



Bilgisayar Programlama 2

Ders Notu: 5

Listeler

- **Listeler ve rastgele sayı üretme modülü:**
- Rastgele sayı üretme modülünde listeler üzerinde çalışan bazı fonksiyonlar vardır.

Function	Description
<code>choice(L)</code>	picks a random item from L
<code>sample(L, n)</code>	picks a group of n random items from L
<code>shuffle(L)</code>	Shuffles the items of L

- Karıştırma işlevi (`shuffle`) orijinal listeyi değiştirir, bu nedenle listenizin değişmesini istemiyorsanız, bir kopyasını oluşturmanız gerekecektir.
- **Örnek 1:** İsim listesinden bir isim seçmek için "seçme" (`choice`) özelliği kullanımı:

```
1 from random import choice
2 names = ['Joe', 'Bob', 'Sue', 'Sally']
3 current_player = choice(names)
4
5 print(current_player)
```

Listeler

- **Örnek 2:** Örnekleme (sample) işlevi, seçim işlevine benzer. Seçim işlevi bir listeden tek bir öge seçerken, örneklem işlevi birden fazla öge seçmek için kullanılabilir.

```
1 from random import sample
2 names = ['Joe', 'Bob', 'Sue', 'Sally']
3 team = sample(names, 2)
4
5 print(team)
```

- **Örnek 3:** `choice` fonksiyonu, stringlerde de çalışır ve bir metinden rastgele bir karakter seçer. Ekranı rastgele karakterle doldurmak için `choice` fonksiyonunu kullanan bir örnek.

```
1 from random import choice
2
3 s='abcdefghijklmnopqrstuvwxyz1234567890!@#%^&*()'
4 for i in range(10000):
5     | print(choice(s), end='')
```

Listeler

- **Örnek 4:** Bir oyunda oyuncuların rastgele sıralanmasını sağlamak için karıştırma işleminin kullanımı.

```
1  from random import shuffle
2
3  players = ['Joe', 'Bob', 'Sue', 'Sally']
4  shuffle(players)
5  for p in players:
6      print(p, 'it is your turn.')
7      # code to play the game goes here...
```

- **Örnek 5:** Burada, bir grup insanı ikişer kişilik takımlara ayırmak için karıştırma (shuffle) yöntemini kullanıyoruz. Bize isimler adı verilen bir liste verildiğini varsayalım.

```
1  import random
2
3  isimler = ["Joe", "Bob", "Sue", "Sally"]
4
5  random.shuffle(isimler)
6
7  print("Karıştirilmiş isimler:")
8  print(isimler) # Karıştirilmiş isimleri yazdır
9
10 print("Oluşturulan takımlar:")
11 for i in range(0, len(isimler), 2):
12     print(isimler[i], "-", isimler[i+1])
```

- Takımlardaki her öğe, iki isimden oluşan bir listedir. Kodun çalışma şekli, isimleri rastgele bir sırada olacak şekilde karıştırmak şeklindedir. Karıştırılmış listedeki ilk iki isim birinci takım olur, sonraki iki isim ikinci takım olur vb.
- İsim listesinde ikişer ikişer atlamak için isteğe bağlı üçüncü argümanı (range) kullanılmıştır.

Listeler

- **Split:** Bölme yöntemi, bir string'in kelimelerinin listesini döndürür. Yöntem, kelimelerin boşluk, sekme veya satır sonu karakterleri gibi boşluklarla ayrıldığını varsayar.

```
1 s = 'Hi! This is a test.'
2 print(s.split())
```

['Hi!', 'This', 'is', 'a', 'test.']

- Split, stringi boşluklardan ayırdığı için noktalama işaretleri kelimelerin bir parçası olacaktır. Yaygın noktalama işaretlerini içeren 'punctuation' adlı bir dizgi değişkeni içeren string adlı bir modül vardır. Aşağıdaki kodla bir metinden noktalama işaretlerini kaldırabiliriz:

```
1 from string import punctuation
2
3 s = 'Hi! This is a test.'
4
5 for c in punctuation:
6     | s = s.replace(c, '')
7
8 print(s)
```

Listeler

- **Örnek 5:** Bir metin dizisinde belirli bir kelimenin kaç kez geçtiğini sayan bir program.

```
1  from string import punctuation
2  s = input('Bir metin giriniz: ')
3  for c in punctuation:
4      s = s.replace(c, '')
5
6  s = s.lower()
7  L = s.split()
8
9  word = input('Bir kelime giriniz: ')
10 print(word, 'kelimesi', L.count(word), 'kez bulundu.')
```

Listeler

- **join:** `join` metodu bir anlamda `split` metodunun tam tersidir. Bir string listesini alıp tek bir string halinde birleştiren bir string metodudur.

Operation	Result
' '.join(L)	A B C
''.join(L)	ABC
', '.join(L)	A, B, C
'***'.join(L)	A***B***C

- **Örnek:** Verilen bir kelimenin anagramını oluşturan bir program yazın. Bir kelimenin anagramı, kelimedeki harfleri farklı bir sırada kullanır. Örneğin, "there" kelimesinin iki anagramı "three" ve "ether"dir.
- Bu problem, karıştırma (shuffle) işlemiyle çözebilir, ancak karıştırma işlemi yalnızca listelerle çalışır. Bu durumda, string'i bir listeye dönüştürmek, üzerinde karıştırma işlemi uygulamak ve ardından listeyi tekrar dizeye dönüştürmektir. Bir stringi (s) listeye dönüştürmek için liste(s) kullanılabilir. Listeyi tekrar dizeye dönüştürmek için birleştirme (join) işlemi kullanılacaktır.

```
1 from random import shuffle
2
3 word = input('Bir kelime giriniz: ')
4 harf_list = list(word)
5 shuffle(harf_list)
6 anagram = ''.join(harf_list)
7
8 print(anagram)
```

Listeler

- **List comprehensions:** Liste oluşturma yöntemini kullanmadan da tamamlanabilir, ancak aşırı olduğunda, daha uzun liste oluşturma yöntemlerine kıyasla hem daha hızlı yazılabilir hem de daha kolay okunurluğu mümkün kılar.
- **Örnek 1:** 1 ile 100 arasında rastgele 50 sayıdan oluşan bir L listesi üreten bir program yazın.

```
1  from random import randint
2
3  L = [randint(1,100) for i in range(50)]
4  print(L)
```

- **Örnek 2:** L listesindeki her elemanı karesiyle değiştirin. `L=[1,3,4]`
`L = [i**2 for i in L]`
- **Örnek 3:** L listesindeki 50'den büyük olan öğelerin sayısını sayın.,

```
L=[1,3,4,85]
a= len([i for i in L if i>50])
print(a)
```

- **Örnek 4:** 1 ile 100 arasında sayılar içeren bir L listesi verildiğinde, ilk elemanı L listesindeki birlerin sayısını, ikinci elemanı L listesindeki ikilerin sayısını vb. içeren yeni bir liste oluşturun.

```
L=[1,3,4,85,4]
frequencies = [L.count(i) for i in range(1,101)]
print(frequencies)
```

Listeler

- **Örnek 5:** join metodu, bir stringi hızlıca oluşturmak için genellikle 'list comprehension' ile kullanılabilir. Bu örnekte, rastgele seçilmiş 1000 harften oluşan bir string oluşturulur.

```
1  from random import choice
2  alphabet = 'abcdefghijklmnopqrstuvwxyz'
3  s = ''.join([choice(alphabet) for i in range(1000)])
4  print(s)
```

Sorular

- **Soru 1:** Kullanıcıdan bir metin girmesini isteyen ve ardından metindeki 'a', 'an' ve 'the' kelimelerinin sayısını hesaplayan bir program yazın.

```
1  metin = input("Bir metin girin: ")
2
3  metin = metin.lower()
4  kelimeler = metin.split()
5
6  a_say = 0
7  an_say = 0
8  the_say = 0
9
10 for kelime in kelimeler:
11     if kelime == "a":
12         a_say += 1
13     elif kelime == "an":
14         an_say += 1
15     elif kelime == "the":
16         the_say += 1
17
18 print("a sayisi:", a_say)
19 print("an sayisi:", an_say)
20 print("the sayisi:", the_say)
```

Sorular

- **Soru 2:** Kullanıcının beş sayı (string olarak) girmesine olanak tanıyan bir program yazın. Kullanıcının girdiği sayıları artı işaretleriyle ayırarak bir string oluşturun. Örneğin, kullanıcı 2, 5, 11, 33 ve 55 girerse, string '2+5+11+33+55' olmalıdır.

```
1  sayilar = []
2
3  for i in range(5):
4      s = int(input("Bir sayi girin: "))
5      sayilar.append(str(s))
6
7  sonuc = "+".join(sayilar)
8
9  print("Oluşturulan string:", sonuc)
```

Sorular

- **Soru 3:**
- **(a)** Kullanıcıdan bir cümle girmesini isteyin ve cümlenin üçüncü kelimesini yazdırın.
- **(b)** Kullanıcıdan bir cümle girmesini isteyin ve cümlenin her üçüncü kelimesini yazdırın.

```
1  cumle = input("Bir cümle girin: ")
2  kelimeler = cumle.split()
3
4  if len(kelimeler) >= 3:
5      print("Üçüncü kelime:", kelimeler[2])
6  else:
7      print("Cümlede üçten az kelime var.")
```

```
1  cumle = input("Bir cümle girin: ")
2  kelimeler = cumle.split()
3
4  print("Her üçüncü kelime:")
5  for i in range(2, len(kelimeler), 3):
6      print(kelimeler[i])
```

Sorular

- **Soru 4:** Kullanıcıdan bir cümle girmesini isteyen ve ardından cümlenin kelimelerini rastgele yeniden düzenleyen bir program yazın. Noktalama işaretleri veya büyük harf kullanımı göz ardı edilebilir.

```
1  from random import shuffle
2
3  cumle = input("Bir cümle girin: ")
4
5  kelimeler = cumle.split()
6  shuffle(kelimeler)
7
8  yeni_cumle = " ".join(kelimeler)
9  print("Karıştırılmış cümle:")
10 print(yeni_cumle)
```

Sorular

- **Soru 5:** Listedeki rastgele bir söz seçip ekrana yazdıran bir Python programı yazınız.

```
1  import random
2
3  sozler = [
4      "Başlamak bitirmenin yarısıdır.",
5      "Azim, yeteneğin önüne geçer.",
6      "Bugün yap, yarına bırakma.",
7      "Küçük adımlar büyük sonuçlar doğurur.",
8      "Disiplin, hedefe giden köprüdür."
9  ]
10
11  secim = random.choice(sozler)
12
13  print("Günün sözü:")
14  print(secim)
```

Sorular

- **Soru 6:** Basit bir piyango çekiliş programı yazın. Piyango çekilişi, 1 ile 48 arasında altı farklı sayıdan oluşmalıdır.

```
1  from random import sample
2
3  # 1-48 arasından 6 farklı sayı seç
4  cekilis = sample(range(1, 49), 6)
5
6  cekilis.sort() # sıralı gösterim
7
8  print("Piyango sonuçları:")
9  for s in cekilis:
10 |     print(s, end=" ")
```



Sorular

- **Soru 7:** On sorudan oluşan bir liste ve bu soruların cevaplarından oluşan bir liste içeren basit bir bilgi yarışması oyunu yazın.
- Oyun, oyuncuya cevaplama için rastgele seçilmiş dört soru vermelidir.
- Soruları tek tek sormalı ve oyuncuya soruyu doğru mu yoksa yanlış mı cevapladığını söylemelidir.
- Sonunda, dört sorudan kaçını doğru cevapladığını yazdırmalıdır.

Sorular

- Soru 7:

```
1 from random import sample
2
3 sorular = [
4     "Fransa'nin başkenti nedir?",
5     "5 + 7 kaçtır?",
6     "Dünyanın en büyük okyanusu hangisidir?",
7     "Python bir programlama dili midir? (evet/hayır)",
8     "Türkiye'nin para birimi nedir?",
9     "9 * 6 kaçtır?",
10    "En hızlı kara hayvani hangisidir?",
11    "Su kaç derecede kaynar?",
12    "Ay bir gezegen midir? (evet/hayır)",
13    "2 üzeri 5 kaçtır?"
14 ]
15
16 cevaplar = [
17     "paris",
18     "12",
19     "pasifik",
20     "evet",
21     "tl",
22     "54",
23     "çita",
24     "100",
25     "hayır",
26     "32"
27 ]
28
29 dogru_sayisi = 0
30
31 secili_indeksler = sample(range(10), 4)
32
33 for i in secili_indeksler:
34     cevap = input(sorular[i] + " ").lower()
35
36     if cevap == cevaplar[i]:
37         print("Doğru!")
38         dogru_sayisi += 1
39     else:
40         print("Yanlış! Doğru cevap:", cevaplar[i])
41
42 print("Toplam doğru sayınız:", dogru_sayisi, "/ 4")
```

