**Ενότητα :** Βασικοί τύποι και πράξεις

**Σχέδιο Δραστηριότητας:** Μια πρώτη εισαγωγή στην Python

**Δραστηριότητα 1 :** *Τύποι και Πράξεις*

Ανοίξτε τον διερμηνευτή της Python και δώστε τις παρακάτω εντολές. Πριν εκτελέσετε τις εντολές προσπαθήστε να μαντέψετε τα αποτελέσματα. Πόσα από αυτά πετύχατε;

Να γράψετε δίπλα από κάθε εντολή το σωστό αποτέλεσμα

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| >>> 2 + 4  >>> 3\*4  >>> 7 – 3  >>> 2\*\*3  >>> 3\*\*9 |  | >>> 8.0 / 4  >>> 8 / 4.0  >>> 8.0 / 4.0  >>> 8 / 4  >>> 8 % 4 |  | >>> 92 % 100  >>> 92 / 100  >>> 20.0 / 7  >>> 20 / 7.0  >>> 20 / 7 |  | >>> 162 / 100  >>> 162 % 100  >>> 162.0 / 100  >>> 82 / 100  >>> 82 % 100 |

Καθώς εκτελείτε τις παραπάνω εντολές απαντήστε στις παρακάτω ερωτήσεις:

1. Τι υπολογίζει ο τελεστής / ; Επιτελεί πάντα την ίδια λειτουργία; Από τι εξαρτάται;

Σε τι διαφέρει η έκφραση 1/2 από την 1.0 / 2.0 ;

|  |
| --- |
|  |

1. Ποια είναι η λειτουργία του τελεστή % ;

|  |
| --- |
|  |

1. Πως ονομάζονται όλοι οι παραπάνω τελεστές που χρησιμοποιήσαμε ;

|  |
| --- |
|  |

**Δραστηριότητα 2 :** *Τύποι και Πράξεις ( συνέχεια )*

Ανοίξτε τον διερμηνευτή της Python και δώστε τις παρακάτω εντολές. Πριν εκτελέσετε τις εντολές προσπαθήστε να μαντέψετε τα αποτελέσματα. Πόσα από αυτά πετύχατε;

Να γράψετε δίπλα από κάθε εντολή το σωστό αποτέλεσμα

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| >>> “python” + “2.7”  >>> 3 \* “python”  >>> 3 \* 4  >>> 3 \* “4” |  | >>> 2 + 8  >>> “2” + “8”  >>> “9” + “8” + “2”  >>> 9 + 8 + 2 |  | >>> “a” + “b” \* 3  >>> 2 > 1  >>> 1 < 2  >>> 2 <= 2 |  | >>> 2 == 1+1  >>> not (1 == 2)  >>> 4==4 or 2==1  >>> 4==4 and 2==1 |

Ποιοι είναι οι συγκριτικοί και ποιοι οι λογικοί τελεστές;

|  |
| --- |
|  |

Να περιγράψετε την λειτουργία του τελεστή ( **+** ) όταν βρίσκεται μεταξύ λέξεων

|  |
| --- |
|  |

**Ερώτηση :** Σε τι διαφέρουν οι παρακάτω εκφράσεις; Ποια από τις δυο καταλαβαίνετε καλύτερα; Ποια θα χρησιμοποιούσατε;

**2 != 3 ή not ( 2 == 3 )**

**Δραστηριότητα 3 :** *Μετατροπές μεταξύ τύπων*

Ανοίξτε τον διερμηνευτή της Python και δώστε τις παρακάτω εντολές. Πριν εκτελέσετε τις εντολές προσπαθήστε να μαντέψετε τα αποτελέσματα. Πόσα από αυτά πετύχατε;

Να γράψετε δίπλα από κάθε εντολή το σωστό αποτέλεσμα

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| >>> type( 5 )  >>> type( 3.14 )  >>> type( “python” )  >>> type( 1 + 1 == 2 )  >>> type( “False” )  >>> type( False )  >>> type( [2,3,4] ) |  | >>> [2, 3, 4] + [6,1]  >>> int( 3.141 )  >>> int(False )  >>> int(True )  >>> bool( 1 )  >>> bool( 1024 )  >>> bool( 0 ) |  | >>> 3 + 4  >>> str( 3 ) + str( 4 )  >>> 0 + 1  >>> str( 0 ) + str( 1 )  >>> “False” + “True”  >>> False + True  >>> ‘0’ + 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| >>> len( “” )  >>> len( “ ” )  >>> len( ‘python’ ) |  | >>> len( ‘0’ )  >>> len( ‘124’ )  >>> len( ‘26786’ ) |  | >>> len( [1, 2, 3] )  >>> len( [ ] )  >>> str( 124 ) |

Να εξηγήσετε τα παραπάνω αποτελέσματα και στη συνέχεια να αναφέρετε τους τύπους δεδομένων της Python που έχετε συναντήσει ως τώρα

|  |
| --- |
|  |

Ποια λειτουργία επιτελεί ο τελεστής \*\* ; Τι υπολογίζει η έκφραση 2\*\*1000 ; Μπορεί να υπολογιστεί από την Python; Πόσα ψηφία έχει;

|  |
| --- |
|  |

*Πηγή: Ευριπίδης Βραχνός*