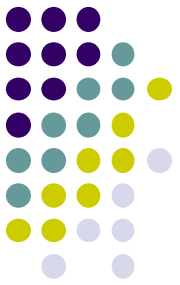


Στην εντολή $x = x + 1$ εμφανίζεται η ίδια μεταβλητή και στις δυο πλευρές της εκχώρησης.

Πρώτα υπολογίζεται η έκφραση που βρίσκεται στο δεξί μέρος και στη συνέχεια το αποτέλεσμα καταχωρείται στη μεταβλητή της εκχώρησης.

Μπορούμε να χρησιμοποιήσουμε αντί για την εντολή $x = x + 1$

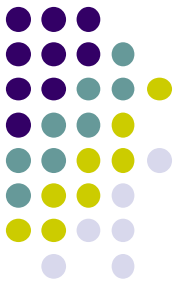
την σύντομη εντολή $x += 1$



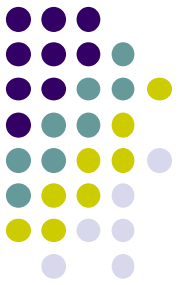
Βασικές Εντολές

- Για την **εμφάνιση τιμών** στην **οθόνη** του υπολογιστή χρησιμοποιούμε την εντολή **print**.
- Μορφές της **print**:
- **print** όνομα_μεταβλητής, ή
- **print** αριθμός, ή
- **print** 'συμβολοσειρά', ή
- **print** ονομα_μεταβλητής (τελεστής) αριθμός

Εισαγωγή τιμής σε μεταβλητή από το πληκτρολόγιο

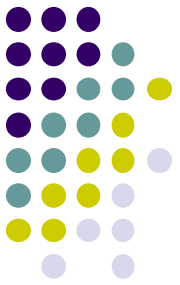


- Για την εισαγωγή τιμής σε μια μεταβλητή από το πληκτρολόγιο, κατάσταση όπου αναμένει από το χρήστη να εισάγει μία τιμή από το πληκτρολόγιο, την οποία την αποδίδει αυτόματα στη μεταβλητή, χρησιμοποιούμε δύο βασικές συναρτήσεις: την `input()` και την `raw_input()`.
- *Μορφή:*
- **Όνομα_μεταβλητής = `input`("κείμενο προς εμφάνιση")**
- Διαβάζει από το πληκτρολόγιο έναν αριθμό από τον χρήστη και τον αποθηκεύει στη μεταβλητή `x`.
- *Παράδειγμα*
$$x = \text{input}(\text{"Δώσε έναν αριθμό: "})$$



raw_input

- Αν θέλουμε να εισάγουμε ένα αλφαριθμητικό (συμβολοσειρά) χρησιμοποιούμε την εντολή `raw_input`:
- *Μορφή:*
- **Όνομα_μεταβλητής = `raw_input`("κείμενο προς εμφάνιση")**
- *Παράδειγμα:*
- `name = raw_input("Δώσε το όνομά σου : ")`
- Γενικά ότι εισάγεται με τη `raw_input` θεωρείται αυτόματα αλφαριθμητικό ενώ η `input` προσπαθεί να το υπολογίσει. Για παράδειγμα αν δώσουμε στην `input` το όνομα μια μεταβλητής θα επιστρέψει το περιεχόμενο της μεταβλητής.



Εισαγωγή σχολίων

- Τα σχόλια σε ένα πρόγραμμα διευκολύνουν την κατανόησή του.
- Στην *Python* τα σχόλια εισάγονται θέτοντας μπροστά από αυτά το σύμβολο **#**
- Τα σχόλια μπορούν να αρχίζουν και μετά από εντολές στη μέση μιας γραμμής. Ότι βρίσκεται δεξιά από το σύμβολο **#**, αγνοείται από το διερμηνευτή.
- Παράδειγμα
Παράδειγμα
`x = input("Δώσε έναν αριθμό: ")` **# Εισαγωγή αριθμού**