

Python Βασικές Εντολές

Οι ακέραιοι αριθμοί αριθμοί (π.χ. 2, 4, 20) έχουν τύπο int, εκείνοι με ένα κλασματικό μέρος (π.χ. 5.0, 1.6) έχουν τον τύπο float.

Ο τελεστής της διαίρεσης (/) επιστρέφει πάντα έναν float αριθμό. Για να το κάνετε διαίρεση και να πάρετε ένα ακέραιο αποτέλεσμα (απορρίπτοντας κάθε κλασματικό αποτέλεσμα), μπορείτε να χρησιμοποιήσετε το (/ /). Για να υπολογίσετε το υπόλοιπο μπορείτε να χρησιμοποιήσετε το (%):

```
>>> 17 / 3 # classic division returns a float
5.666666666666667
>>>
>>> 17 // 3 # floor division discards the fractional part
5
>>> 17 % 3 # the % operator returns the remainder of the division
2
>>> 5 * 3 + 2 # result * divisor + remainder
17
```

Με την Python είναι δυνατόν να χρησιμοποιήσετε τον τελεστή (**) για να υπολογίσετε δυνάμεις:

```
>>> 5 ** 2 # 5 squared
25
>>> 2 ** 7 # 2 to the power of 7
128
```

Το σύμβολο της ισότητας (=) χρησιμοποιείται για να εκχωρήσετε μια τιμή σε μια μεταβλητή. Στη συνέχεια, δεν εμφανίζεται κανένα αποτέλεσμα πριν από την επόμενη διαδραστική προτροπή:

```
>>> width = 20
>>> height = 5 * 9
>>> width * height
900
```

Υπάρχει πλήρης υποστήριξη για float κινητής υποδιαστολής. Φορείς με τελεστές μικτού τύπου μετατρέπονται σε float κινητής υποδιαστολής:

```
>>> 3 * 3.75 / 1.5
7.5
>>> 7.0 / 2
3.5
```

Όταν χρησιμοποιείτε η Python ως υπολογιστής χειρός, είναι κάπως πιο εύκολο να συνεχίσει κανείς τους υπολογισμούς του με τη χρήση της μεταβλητής _, αφού η τελευταία έντυπη έκφραση αποδίδεται στη μεταβλητή αυτή. Για παράδειγμα:

```
>>> tax = 12.5 / 100
>>> price = 100.50
>>> price * tax
12.5625
>>> price + _
113.0625
>>> round(_, 2)
113.06
```

Εκτός από τους αριθμούς, η Python μπορεί να χειριστεί και αλφαριθμητικά, τα οποία μπορούν να εκφραστούν με διάφορους τρόπους. Μπορούν να περικλείονται σε μονά εισαγωγικά ('...') ή διπλά εισαγωγικά ("..."), με το ίδιο αποτέλεσμα.

```
>>> 'spam eggs' # single quotes
'spam eggs'
>>> 'doesn\'t' # use \' to escape the single quote...
"doesn't"
>>> "doesn't" # ...or use double quotes instead
"doesn't"
>>> '"Yes," he said.'
'"Yes," he said.'
>>> "\'Yes,\" he said."
'"Yes," he said.'
>>> '"Isn\'t," she said.'
'"Isn't," she said.'
```

Δύο ή περισσότερες συμβολοσειρές η μία δίπλα στην άλλη αυτόματα αποτελούν μία ενιαία συνεχόμενη συμβολοσειρά.

```
>>> 'Py' 'thon'
'Python'
```

Αυτό λειτουργεί μόνο με string, αλλά ΟΧΙ με μεταβλητές ή εκφράσεις:

```
>>> prefix = 'Py'
>>> prefix 'thon' # can't concatenate a variable and a string literal
...
SyntaxError: invalid syntax
>>> ('un' * 3) 'ium'
...
SyntaxError: invalid syntax
```

Αν θέλετε να ενώσετε μεταβλητές ή μια μεταβλητή και ένα string, χρησιμοποιήστε +:

```
>>> prefix + 'thon'  
'Python'
```

Τα παραπάνω χαρακτηριστικά είναι ιδιαίτερα χρήσιμα όταν θέλετε να διαιρέσετε σε μικρότερα τμήματα strings μεγάλου μεγέθους:

```
>>> text = ('Put several strings within parentheses '  
'to have them joined together.')  
>>> text  
'Put several strings within parentheses to have them joined together.'
```

Τα strings μπορούν να αναπροσαρμόζονται (subscripted), σύμφωνα με τον πρώτο χαρακτήρα που έχει δείκτη 0. Δεν υπάρχει ξεχωριστός τύπος χαρακτήρα. Ένας χαρακτήρας είναι απλά μια συμβολοσειρά μεγέθους ένα(1):

```
>>> word = 'Python'  
>>> word[0] # character in position 0  
'P'  
>>> word[5] # character in position 5  
'n'
```

Οι δείκτες μπορεί επίσης να είναι αρνητικοί αριθμοί. Σε αυτή την περίπτωση αρχίζουν να μετρούν από τα δεξιά προς τα αριστερά:

```
>>> word[-1] # last character  
'n'  
>>> word[-2] # second-last character  
'o'  
>>> word[-6]  
'P'
```

Σημειώστε ότι εφόσον το -0 είναι το ίδιο με το 0, οι αρνητικά δείκτες ξεκινούν από -1.

Επιπλέον από την Python υποστηρίζεται ο τεμαχισμός, ο οποίος επιτρέπει να αποκτήσετε ένα τμήμα της συμβολοσειράς:

```
>>> word[0:2] # characters from position 0 (included) to 2 (excluded)  
'Py'  
>>> word[2:5] # characters from position 2 (included) to 5 (excluded)  
'tho'
```

Σημειώστε πως η αρχή περιλαμβάνεται πάντα, ενώ το τέλος εξαιρείται. Αυτό εξασφαλίζει ότι `s[:i] + s[i:]` είναι πάντα ίσο με `s`:

```
>>> word[:2] + word[2:]  
'Python'  
>>> word[:4] + word[4:]  
'Python'
```

Ένα τμήμα δεικτών έχει χρήσιμες προεπιλογές. Είναι δυνατόν να παραλείπεται το πρώτο στοιχείο, όταν η τιμή του δείκτη ισούται με μηδέν ή να παραλείπεται το τέλος αυτού αφού το μέγεθος της συμβολοσειράς είναι σταθερό.

```
>>> word[:2] # character from the beginning to position 2 (excluded)  
'Py'  
>>> word[4:] # characters from position 4 (included) to the end  
'on'  
>>> word[-2:] # characters from the second-last (included) to the end  
'on'
```

Ένας τρόπος για να θυμάστε πώς λειτουργούν οι τομές είναι να σκεφτούμε τους δείκτες που δείχνουν μεταξύ των χαρακτήρων, με το αριστερό άκρο του πρώτου χαρακτήρα αριθμούνται από το 0. Τότε η δεξιά ακμή του τον τελευταίο χαρακτήρα μιας συμβολοσειράς N χαρακτήρων έχει δείκτη n, για παράδειγμα:

```
+---+---+---+---+---+---+  
| P | y | t | h | o | n |  
+---+---+---+---+---+---+  
0 1 2 3 4 5 6  
-6 -5 -4 -3 -2 -1
```

Η πρώτη σειρά των αριθμών δίνει τη θέση των δεικτών από 0 έως 6 της συμβολοσειράς, η δεύτερη γραμμή δίνει τους αντίστοιχους αρνητικούς δείκτες. Τα τμήματα από i έως j αποτελούνται από όλους τους χαρακτήρες μεταξύ των άκρων i και j, αντίστοιχα.

Για τους μη-αρνητικούς δείκτες, το μήκος ενός τμήματος είναι η διαφορά των δεικτών, με την προϋπόθεση ότι και οι δύο είναι εντός των ορίων. Για παράδειγμα, το μήκος της λέξης [1:03] είναι 2.