Παραγωγή τυχαιότητας: η βιβλιοθήκη random της Python

Η βιβλιοθήκη `random` είναι μια από τις βιβλιοθήκες (modules) που περιέχονται στην τυπική διανομή της python. Χρησιμεύει για την παραγωγή ψευδο-τυχαίων αριθμών, καθώς και να επιλέγει με τυχαίο τρόπο στοιχεία από μια ακολουθία, να αναδιατάσσει τα στοιχεία μιας ακολουθίας με τυχαίο τρόπο, κλπ.

Η βιβλιοθήκη περιλαμβάνει κάποιες χρήσιμες συναρτήσεις που χρησιμοποιούνται σε παιχνίδια, προσομοιωτές, κλπ.

Οι κυριότερες από αυτές είναι:

`random.random()`

Η μέθοδος αυτή επιστρέφει ένα τυχαίο πραγματικό αριθμό x με τιμή 0 <= x < 1 .

```
import random
random.random()
0.6575867922996889
```

Ένας κώδικας που ελέγχει την τυχαιότητα της συνάρτησης αυτής είναι ο παρακάτω που επιλύει το εξής πρόβλημα:

Να δημιουργήσετε μια λίστα με 1000 τυχαίους αριθμούς στο πεδίο 0,1 και να ελέγξετε την κατανομή τους στα διαστήματα [0, 0.1], [0.1, 0.2] ... [0.9, 1.0]

```
import random
r = []
for _ in range(1000):
    r.append(random.random())
freq = {}
for i in r:
    for j in [i/10 for i in range(11)]:
        if i <= j:
            freq[j] = freq.get(j, 0) + 1
            break
print(freq)
```

Μια τυπική έξοδος του προγράμματος αυτού είναι:

```
{0.3: 120, 0.7: 103, 0.5: 99, 0.6: 88, 0.8: 91, 0.4: 90, 0.2: 93, 0.1: 97,
1.0: 100, 0.9: 119}
```

Παρατηρούμε ότι για όλες τις περιοχές υπάρχουν περίπου 100 τιμές, με κάποια όμως απόκλιση. Αν ξανατρέξουμε το πρόγραμμα θα παρατηρήσουμε άλλη κατανομή. Αν αλλάξει το πλήθος τιμών της λίστας r τι πιστεύετε ότι θα γίνει, πχ αν η λίστα περιέχει 100 ή 1,000,000 τιμές;

`random.randint(a,b)`

Η συνάρτηση αυτή παράγει ένα τυχαίο ακέραιο αριθμό στο διάστημα `[a, b]`, όπου a,b είναι ακέραιοι και a > b. Για παράδειγμα αν θέλουμε έναν τυχαίο ακέραιο αριθμό από 1 μέχρι 100, αυτό γίνεται ως εξής:

```
import random
random.randint(1,100)
61
```

Μπορούμε συνεπώς να τροποποιήσουμε τον παραπάνω κώδικα ώστε να δημιουργήσουμε μια λίστα ακεραίων με τιμές στο διάστημα [1,100] και στη συνέχεια να βρούμε τη συχνότητα εμφάνισης τιμών στα διαστήματα [1,10], [11, 20], ... [91,100] ως εξής:

```
import random
r = []
for _ in range(1000):
    r.append(random.randint(1,100))
freq = {}
for i in r:
    for j in [i for i in range(10, 101, 10)]:
        if i <= j:
            freq[j] = freq.get(j, 0) + 1
            break
print(freq)
```

Μια τυπική έξοδος αυτού του προγράμματος είναι:

```
{30: 116, 80: 101, 40: 79, 100: 99, 90: 100, 10: 102, 20: 92, 70: 116, 60:
97, 50: 98}
```

`random.randrange(a, b, step)`

Μια παραλλαγή της προηγούμενης συνάρτησης είναι η `randrange(a, b, step)` η οποία επιστρέφει μια τυχαία τιμή στο πεδίο τιμών της range(a, b, step).

Για παράδειγμα για να τυπώσουμε 20 τυχαίους ακέραιους ζυγούς αριθμούς μικρότερους του 10 εκτελούμε τον παρακάτω κώδικα:

```
for _ in range(20):
    print(random.randrange(0,10, 2), end = ",")
```

Μια τυπική έξοδος του προγράμματος αυτού είναι:

```
6,4,8,4,2,2,8,2,8,0,0,8,0,2,0,2,6,0,2,6,
```

`random.uniform(a,b)`

Μια ακόμη χρήσιμη συνάρτηση της βιβλιοθήκης αυτής είναι η `random.uniform(a,b)` η οποία έχει παρόμοια συμπεριφορά με την `random.randint(a,b)` μόνο που επιστρέφει πραγματικούς αριθμούς στο πεδίο [a,b], αντί για ακεραίους.

Παράδειγμα

```
import random
random.uniform(1,100)
22.246065371204498
```

`random.seed(s)`

Όταν παράγουμε μια ακολουθία τυχαίων αριθμών με κάποια από τις συναρτήσεις που είδαμε ως τώρα, κάθε νέος τυχαίος αριθμός παράγεται από τον προηγούμενό του με κάποιο μετασχηματισμό. Το θέμα όμως είναι ποιος αριθμός ξεκινάει την διαδικασία. Συνήθως ο αρχικός αυτός αριθμός, που ονομάζεται σπόρος της ακολουθίας, παράγεται από την χρονοσήμανση του ρολογιού του υπολογιστή ώστε να είναι διαφορετικός κάθε φορά που καλούμε τις συναρτήσεις της random.

Κάποιες φορές όμως επιθυμούμε η ακολουθία τυχαίων αριθμών να είναι η ίδια όλες τις φορές που τρέχει το πρόγραμμά μας. Αυτό μπορεί να είναι χρήσιμο για την περίπτωση που επιθυμούμε να ξαναπαράγουμε ακριβώς την ίδια συμπεριφορά τυχαιότητας σε ένα πείραμα ή σε ένα τρέξιμο του προγράμματος που περιλαμβάνει μια τυχαία ακολουθία, κατά τη φάση εκσφαλμάτωσης.

Η βιβλιοθήκη random μας επιτρέπει να ορίσουμε εμείς οι ίδιοι τον τυχαίο αυτό σπόρο της ακολουθίας με κλήση της συνάρτησης `random.seed(s)`, όπου `s` ένας ακέραιος.

Ας δούμε ένα παράδειγμα:

```
random.seed(100)
for _ in range(20):
    print(random.randrange(0,10, 2), end = ",")
2,6,6,2,6,4,6,8,0,8,0,0,6,4,0,2,4,2,4,2,

random.seed(100)
for _ in range(20):
    print(random.randrange(0,10, 2), end = ",")

2,6,6,2,6,4,6,8,0,8,0,0,6,4,0,2,4,2,4,2,
```

Παρατηρούμε ότι και τις δύο φορές που τρέξαμε τον κώδικα, αφού ορίσαμε τον σπόρο στην αρχική τιμή 100, πήραμε ακριβώς την ίδια ακολουθία ακέραιων αριθμών.

Με τον ίδιο τρόπο μπορούμε να καλέσουμε την seed() πριν την κλήση και των άλλων συναρτήσεων παραγωγής τυχαίων ακολουθιών, όπως της random(), randint(), uniform().

`random.choice(seq)`

Πολλές φορές θέλουμε να επιλέξουμε ένα στοιχείο μιας ακολουθίας. Αυτό γίνεται με χρήση της `random.choice(seq)` , όπου seq είναι μια ακολουθία, λίστα, συμβολοσειρά, πλειαδα.

Ακολουθεί ένα παράδειγμα, στο οποίο θέλουμε να πάρουμε μια τυχαία μέρα της βδομάδας.

```
wd = ("Δε", "Τρ", "Τε", "Πε", "Πα", "Σα", "Κυ")
random.choice(wd)
'Τρ'
```

Παραλλαγή της είναι η συνάρτηση `random.choices(seq, weights, k)` η οποία επιστρέφει μια ή περισσότερες τιμές (η παράμετρος κ ορίζει το πλήθος τιμών που θα επιστραφεί), από μια ακολουθία seq, ενώ η ακολουθία weights καθορίζει μια κατανομή εμφάνισης κάθε μέλους της αρχικής ακολουθίας.

Ένα παράδειγμα, Αν υποθέσουμε ότι η πιθανότητα να αθληθούμε είναι μικρή κατά τις εργάσιμες μέρες και μεγαλύτερη το σαββατοκύριακο, θα μπορούσαμε να παράγουμε τέσσερες τυχαίες μέρες άθλησης το μήνα χρησιμοποιώντας τον ακόλουθο κώδικα, στον οποίο ορίζουμε ότι τη Δευτέρα δεν αθλούμαστε, η Τετάρτη και Σάββατο έχουν διπλάσια βαρύτητα από τις υπόλοιπες εργάσιμες μέρες και η Κυριακή τριπλάσια:

```
wd = ("Δε", "Τρ", "Τε", "Πε", "Πα", "Σα", "Κυ")
random.choices(wd, [0, 1, 2, 1, 1, 2, 3], 4))
['Κυ', 'Πε', 'Κυ', 'Σα']
```

Παρατηρούμε ότι οι 4 προτάσεις άθλησης είναι από μια το Σάββατο και την Πέμπτη και δύο την Κυριακή.

Για να ελέγξουμε τη χρήση αυτής της συνάρτησης συλλέγουμε προτάσεις διακοσίων ημερών άθλησης και καταγράφουμε τη συχνότητα:

```
import random
wd = ("Δε", "Τρ", "Τε", "Πε", "Πα", "Σα", "Κυ")
we = [0, 1, 2, 1, 1, 2, 3]
mytraining = []
for _ in range(20):
    mytraining.extend(random.choices(wd, weights=we, k=10))
freq = {}
for i in mytraining:
    freq[i] = freq.get(i, 0) + 1
print(freq)
```

Ένα τυπικό αποτέλεσμα είναι:

```
{'Τε': 37, 'Σα': 41, 'Τρ': 16, 'Κυ': 65, 'Πε': 22, 'Πα': 19}
```

Παρατηρούμε την απουσία της Δευτέρας και την αυξημένη συχνότητα της Κυριακής στον πληθυσμό.

`random.shuffle(seq)`

Μια χρήσιμη συνάρτηση είναι η shuffle που επιτρέπει το ανακάτεμα μιας ακολουθίας.

Ας δούμε ένα παράδειγμα. Έστω ότι παράγουμε μια λίστα που περιέχει τα φύλλα μιας τράπουλας τα οποία κωδικοποιούνται με βάση την αξία τους 1,2, ... 10, J, Q, K και τον τύπο (clubs, diamonds, hearts, spades).

Ο παρακάτω κώδικας επιτρέπει την δημιουργία μιας τράπουλας και το ανακάτεμα των φύλλων, κάτι που συχνά χρειαζόμαστε σε ένα παιχνίδι με χαρτιά.

```
import random
cards = []
values = ['c', 'd', 'h', 's',  ]
for _ in range(1,11):
    cards.extend([str(_)+x for x in values])
for f in ['J', 'Q', 'K']:
    cards.extend([f+x for x in values])
print(cards)
random.shuffle(cards)
print(cards)
```

Ο κώδικας αυτός τυπώνει αρχικά την τράπουλα με τη σειρά που δημιουργήθηκε:

```
['1c', '1d', '1h', '1s', '2c', '2d', '2h', '2s', '3c', '3d', '3h', '3s',
'4c', '4d', '4h', '4s', '5c', '5d', '5h', '5s', '6c', '6d', '6h', '6s',
'7c', '7d', '7h', '7s', '8c', '8d', '8h', '8s', '9c', '9d', '9h', '9s',
'10c', '10d', '10h', '10s', 'Jc', 'Jd', 'Jh', 'Js', 'Qc', 'Qd', 'Qh',
'Qs', 'Kc', 'Kd', 'Kh', 'Ks']
```

Στη συνέχεια όμως μετά την κλήση της `random.shuffle()` έχουμε μια ανακατεμένη τράπουλα:

```
['5c', '2d', '3h', '4c', '3c', 'Qh', 'Qc', '10d', 'Jh', '7c', '10h', '4h',
'8s', '6c', '1h', '3s', 'Kh', '6h', '1d', '3d', '1c', '2c', 'Kc', '5d',
'6d', '9c', 'Jc', 'Js', '7s', '5h', '2h', '9s', '8c', '8d', '10c', '8h',
'5s', '2s', '7h', '4d', '6s', '9d', '7d', 'Ks', 'Kd', '9h', 'Qs', '10s',
'Qd', 'Jd', '4s', '1s']
```