

ΗΜΥ01Κ06
Επιστημονικός Προγραμματισμός με Python







Τυχαία επιλεγμένες επαναληπτικές ερωτήσεις
εφ' όλης (σχεδόν) της ύλης

Φθινόπωρο 2025

1/15 (Σωστό / Λάθος)

Αν προσπαθήσουμε να τρέξουμε ένα πρόγραμμα της Python που περιέχει συντακτικά λάθη, τότε:

-  Το πρόγραμμα δεν ξεκινάει καν να εκτελείται
-  Το πρόγραμμα εκτελείται κανονικά μέχρι να χτυπήσει στο πρώτο συντακτικό λάθος που θα συναντήσει
-  Υπάρχει περίπτωση το πρόγραμμα να εκτελεστεί μέχρι τέλους χωρίς να χτυπήσει πουθενά
-  Αν ένα πρόγραμμα σε Python εκτελείται κανονικά σε συστηματική βάση, τότε το πρόγραμμα δεν περιέχει συντακτικά λάθη

2/15 (Σωστό / Λάθος)

Ένα *αντικείμενο* της Python

- ✓ έχει ένα και μοναδικό *τύπο* που δεν μπορεί να αλλάξει
- ✓ έχει μια και μοναδική *ταυτότητα* που δεν μπορεί να αλλάξει
- ✗ έχει *περιεχόμενο* που μπορεί πάντα να αλλάζει
- ✓ μπορεί να έχει *περισσότερους από έναν προσδιοριστές* να αναφέρονται σε αυτό
- ✓ μπορεί να μην έχει *κανέναν προσδιοριστή* να αναφέρεται σε αυτό

3/15

Ποια από τις δυο αριθμητικές εκφράσεις είναι μεγαλύτερη;

(α) $10+20*30$

(β) $(10+20)*30$

Ποια από τις δυο αριθμητικές εκφράσεις είναι μεγαλύτερη;

(α) $10*8//5$

(β) $10*(8//5)$

Ποια από τις δυο αριθμητικές εκφράσεις είναι μεγαλύτερη;

(α) $3 \ll 3$

(β) $3 ** 3$

$3*2^3 = 3*8 = 24$

$3^3 = 27$

4/15

Είναι συντακτικά σωστές ή όχι οι επόμενες εντολές στην Python;

✓ `a = b = c = 12`

✓ `x < y < z`

✓ `'a' + 'b'`

✗ `'a' * 'b'`

✓ `'a' * 3`

✗ `10.0 << 2`

5/15

Αν $s = \text{'Python'}$ τότε

ποιά η τιμή του $s[-1] + s[0]$

(α) 'nP'

(β) 'oP'

(γ) συντακτικό λάθος

ποιά η τιμή του $s[:-1] + s[:0]$

(α) 'Python'

(β) 'P'

(γ) 'Pytho'

ποιά η τιμή του $s[:4:2]$

(α) 'oht'

(β) 'Pt'

(γ) συντακτικό λάθος

6/15

Αν $L1 = [1, 2, 3, \text{'Hello'}, (7,3), 6]$ και $L2 = L1[::-1]$ τότε

ποιά η τιμή του $L1[1] + L2[0]$

(α) 7 (β) 8 (γ) συντακτικό λάθος

ποιά η τιμή του $L1[:2] + L2[:2]$

(α) $[1, 2, 6, (7, 3)]$ (β) $[1, 2, 6]$ (γ) συντακτικό λάθος

7/15

Έστω το λεξικό `d = {'Name' : 'John', 'Age':21, 'Height':1.80}`

Ποιο από τα δυο κομμάτια κώδικα εκτυπώνει *τα ονόματα των κλειδιών* του λεξικού;

(α) `for x in d: print(d[x])`

(β) `for x in d: print(x)`

με το for διατρέχουμε τα κλειδιά ενός λεξικού

Τι εκτυπώνει αυτό κομμάτι του κώδικα `for x in d: d[x] = x
print(d)`

(α) `{'Name': 'Name', 'Age': 'Age', 'Height': 'Height'}`

(β) `{'John': 'John', '21': 21, '1,80': 1,80 }`

(γ) συντακτικό λάθος

8/15

Έστω η συνάρτηση

```
def g(x=4):  
    if bool(x):  
        return 5 * x
```

Τι θα εκτυπώσει καθένα από αυτά τα print;

`print(g(3))` _____ 15

`print(f'{g(3*4/2)}')` _____ 30

`print(g(0))` _____ None

`print(g('z'))` _____ ZZZZZ

`print(g())` _____ 20

Η συνάρτηση `bool(x)` εφαρμόζεται σε οποιοδήποτε αντικείμενο `x` της Python και επιστρέφει την τιμή `False` όταν:

- το `x` είναι *κενός επαναλήψιμος τύπος* π.χ. `[]`, `()`, `'`, `{ }`, `set()`, `range(0)`
- το `x` είναι το *μηδέν* σε οποιαδήποτε μορφή όπως: `0` `0.0` `0.0E+3` `0/17`
- το `x` είναι ένα από τα δυο κυριολεκτήματα (*literals*) `False` ή `None`

Σε κάθε άλλη περίπτωση η συνάρτηση `bool(x)` επιστρέφει την τιμή `True`

Γενικά μπορούμε να πούμε ότι η `bool(x)` επιστρέφει `True` όταν το αντικείμενο `x` είναι "*μη τετριμμένο*" και `False` στην αντίθετη περίπτωση

9/15

Τι θα συμβεί αν προσπαθήσουμε να ανοίξουμε ένα αρχείο κειμένου που ήδη υπάρχει με την εντολή `f = open("Existing_file.txt", "w+")`

Το αρχείο θα ανοίξει κανονικά, *αλλά το προηγούμενο περιεχόμενό του θα χαθεί*

Τι θα συμβεί αν προσπαθήσουμε να ανοίξουμε ένα αρχείο κειμένου που ήδη υπάρχει με την εντολή `f = open("Existing_file.txt", "x+")`

Το αρχείο δεν θα ανοίξει (*FileExistsError: File exists: 'Existing_file.txt'*)

Έστω ότι έχουμε ανοίξει ένα αρχείο κειμένου με `f = open("Some_file.txt")`

Υπάρχει τρόπος να μάθουμε το *μέγεθος του αρχείου σε bytes*, χωρίς να διαβάσουμε το περιεχόμενό του (και χωρίς να χρησιμοποιήσουμε εντολές από άλλα δομοστοιχεία);

```
f.seek(0,2) #Μετάβαση στο τελευταίο byte του αρχείου κειμένου
megethos_se_bytes = f.tell()
```

10/15

Αν `L = [x for x in range(1,12) if x%2 != 0]` τότε

(α) τι τιμή έχει το `L[3]` 7

```
L = [1, 3, 5, 7, 9, 11]
```

(β) τι τιμή έχει το `L[-3]` 7

Τι τιμή έχουν οι λογικές εκφράσεις

```
64 in {x*x for x in range(1,8)}      False
```

```
6 in {i : i**2 + 1 for i in range(1,7)}      True
```

```
6 in {i : i**2 + 1 for i in range(1,7)}.values()      False
```

11/15

Αν $L = [(i, j) \text{ for } i \text{ in range}(1,4) \text{ for } j \text{ in range}(1,4) \text{ if } i==j]$ τότε

(α) τι τιμή έχει το $L[0]$ $(1, 1)$

Απλό στοιχείο

$L = [(1, 1), (2, 2), (3, 3)]$

(β) τι τιμή έχει το $L[:1]$ $[(1, 1)]$

Λίστα ενός στοιχείου

Με ποια συμπερίληψη μπορούμε εναλλακτικά να παράγουμε την επόμενη λίστα;

$L = [(0, 0, 0), (1, 1, 1), (2, 4, 8), (3, 9, 27), (4, 16, 64)]$

α) $L = [(i, i**2, i**3) \text{ for } i \text{ in range}(5)]$

β) $L = [(i, j**2, k**3) \text{ for } i, j, k \text{ in range}(4)]$

γ) $L = [(i, i**2, i**3) \text{ for } i \text{ in range}(4)]$


δ) $L = [(i, j**2, k**3) \text{ for } i, j, k \text{ in range}(5)]$

12/15 (Σωστό / Λάθος)

- ✓ Η κανονική έκφραση `'12?3'` ταιριάζει (εντοπίζει) την συμβολοσειρά `'13'`
- ✗ Η κανονική έκφραση `'^be'` ταιριάζει την συμβολοσειρά `'rebel'`
- ✓ Η κανονική έκφραση `'ab{1,3}c'` ταιριάζει την συμβολοσειρά `'abbbc'`
- ✗ Η κανονική έκφραση `'a.le'` ταιριάζει την συμβολοσειρά `'apple'`
- ✗ Η κανονική έκφραση `'a.?le'` ταιριάζει την συμβολοσειρά `'apple'`
- ✓ Η κανονική έκφραση `'a.?le'` ταιριάζει την συμβολοσειρά `'ale'`
- ✓ Η κανονική έκφραση `'a.*le'` ταιριάζει την συμβολοσειρά `'ale'`

13/15

```
L1 = re.findall( '<.*>', '<H1> title </H1> <H1> subtitle </H1>' )  
L2 = re.findall( '<.*?>', '<H1> title </H1> <H1> subtitle </H1>' )
```



(α) Τι θα εμφανίσει το `print(L1)`

`['<H1> title </H1> <H1> subtitle </H1>']` λίστα με ένα στοιχείο

Greedy matching: ταιριάζει από το πρώτο "<" μέχρι το πιο μακρινό ">"

(β) Τι θα εμφανίσει το `print(L2)`

`['<H1>', '</H1>', '<H1>', '</H1>']` λίστα με 4 στοιχεία

Non greedy matching: ταιριάζει από το πρώτο "<" μέχρι το πλησιέστερο ">", όσες φορές μπορέσει

14/15

```
Av p1 = ("Alice", "Bob", "Chris")
    p2 = ("Artist", "Bartender", "Captain", "Dancer")
    p3 = ["Athens", "Barcelona", "Cairo"]
    z = zip(p1, p2, p3)
    e = enum(p3)
```

(α) Τι θα εμφανίσει το `print(z)`;

```
<zip object at 0x000000000A16E580>
```

(β) Τι θα εμφανίσει το `print(tuple(z))`;

```
(( 'Alice', 'Artist', 'Athens' ), ( 'Bob', 'Bartender', 'Barcelona' ),
 ( 'Chris', 'Captain', 'Cairo' ))
```

(γ) Τι θα εμφανίσει το `print(list(e))`;

```
[(0, 'Athens'), (1, 'Barcelona'), (2, 'Cairo')]
```

15/15 (Σωστό / Λάθος)

Αφαίρεση (abstraction) Ενθυλάκωση (encapsulation)
Κληρονομικότητα και σύνθεση (inheritance and composition)
Πολυμορφισμός (polymorphism)

- ✓ Ειδικό γνώρισμα της αντικειμενοστρέφειας αποτελεί η ενθυλάκωση (encapsulation)
- ✗ Τα δυο βασικά δομικά στοιχεία των κλάσεων είναι τα χαρακτηριστικά και οι ιδιότητες
 - 1. Ιδιότητες ή χαρακτηριστικά (attributes) 2. Μέθοδοι χειρισμού (methods):
- ✓ Οι ειδικές ιδιότητες δεδομένων στιγμιότυπου (data attributes) μιας κλάσης ορίζονται μέσα στην ειδική συνάρτηση `__init__`
- ✗ Οι ιδιωτικές (τοπικές) συναρτήσεις μιας κλάσης ξεκινάνε και τερματίζουν με διπλή κάτω παύλα `__`
 - Ξεκινάνε, αλλά δεν τερματίζουν με διπλή κάτω παύλα
- ✗ Οι θυγατρικές κλάσεις ονομάζονται αλλιώς και *υπερκλάσεις (superclasses)*

Γονεϊκή κλάση ή υπερκλάση (parent class): η κλάση της οποίας τα χαρακτηριστικά κληρονομούνται
Θυγατρική κλάση ή υποκλάση (child class): η κλάση η οποία κληρονομεί τα χαρακτηριστικά

Καλή επιτυχία στο διαγώνισμα!

