Η Απάντηση

Ενδεικτικές Απαντήσεις Φύλλου Εργασίας

1

22 Ιουλίου 2016

13:16

Μηνύματα

1. Πληκτρολογήστε την εντολή:

```
print("Καλημέρα.")
```

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

Θα εμφανιστεί στην οθόνη το μήνυμα `Καλημέρα` .

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
```

Η εντολή που θα προσθέσουμε στο πρόγραμμα είναι:

```
print("Η Απάντηση είναι... 42")
```

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.

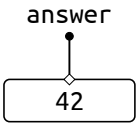
Τροποποιήστε την `print` του βήματος 2 ως εξής:

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

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

Δεν υπάρχει διαφορά στα αποτελέσματα του προγράμματος. Το `answer` *είναι το όνομα μιας τιμής. Κατά την εκτέλεση της* `print`*, στη θέση του ονόματος* `answer` *εμφανίζεται η τιμή του που του έχουμε αναθέσει, δηλαδή το* `42`*.*

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



Η μεταβλητή `answer` έχει την τιμή `42`. Εναλλακτικά, θα λέγαμε ότι στην τιμή `42` έχει δοθεί το όνομα `answer`.

Η πρώτη `print` *εμφανίζει πάντα ακριβώς το ίδιο, σταθερό μήνυμα. Η δεύτερη* `print` *εμφανίζει την τιμή της* `answer`*, η οποία εδώ είναι* `42`*, αλλά θα μπορούσε να είναι διαφορετική.*

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

Όχι, δεν θα λειτουργούσε αφού κατά την εκτέλεση της `print` *το όνομα* `answer` *δεν θα είχε ηδη αντιστοιχισθεί σε κάποια τιμή, επομένως θα εμφανίζονταν μήνυμα σφάλματος αν το χρησιμοποιούσαμε.*

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

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

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

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

   Η τιμή της `answer` *θα είναι* `42`*. Η ιεραρχία των πράξεων είναι ίδια με τα μαθηματικά, επομένως θα προηγηθεί η πράξη του πολλαπασιασμού και στη συνέχεια θα εκτελεστεί η πράξη της πρόσθεσης.*

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

   Η τιμή της `answer` *είναι* `42`*, η οποία εμφανίζεται στην οθόνη.*

   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 πηλίκο και το υπόλοιπο της ακέραιας διαίρεσης.

   Δύο παραδείγματα υπολογισμού είναι:

   i. `answer = 100 // 2 - 8`

   ii. `answer = 40 + 6 % 4`

Ερωταποκρίσεις

Θα προγραμματίσουμε τον Deep Thought έτσι ώστε να ζητάει το όνομα του χρήστη και να τον καλημερίζει κατάλληλα. Έτσι θα υπάρχει 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`.

   Η εντολή που θα προσθέσουμε στο πρόγραμμα είναι:

   ```
   print("Καλημέρα", name)
   ```

   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;

   Ναι, το πρόγραμμα χαιρετά τον χρήστη χρησιμοποιώντας το όνομα που πληκτρολόγησε.

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

   Θα εμφανίζονταν στην οθόνη το μήνυμα `Καλημέρα name`. *Οτιδήποτε βρίσκεται μέσα σε εισαγωγικά αποτελεί μια αλφαριθμητική τιμή, δηλαδή κείμενο και εμφανίζεται ως έχει, χωρίς να αναφέρεται σε ονόματα μεταβλητών.*

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

   Η μεταβλητή `name` *θα ήταν καθορισμένη από τον προγραμματιστή και όχι από τον χρήστη και θα είχε σε κάθε εκτέλεση του προγράμματος την τιμή* `"Μαρία"`.

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

   Ο χρήστης θα καθόριζε την τιμή της μεταβλητής `answer`, *ανάλογα με το τι θα πληκτρολογούσε σε κάθε εκτέλεση του προγράμματος Η μεταβλητή* `answer` *δεν θα είχε απαραίτητα την τιμή* `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.

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

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;

Εισάγεται μια καθυστέρηση ανάμεσα στην εμφάνιση των δύο μηνυμάτων.

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`;

Η παράμετρος `3` *καθορίζει ότι θα υπάρχει μια καθυστέρηση τριών δευτερολέπτων. Αν χρησιμοποιήσουμε άλλον αριθμό στη θέση του* `3` *θα αλλάξει η περίοδος της χρονικής καθυστέρησης.*

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

Η `sleep` *μπορεί να χρησιμοποιηθεί σε οποιοδήποτε πρόγραμμα χρειάζεται να εισάγουμε καθυστέρηση μεταξύ της εκτέλεσης κάποιων εντολών. Ένα απλό παράδειγμα είναι η υλοποίηση ενός φωτεινού σηματοδότη που θα έδειχνε τα χρώματα κόκκινο, πορτοκαλί, πράσινο αφήνοντας ένα χρονικό διάστημα για να μεταβεί από το ένα στο άλλο ή ένα πρόγραμμα που θα προσομοίωνε τη λειτουργία ενός ανελκυστήρα που θα χρειαζόταν μια χρονική καθυστέρηση για να μεταβεί από τον έναν όροφο στον άλλον.*

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`.

Ο χρόνος αναμονής προκύπτει υπολογίζοντας (περίπου) τα δευτερόλεπτα που αντιστοιχούν σε 7.5 εκατομμύρια χρόνια.

```
wait = 7500000 * 365 * 24 * 60 * 60
```

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.

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

Ο υπολογισμός των δευτερολέπτων που αντιστοιχεί σε ένα έτος μπορεί να γίνει παραλείποντας από την παραπάνω αριθμητική έκφραση τον αριθμό `7500000` *όπως παρακάτω:*

```
wait = 365 * 24 * 60 * 60
```

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;

Ναι, εμφανίζονται και τα δύο μηνύματα, αφού σε κάθε περίπτωση εκτελούνται και οι δύο `print()`.

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)
```

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.

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*.

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"`;

Όταν η τιμή της μεταβλητής `hour` *είναι μικρότερη του* `16` *θα εμφανιστεί το μήνυμα* `Καλημέρα` *, ενώ όταν είναι από* `16` *μεχρι και* `23` *θα εμφανιστεί το μήνυμα* `Καλησπέρα` *.*

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

Έστω ότι εκτελούμε το πρόγραμμα στις `10` *το πρωί. Η τιμή της hour θα είναι μικρότερη του* `16`*, οπότε θα εμφανιστεί το μήνυμα* `Καλημέρα` *.*

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

Αν εκτελέσουμε το πρόγραμμα στις `9` *το βράδυ (*`21`*) η συνθήκη της* **`if`** *δεν θα ισχύει, επομένως θα εκτελεστούν οι εντολές της* **`else`** *και θα εμφανιστεί το μήνυμα* `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;

Αν αντικαταστήσουμε την εντολή που αναθέτει τιμή στην `hour` *με την παρακάτω:*

```
hour = 21
```

και εκτελέσουμε το πρόγραμμα, τότε εμφανίζεται στην οθόνη το μήνυμα `Καλησπέρα` *.*

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 γεννήθηκες;
2001
```

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

Οι εντολές που θα προσθέσουμε στο πρόγραμμα είναι:

```
print("Ποιο έτος γεννήθηκες;")
birth = input()
```

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.

Η ηλικία του χρήστη θα υπολογιστεί αφαιρώντας από το τρέχον έτος το έτος γέννησής του. Η εντολή που θα προσθέσουμε στο πρόγραμμα είναι:

```
age = year - birth
```

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 τιμες* (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 γεννήθηκες;
2001
Είσαι 15 χρονών.
```

Η εντολή που θα προσθέσουμε στο πρόγραμμα είναι:

```
print("Είσαι", age, "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 γεννήθηκες;
2001
15 χρονών, η κατάλληλη ηλικία να μάθεις την Απάντηση.

Ποιό έτος γεννήθηκες;
1976
Πάτησες τα 40, είναι λίγο αργά να μάθεις την Απάντηση.
```

Εφόσον θέλουμε το πρόγραμμά μας να εκτελεί διαφορετικές εντολές ανάλογα με το αν η ηλικία είναι μικρότερη του 18 ή όχι θα χρησιμοποιήσουμε την **if–else**, *ώστε το πρόγραμμα να εμφανίζει διαφορετικά μηνύματα σε κάθε περίπτωση όπως παρακάτω:*

```
if age < 18:
    print(age, "χρονών, η κατάλληλη ηλικία...")
else:
    print("Πάτησες τα", age, ", είναι λίγο αργά...")
```

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;

Δεδομένου ότι το τρέχον έτος είναι `2016`*, δίνοντας τον αριθμό* `2001` *ως έτος γέννησης εμφανίζεται το μήνυμα:*

```
15 χρονών, η κατάλληλη ηλικία...
```

ενώ δίνοντας τον αριθμό `1978` *εμφανίζεται το μήνυμα:*

```
Πάτησες τα 38, είναι λίγο αργά ...
```

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.

Θα μετακινήσουμε την εμφάνιση της απάντησης "μέσα" στην **if**, *όπως φαίνεται παρακάτω:*

```
if age < 18:
    print(age, "χρονών, η κατάλληλη ηλικία...")
    print("Η Απάντηση είναι... ", answer)
else:
    print("Πάτησες τα", age, ", είναι λίγο αργά...")
```

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