

Πιστολέρο (Fast Draw)

Ωρα για να δοκιμάσετε τα αντανακλαστικά σας και να δείτε αν είστε ο πιο γρήγορος enter-οπατητής. Οι κανόνες είναι απλοί.

Προγραμματίστε ένα παιχνίδι που:

Όταν δείτε "DRAW", έχετε 0.3 δευτερόλεπτα να πατήσετε Enter.

Αν πατήσετε Enter νωρίτερα, χάνετε.

Χρήσιμες Πληροφορίες

1. `import time → time.sleep(), time.time()`

- `time.sleep()`

Syntax : `sleep(sec)`

Parameter: `sec` : Number of seconds for which the code is required to be stopped.

Return: VOID.

- `time.time()`

Syntax: `time.time()`

Parameter: No parameter is required

Return type: This method returns a float value which represents the time in seconds since the epoch.

2. `import random → random.randint()`

Syntax: `randint(start, end)`

Parameters : (start, end) : Both of them must be integer type values.

Returns : A random integer in range `[start, end]` including the end points.

Errors and Exceptions :

ValueError : Returns a `ValueError` when floating point values are passed as parameters.

TypeError : Returns a `TypeError` when anything other than numeric values are passed as parameters.

3. `import sys → sys.exit()` Τελειώνει το πρόγραμμα

4. `input()` → Περιμένει να πατήσουμε κάτι από το πληκτρολόγιο

Syntax

`input(prompt)`

Parameter: `prompt` (optional) A message you display to the user before taking input.

Return Value: Always returns the entered data as **a string**.

`print("The rose is", input("What color is the rose? "))`

<https://www.geeksforgeeks.org/python/python-input-function/>

Python if .. elif ..

```
If ..... :  
    # command  
    # command  
    # command  
elif ..... :  
    # command  
    # command  
    # command  
else:  
    # command  
    # command  
    # command
```

Τελεστές Ελέγχου

Σύμβολο	Χρήση
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
==	Equal to
!=	Not equal to

Guess the Number

1. Ο υπολογιστής διαλέγει έναν τυχαίο αριθμό από 1 έως 100.
2. Ο παίκτης μαντεύει αριθμούς **και έχει 6 προσπάθειες**.
3. Το πρόγραμμα λέει αν ο αριθμός είναι **μεγαλύτερος ή μικρότερος**.
4. Κάθε μαντεψιά έχει **οικονομική αξία**:
 - ο Πρώτη προσπάθεια → 5€
 - ο Δεύτερη → 4€
 - ο Τρίτη → 3€
 - ο ...
 - ο Έκτη → χάνεις 1€
 - ο Έβδομη → χάνεις 2€
5. Στο τέλος, εμφανίζεται:

- Ο αριθμός των προσπαθειών
- Το συνολικό ποσό που κέρδισε ή έχασε ο παίκτης

binary search

Είναι μια στρατηγική **αναζήτησης σε ταξινομημένο διάστημα**. Κάθε φορά που μαντεύεις έναν αριθμό, **διαιρείς το διάστημα στη μέση** και περιορίζεις την αναζήτηση.

Αρχικά: n στοιχεία

Μετά 1 προσπάθεια $\rightarrow \frac{n}{2}$ στοιχεία

Μετά 2 προσπάθειες $\rightarrow \frac{n}{4}$ στοιχεία

Μετά 3 προσπάθειες $\rightarrow \frac{n}{8}$ στοιχεία

...

Μετά k προσπάθειες $\rightarrow \frac{n}{2^k}$ στοιχεία

Εμείς θέλουμε να μείνουμε με ένα στοιχείο άρα:

$$\frac{n}{2^k} \leq 1 \rightarrow n \leq 2^k \rightarrow \log_2 n \leq k$$

$$\text{Άρα } k \geq \log_2 100 \geq 6.64$$