

Το παιχνίδι Dragon Realm



Πλησιάζετε σε ένα μυστηριώδες νησί. Εδώ κατοικούν πολλοί δράκοι. Άλλοι από αυτούς είναι φιλικοί και ήσυχοι, ενώ άλλοι πεινασμένοι και άγριοι. Κάθε δράκος κατοικεί σε μια σπηλιά και έχει κρυμμένους θησαυρούς. Οι φιλικοί δράκοι θέλουν να μοιραστούν μαζί σας τους θησαυρούς τους. Οι άγριοι δράκοι θα σας κατασπαράξουν μόλις τους πλησιάσετε.

Θέλουμε να δημιουργήσουμε ένα πρόγραμμα, όπου θα υποθέτουμε ότι ο παίκτης βρίσκεται μπροστά σε δύο σπηλιές. Στη μια ζει ένας φιλικός δράκος και στην άλλη ένας πεινασμένος κι άγριος.

Ο **παίκτης** καλείται να διαλέξει ποια σπηλιά θα πλησιάσει (πληκτρολογώντας τον αριθμό 1 ή 2).

Το **πρόγραμμα** επιλέγει με τυχαίο τρόπο την σπηλιά του φιλικού δράκου.

Η **επιλογή του παίκτη συγκρίνεται με την επιλογή του προγράμματος** και εμφανίζεται κατάλληλο μήνυμα (αν ο παίκτης πήρε το θησαυρό ή αν τον έφαγε ο δράκος).

Ο παίκτης μπορεί να παίξει όσες φορές θέλει. Δηλ. **το πρόγραμμα θα τον ρωτάει στο τέλος κάθε γύρου αν θέλει να ξαναπαίξει**.

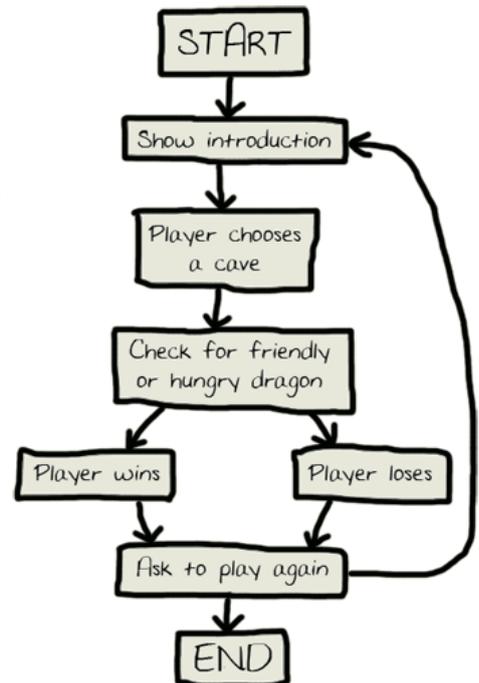


Το **διάγραμμα του παιχνιδιού** θα μπορούσε να είναι κάπως έτσι. Με βάση αυτό, δοκιμάστε να φτιάξετε έναν δικό σας κώδικα.

Στο τμήμα **Show introduction**, θα δώσετε με πολλές εντολές `print()`, όλες τις απαραίτητες οδηγίες στον παίκτη.

Στο τμήμα **Player chooses a cave**, θα πάρετε με εντολή `input()` την επιλογή του παίκτη (1 ή 2) και θα πάρετε με εντολή `randint(1,2)` την επιλογή του υπολογιστή (για το ποιος είναι ο φιλικός δράκος).

Στο τμήμα **Check for friendly or hungry dragon**, θα συγκρίνετε την επιλογή του παίκτη και την επιλογή του υπολογιστή και θα εμφανίσετε το κατάλληλο μήνυμα.



Dragon Realm με Συναρτήσεις (functions) που ορίζονται από εμάς

Όταν έχουμε ένα μεγάλο πρόγραμμα, μπορούμε κάποια τμήματα που επιτελούν **αυτόνομη εργασία**, να τα γράψουμε ως **ξεχωριστά κομμάτια προγράμματος**.

Κάθε τέτοιο κομμάτι προγράμματος ονομάζεται **function (συνάρτηση)** και πρέπει να έχει μοναδικό όνομα.



Ας δημιουργήσουμε μια function, την **displayIntro()**, που δουλειά της είναι να διηγείται την ιστορία του παιχνιδιού.

```
# ορισμός της συνάρτησης με όνομα: displayIntro()

# Εμφανίζει την ιστορία του παιχνιδιού

def displayIntro():
    print("You are on a planet full of dragons. In front of you,")
    print("you see two caves. In one cave, the dragon is friendly")
    print("and will share his treasure with you. The other dragon")
    print("is greedy and hungry, and will eat you on sight.")
    print()
```

Παρατήρηση 1: αυτή η συνάρτηση για να κάνει τη δουλειά της δεν χρειάζεται κάποιες τιμές που να έχουν υπολογιστεί πιο πριν. Ούτε παράγει κάποιες τιμές που θα χρειαστούν στη συνέχεια.



Ας δημιουργήσουμε μια δεύτερη function, την **chooseCave()**, που δουλειά της είναι να ζητάει από τον παίκτη να επιλέξει σπηλιά.

Παρατήρηση 2: αυτή η συνάρτηση για να κάνει τη δουλειά της δεν χρειάζεται κάποιες τιμές που να έχουν υπολογιστεί πιο πριν. Όμως παράγει μία τιμή που θα χρειαστεί στη συνέχεια: η σπηλιά που διάλεξε ο παίκτης. Αυτή την τιμή την **επιστρέφει** με την εντολή **return**.

7

```
# ορισμός της συνάρτησης με όνομα: chooseCave()

# Ρωτάει τον παίκτη σε ποια σπηλιά θέλει να μπει (1 ή 2) και
# όσο ο παίκτης απαντάει με άσχετους αριθμούς τον ξαναρωτάει.
# Στο τέλος επιστρέφει τον αριθμό της σπηλιάς που διάλεξε ο παίκτης.

def chooseCave():
    cave = 0
    while cave != 1 and cave != 2:
        print('Which cave will you go into? (1 or 2)')
        cave = input()
        cave = int(cave)
    return cave
```



Ας δημιουργήσουμε μια τρίτη function, την `checkCave()`, που δουλειά της είναι να ζητάει από τον υπολογιστή να επιλέξει με τυχαίο τρόπο ποια είναι η σπηλιά του καλού δράκου και στη συνέχεια να ελέγχει αν ο παίκτης μπήκε σε αυτή ή στην άλλη.

```
import random
import time

# ορισμός της συνάρτησης με όνομα: checkCave()

# Επιλέγει με τυχαίο τρόπο την σπηλιά του καλού δράκου (1 ή 2).
# Στη συνέχεια συγκρίνει την επιλογή του παίκτη και εμφανίζει κατάλληλο μήνυμα .

def checkCave(chosenCave):
    print('You approach the cave...')
    time.sleep(2)
    print('It is dark and spooky...')
    time.sleep(2)
    print('A large dragon jumps out in front of you!')
    print('He opens his jaws and...')
    print()
    time.sleep(2)

    friendlyCave = random.randint(1, 2)

    if chosenCave == friendlyCave:
        print('Gives you his treasure!')
    else:
        print('Gobbles you down in one bite!')
```

7

Παρατήρηση 3: αυτή η συνάρτηση για να κάνει τη δουλειά της χρειάζεται κάποια τιμή που έχει υπολογιστεί πιο πριν: την σπηλιά που επέλεξε ο παίκτης. Γιαυτό και μέσα στην παρένθεση γράφουμε ένα όνομα για αυτή την τιμή – αυτό λέγεται *παράμετρος*.

Η συνάρτηση δεν παράγει κάποιες τιμές που θα χρειαστούν στη συνέχεια, γαυτό και δεν υπάρχει εντολή return.



Οι παραπάνω ορισμοί των συναρτήσεων δεν κάνουν τίποτε από μόνοι τους.

Πρέπει λοιπόν να γράψουμε το **κυρίως πρόγραμμα** (το κανονικό πρόγραμμα) το οποίο θα εκτελεί τις συναρτήσεις όταν χρειάζεται. Και βέβαια, θα κάνει κι άλλα πράγματα! Στην ρυθση το κυρίως πρόγραμμα ξεκινάει άκρη-άκρη, χωρίς εσοχή.

Κυρίως Πρόγραμμα

```
playAgain = 'y'
```

```
while playAgain == 'y':  
    displayIntro()
```

```
    caveNumber = chooseCave()
```

```
    checkCave(caveNumber)
```

```
    print('Do you want to play again? (y or n)')  
    playAgain = input()
```

Δεν υπάρχουν μεταβλητές ούτε για είσοδο ούτε για έξοδο.

Η τιμή που επιστρέφει η chooseCave() με την εντολή return, αποθηκεύεται στη μεταβλητή caveNumber.

Η checkCave() χρειάζεται ως είσοδο τον αριθμό της σπηλιάς που διάλεξε ο παίκτης.

Αυτή η πληροφορία υπάρχει (από ακριβώς πριν) στην μεταβλητή caveNumber.

Για να την δώσουμε στην checkCave(), γράφουμε το όνομα της μεταβλητής μέσα στην παρένθεση της checkCave().

Τρέξτε τον κώδικα και βεβαιωθείτε ότι ο παιχνίδι δουλεύει σωστά.

Μπορείτε να εξηγήσετε πως ακριβώς δουλεύει το πρόγραμμα;

