

Η Απάντηση

Φύλλο Εργασίας

Στο βιβλίο *The Hitchhiker's Guide to the Galaxy*, περιγράφεται μια υπερευφυής φυλή η οποία αποφασίζει να φτιάξει έναν υπολογιστή που θα δώσει οριστικά την απάντηση για το νόημα της ζωής. Ο υπολογιστής χρειάστηκε 7.5 εκατομμύρια χρόνια για να υπολογίσει και να ελέγξει την Απάντηση για τη Ζωή, το Σύμπαν και τα Πάντα. Η Απάντηση ήταν... σαράντα δύο.

Ο υπολογιστής λέγονταν Deep Thought και μπορούμε να φτιάξουμε κι εμείς ένα πρόγραμμα σαν το δικό του. Επειδή γνωρίζουμε ήδη την Απάντηση, θα κλέψουμε λίγο: το πρόγραμμά μας δεν θα υπολογίζει την Απάντηση, αλλά μόνο θα την ανακοινώνει στο χρήστη.

Έννοιες: είσοδος, έξοδος, μεταβλητές, δομή επιλογής.

Με αυτό το φύλλο εργασίας θα γνωρίσουμε τα βασικά είδη εντολών που θα επιτρέπουν στα προγράμματά μας να *αλληλεπιδρούν* με το χρήστη, δηλαδή να του εμφανίζουν μηνύματα στην οθόνη και να του ζητούν να εισάγει τιμές από το πληκτρολόγιο. Θα εξετάσουμε επίσης πως μπορούμε να κάνουμε τα προγράμματά μας να *επιλέγουν* τον τρόπο που θα συμπεριφερθούν, εξετάζοντας τις *συνθήκες* που επικρατούν κατά την εκτέλεσή τους.

Μηνύματα

1. Πληκτρολογήστε την εντολή:

```
print("Καλημέρα.")
```

Αυτό είναι το πρώτο σας πρόγραμμα. Εκτελέστε το για να δείτε τι θα συμβεί.

2. Βασιστείτε στην εντολή του προηγούμενου βήματος και *προσθέστε* στο πρόγραμμά σας μια ακόμα εντολή, έτσι ώστε να εμφανίζεται στην οθόνη η Απάντηση.

Η Απάντηση είναι... 42

Εκτελέστε ξανά το πρόγραμμα. Τα καταφέρατε;

.....

1

22 Ιουλίου 2016

13:14

Διαβάστε το αντίστοιχο κεφάλαιο:
[pythonies.mysch.gr/chapters/
answer.pdf](http://pythonies.mysch.gr/chapters/answer.pdf)

Στη διεύθυνση pythonies.mysch.gr μπορείτε να βρείτε κι άλλα φύλλα εργασίας σαν αυτό, καθώς και το συνοδευτικό τους υλικό.



3. Προσθέστε πριν την `print` του προηγούμενου βήματος την εντολή:

```
answer = 42
```

Η `answer` είναι μια μεταβλητή στην οποία δίνουμε την τιμή 42. Τώρα μπορούμε να αναφερόμαστε στην Απάντηση, χωρίς να έχει σημασία ποια είναι αυτή.

Τροποποιήστε την `print` του βήματος 2 ως εξής:

```
print("Η Απάντηση είναι...", answer)
```

Εκτελέστε το πρόγραμμα. Παρατηρείτε κάποια διαφορά στα μηνύματα που εμφανίζονται μετά από τις τροποποιήσεις που κάνατε;

.....

Σε τι πιστεύετε ότι διαφέρει το πρόγραμμα μετά από αυτές τις τροποποιήσεις;

.....

.....

Πιστεύετε ότι θα λειτουργούσε το πρόγραμμα αν είχατε τοποθετήσει την εντολή `answer = 42` μετά τις `print`; Δικαιολογήστε την απάντησή σας.

.....

.....

4. Τροποποιήστε την εντολή που δίνει τιμή στην `answer`:

```
answer = 3 + 13 * 3
```

Ποιά πιστεύετε ότι θα είναι τώρα η τιμή της `answer`;

.....

Εκτελέστε το πρόγραμμα. Ποιά είναι η τιμή της `answer`;

.....

Η τιμή της `answer` προέκυψε τώρα από τον υπολογισμό της τιμής μιας έκφρασης. Προσπαθήστε να γράψετε άλλες δύο παρόμοιες αριθμητικές εκφράσεις που χρησιμοποιούν διαφορετικά νούμερα και διαφορετικές πράξεις αλλά δίνουν την ίδια τιμή στην `answer`.

Προσπαθήστε να χρησιμοποιήσετε και τους τελεστές `//` και `%` για το ηλίκιο και το υπόλοιπο της ακέραιας διαίρεσης.

.....

.....



Το σύμβολο * αντιστοιχεί στην πράξη του πολλαπλασιασμού. Μπορείτε επίσης να χρησιμοποιήσετε τα κλασικά + και -, καθώς επίσης και το / για τη διαίρεση, το // για το ηλίκιο της ακέραιας διαίρεσης και το % για το υπόλοιπο της ακέραιας διαίρεσης.



Ερωταποκρίσεις

Θα προγραμματίσουμε τον Deep Thought έτσι ώστε να ζητάει το όνομα του χρήστη και να τον καλημερίζει κατάλληλα. Έτσι θα υπάρξει ένας στοιχειώδης διάλογος, πριν ανακοινωθεί η Απάντηση.

5. Στην αρχή του προγράμματος προσθέστε τις παρακάτω εντολές:

```
print("Πώς σε λένε;")
name = input()
```

Η `input()` επιστρέφει το κείμενο που πληκτρολόγησε ο χρήστης, επιστρέφει δηλαδή μια *αλφαριθμητική* τιμή. Εδώ χρησιμοποιούμε την `input()` για να διαβάσουμε την απάντηση του χρήστη, η οποία αποθηκεύεται στη μεταβλητή `name`.

6. Συμπληρώστε την εντολή `print("Καλημέρα.")` έτσι ώστε το πρόγραμμα να χαιρετά τον χρήστη χρησιμοποιώντας το όνομά του, το οποίο είναι αποθηκευμένο στη μεταβλητή `name`. Για παράδειγμα:

Καλημέρα Μαρία

Αν δυσκολευτείτε, ανατρέξτε στο βήμα 3, όπου γίνεται κάτι ανάλογο με τη μεταβλητή `answer`.

Εκτελέστε το πρόγραμμα σας 2-3 φορές και, *παίζοντας το ρόλο του χρήστη*, πληκτρολογήστε κάθε φορά ένα διαφορετικό όνομα. Λειτουργεί το πρόγραμμα όπως θα περιμένατε;

.....

Ποιά θα ήταν η διαφορά αν επιχειρούσατε να καλημερίσετε τον χρήστη με την εντολή `print("Καλημέρα name");`

.....

Ποιά θα ήταν η διαφορά αν, στο βήμα 5, χρησιμοποιούσαμε την εντολή `name = "Μαρία"`, αντί για την εντολή `name = input();`

.....

Ποιά θα ήταν η διαφορά αν, στο βήμα 3, χρησιμοποιούσαμε την εντολή `answer = input()` αντί για την εντολή `answer = 42;`

.....

.....



Βιβλιοθήκες

Για να προσθέσουμε λίγο σασπένς, θα θέλαμε να υπάρχει μια *καθυστέρηση* πριν από την ανακοίνωση της Απάντησης.

Επειδή στις βασικές εντολές της Python δεν συγκαταλέγεται κάποια εντολή καθυστέρησης, θα χρησιμοποιήσουμε τη *βιβλιοθήκη time*, η οποία παρέχει τη λειτουργικότητα που μας χρειάζεται.

7. Προσθέστε στην αρχή του προγράμματος την εντολή που ακολουθεί, για να εισάγετε στο πρόγραμμα τη βιβλιοθήκη `time`:

```
import time
```

Αμέσως πριν από την `print` που ανακοινώνει την Απάντηση στο χρήστη, προσθέστε τη γραμμή που ακολουθεί:

```
time.sleep(3)
```

Εδώ χρησιμοποιείται η *συνάρτηση sleep*, από τη βιβλιοθήκη `time`. Εκτελέστε το πρόγραμμα. Τί αποτέλεσμα έχει η προσθήκη αυτής της εντολής;

.....

Διερευνήστε τί θα συμβεί αν χρησιμοποιήσουμε άλλη τιμή, διαφορετική από το 3, ως *παράμετρο* της `sleep`. Τί ρόλο πιστεύετε ότι παίζει η παράμετρος της `sleep`;

.....

Μπορείτε να αναφέρετε ακόμα ένα ή δύο παραδείγματα εφαρμογών όπου θα χρησιμοποιούσατε την `sleep`;

.....

8. Πριν από το σημείο καθυστέρησης του προηγούμενου βήματος, προσθέστε μια εντολή η οποία υπολογίζει πόσα δευτερόλεπτα αντιστοιχούν σε 7.5 εκατομμύρια χρόνια αναμονής και αποδίδει αυτή την τιμή σε μια νέα μεταβλητή `wait`.

Αν δυσκολευτείτε, μπορείτε να ανατρέξετε στο βήμα 4, όπου υπολογίζεται με ανάλογο τρόπο η τιμή της μεταβλητής `answer`.

9. Τροποποιήστε προσωρινά τη γραμμή όπου χρησιμοποιείται η `sleep` έτσι ώστε, αντί για 3 δευτερόλεπτα, η καθυστέρηση να διαρκεί `wait` δευτερόλεπτα.

Σε ορισμένα περιβάλλοντα προκαλείται *σφάλμα* με μια τόσο μεγάλη τιμή καθυστέρησης. Στην περίπτωση αυτή, δώστε τιμή στη μεταβλητή `wait` που αντιστοιχεί σε μικρότερη καθυστέρηση, π.χ. έναν χρόνο.

Οι βιβλιοθήκες είναι συλλογές από έτοιμα μικρά προγράμματα που μπορούμε να χρησιμοποιήσουμε στα προγράμματά μας.



Αν δεν θέλετε να περιμένετε, μπορείτε να διακόψετε την εκτέλεση του προγράμματός σας με τον συνδυασμό πλήκτρων `Ctrl + C`.

Επιλογές: Τί Ώρα Είναι;

Ο Deep Thought, που γνωρίζει την Απάντηση για τη Ζωή, το Σύμπαν και τα Πάντα, δεν θα έπρεπε να καλημερίζει τον χρήστη ακόμα κι όταν είναι βράδι... Θα θέλαμε το πρόγραμμα να είναι περισσότερο ευέλικτο και να προσαρμόζει τον χαιρετισμό του ανάλογα με την ώρα της ημέρας.

10. Πριν από το σημείο όπου το πρόγραμμά σας καλημερίζει το χρήστη, προσθέστε την εντολή:

```
hour = time.localtime().tm_hour
```

Εδώ χρησιμοποιήσαμε και πάλι τη βιβλιοθήκη `time`. Οι συντακτικές λεπτομέρειες δεν έχουν σημασία, αυτό που μας νοιάζει είναι πως η τιμή της μεταβλητής `hour` είναι η τρέχουσα ώρα του συστήματος: ένας ακέραιος από το 0 μέχρι και το 23. Αν θέλετε να το επιβεβαιώσετε, μπορείτε να προσθέσετε προσωρινά μια `print` που εμφανίζει την τιμή της `hour` στην οθόνη.

11. Αμέσως μετά την `print("Καλημέρα", name)` με την οποία το πρόγραμμα καλημερίζει το χρήστη, προσθέστε την εντολή:

```
print("Καλησπέρα", name)
```

Εκτελέστε το πρόγραμμα. Εμφανίζονται και τα δύο μηνύματα;

.....

Εμείς θέλουμε να εκτελείται μόνο η μία από τις δύο εντολές, ανάλογα με την ώρα της ημέρας, δηλαδή την τιμή της μεταβλητής `hour`. Θα πρέπει λοιπόν να προγραμματίσουμε τον Deep Thought έτσι ώστε να ελέγχει την `hour` και να εμφανίζει διαφορετικό μήνυμα ανάλογα με το αποτέλεσμα του ελέγχου.

12. Τροποποιήστε το σημείο όπου το πρόγραμμα καλημερίζει το χρήστη:

```
if hour < 16:
    print("Καλημέρα", name)
else:
    print("Καλησπέρα", name)
```

Δοκιμάστε να εκτελέσετε το πρόγραμμά σας. Δεν πρόκειται να λειτουργήσει, θα εμφανιστεί ένα μήνυμα σφάλματος.

`IndentationError: expected an indented block`

Στις περισσότερες γλώσσες προγραμματισμού δεν θα αντιμετωπίζατε κάποιο πρόβλημα, όμως η Python έχει μια "ευαισθησία": Οι εντολές που εκτελούνται στη μία ή στην άλλη περίπτωση πρέπει να ξεχωρίζουν, πρέπει με κάποιον τρόπο να επισημανθεί ότι οι εντολές αυτές ανήκουν αντίστοιχα στην `if` και την `else`.

Στην Python, αυτό επιτυγχάνεται με τις εσοχές.



Για να συγκριθούν τιμές μεταξύ τους χρησιμοποιούμε τα `<` (μικρότερο), `<=` (μικρότερο ή ίσο), `>` (μεγαλύτερο) και `>=` (μεγαλύτερο ή ίσο). Επίσης, με τα `==` (ίσο) και το `!=` (διάφορο) ελέγχεται αν δύο τιμές είναι ίσες ή διαφορετικές.



13. Προσθέστε 4 κενά πριν τις δύο `print` κι εκτελέστε το πρόγραμμα.

```
if hour < 16:
    print("Καλημέρα", name)
else:
    print("Καλησπέρα", name)
```

Θυμηθείτε ότι η `hour` αντιστοιχεί στην ώρα της ημέρας και η τιμή της κυμαίνεται από το 0 μέχρι και το 23. Για ποιο διάστημα τιμών της `hour` θα εμφανιστεί το "Καλημέρα" και για ποιες το "Καλησπέρα";

hour	μήνυμα
.....	"Καλημέρα"
.....	"Καλησπέρα"



Εκτελέστε το πρόγραμμα. Συμπληρώστε παρακάτω την ώρα που το εκτελέσατε και ποιο από τα δύο μηνύματα εμφανίστηκε στην οθόνη.

ώρα εκτέλεσης	μήνυμα
0 - 23	"Καλημέρα" / "Καλησπέρα"
.....



Τι ώρα θα έπρεπε να εκτελέσετε το πρόγραμμα για να εμφανιστεί το αντίθετο μήνυμα; Υπάρχουν πολλές εναλλακτικές, συμπληρώστε μία από αυτές.

ώρα εκτέλεσης	μήνυμα
0 - 23	"Καλημέρα" / "Καλησπέρα"
.....



Για να δείτε το άλλο μήνυμα να εμφανίζεται και να διαπιστώσετε ότι το πρόγραμμα πράγματι "προσαρμόζεται" ανάλογα με την ώρα, θα πρέπει να αλλάξετε την ώρα! Τροποποιήστε λοιπόν προσωρινά την εντολή του βήματος 10 που δίνει τιμή στη μεταβλητή `hour`. Ορίστε την τιμή της `hour` να είναι ίση με την ώρα εκτέλεσης που συμπληρώσατε στην προηγούμενη ερώτηση.

Εμφανίστηκε το σωστό μήνυμα και σ' αυτή την περίπτωση;

.....



Κι Άλλες Επιλογές: Η Ηλικία Μετράει

14. Πριν την ανακοίνωση της Απάντησης, προσθέστε τις κατάλληλες εντολές στο πρόγραμμα έτσι ώστε να ζητά από το χρήστη να πληκτρολογήσει το έτος γέννησής του και να το αποθηκεύει σε μια μεταβλητή με όνομα `birth`. Για παράδειγμα:

Ποιό έτος γεννήθηκαν;
2001



Αν δυσκολευτείτε, μπορείτε να ανατρέξετε στο βήμα 5, όπου γίνεται κάτι ανάλογο για το όνομα του χρήστη.

15. Προσθέστε στο πρόγραμμά σας την παρακάτω εντολή, με την οποία η τιμή της μεταβλητής `year` γίνεται ίση με το τρέχον έτος.

```
year = time.localtime().tm_year
```

16. Γνωρίζοντας το τρέχον έτος `year` και το έτος γέννησης του χρήστη `birth`, προσπαθήστε να υπολογίσετε την ηλικία του χρήστη και να αποθηκεύσετε την τιμή σε μια μεταβλητή `age`. Ανατρέξτε στα βήματα 4 και 8 για παραδείγματα όπου γίνεται κάτι ανάλογο.

Αν επιχειρήσετε να εκτελέσετε το πρόγραμμα θα διαπιστώσετε ότι δεν λειτουργεί, ακόμα κι αν ο υπολογισμός σας είναι σωστός. Θα δείτε να εμφανίζεται ένα (μάλλον ακατάληπτο) μήνυμα λάθους.



```
TypeError: unsupported operand type(s) for -:  
'int' and 'str'
```

Το μήνυμα εξηγεί ότι δεν είναι δυνατές οι πράξεις ανάμεσα σε αριθμούς και *αλφαριθμητικές τιμές* (κείμενα). Ίσως αναρωτιέστε που βρέθηκε η αλφαριθμητική τιμή. Η `input()`, που χρησιμοποιήσατε για να διαβάσετε το έτος γέννησης που πληκτρολογεί ο χρήστης, επιστρέφει την απάντηση του χρήστη σε μορφή κειμένου, κι έτσι η τιμή της `birth` θα έχει τη μορφή `"2001"` και όχι `2001`.

17. Επιστρέψτε στη γραμμή όπου διαβάζεται το έτος γέννησης του χρήστη και *τροποποιήστε* την ως εξής:

```
birth = int(input())
```

Η `int` μετατρέπει το κείμενο που πληκτρολογεί ο χρήστης σε ακέραιο αριθμό. Εκτελέστε και πάλι το πρόγραμμά σας και επιβεβαιώστε ότι δεν εμφανίζεται πια μήνυμα λάθους.

18. Συμπληρώστε το πρόγραμμά σας, έτσι ώστε να εμφανίζει στο χρήστη την ηλικία του. Για παράδειγμα:

Ποιό έτος γεννήθηκαν;
2001
Είσαι 15 χρονών.



19. Τροποποιήστε το πρόγραμμα και χρησιμοποιήστε την `if`, έτσι ώστε το μήνυμα που εμφανίζεται να εξαρτάται από το αν ο χρήστης έχει ξεπεράσει τα 18 έτη. Για παράδειγμα:

Ποιό έτος γεννήθηκαν;
2001
15 χρονών, η κατάλληλη ηλικία να μάθεις την Απάντηση.



Ποιό έτος γεννήθηκαν;

1976

Πάτησες τα 40, είναι λίγο αργά να μάθεις την Απάντηση.

Εκτελέστε το πρόγραμμα σας δύο φορές. Την πρώτη, *παίζοντας το ρόλο του χρήστη*, δώστε ένα έτος γέννησης που αντιστοιχεί σε ηλικία μικρότερη των 18 ετών, ενώ τη δεύτερη το αντίθετο. Εμφανίζεται το κατάλληλο μήνυμα σε κάθε περίπτωση;

.....



20. *Τροποποιήστε το πρόγραμμα έτσι ώστε η Απάντηση να εμφανίζεται μόνο στην περίπτωση που ο χρήστης δεν έχει ξεπεράσει τα 18 έτη. Δεν χρειάζεται να γράψετε καμία νέα εντολή, μόνο να μετακινήσετε κάποια/κάποιες από τις ήδη υπάρχουσες μέσα στην **if** του προηγούμενου βήματος.*

Αν παρουσιαστεί κάποιο πρόβλημα, τότε θα πρέπει να βεβαιωθείτε ότι έχετε χρησιμοποιήσει σωστά τις εσοχές, μπροστά από τις εντολές που βρίσκονται μέσα στην **if**.

Εκτελέστε το πρόγραμμα σας δύο φορές. Την πρώτη, *παίζοντας το ρόλο του χρήστη*, δώστε ένα έτος γέννησης που αντιστοιχεί σε ηλικία μικρότερη των 18 ετών, ενώ τη δεύτερη το αντίθετο. Εμφανίζεται η Απάντηση *μόνο* στην πρώτη περίπτωση;

.....



Δραστηριότητες για Εξάσκηση

Για περισσότερη εξάσκηση στις έννοιες που γνωρίσατε σ' αυτό το φύλλο εργασίας, μπορείτε ν' ανατρέξετε στις ασκήσεις του Κεφαλαίου "Η Απάντηση".

pythonies.mysch.gr/complete