

Loendi mõistmine/Loendiavaldis

Loendiavaldis (*list comprehension*, Loendi mõistmine) on Pythonis loendite generaator, mis võimaldab luua täidetud loendeid. See konstruktsioon võtab ainult ühe rea ja sellel on järgmine vorming:

[avaldise for muutuja in kogus]

Standardne loendi loomise viis:

1. Luuakse tühi loend
2. Kasutades tsüklit *for* ja *append* meetodit, lisatakse andmed loendisse

Loendiavaldisi kasutatakse sageli loendite initsialiseerimiseks.

```
# Näide 1...10 numbrite loendi loomisest alates
numbers = []
for i in range(1, 11):
    numbers.append(i)
print(*numbers, sep = " ")
1 2 3 4 5 6 7 8 9 10
```

```
# Näide 1...10 arvude loendi loomisest loendiavaldise abil
numbers = [i for i in range(1, 11)]
print(*numbers, sep = " ")
1 2 3 4 5 6 7 8 9 10
```

```
# Loend, mis on täidetud 1 kuni 9 täisarvude ruutudega
squares = [i ** 2 for i in range(1, 10)]
print(squares)
[1, 4, 9, 16, 25, 36, 49, 64, 81]
```

```
# Loend on täidetud stringimärkidega
chars = [c for c in 'Python']
print(chars)
['P', 'y', 't', 'h', 'o', 'n']
```

Loendiavaldistes saab kasutada tingimuslikku operaatorit. Näiteks kui peab looma paarisarvude loendi vahemikus 0 kuni 20

```
evens = [i for i in range(21) if i % 2 == 0]
print(evens)
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

Ülesanne 1

1. Looge muutujad

```
word = 'Hello',
numbers = [1, 10, 5, 15, 20, 19, 13, 22],
days = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
```

2. Looge loendid muutujaväärtuste ja loendi avaldiste abil:

- [0, 0, 0, 0, 0, 0, 0, 0]
- ['M', 'T', 'W', 'T', 'F', 'S', 'S']
- ['HH', 'ee', 'll', 'll', 'oo']
- [10, 5, 15, 20]
- ['Tuesday', 'Thursday']
- ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']
- ['day', 'day', 'day', 'day', 'day', 'day', 'day']

Ülesanne 2

Programmi sisend on tekstistring. Kirjutage loendi avaldise abil programm, mis kuvab kõik antud stringi numbrimärgid. Arvutage ka kõigi numbrite summa.

Mitmemõõtmeline loend

Loend on järjestatud elementide jada, mis on nummerdatud alates 0. Loendi elemendid võivad olla mis tahes andmetüübid: numbrid, stringid, tõeväärtused (Booli väärtused) jne.

Loendite elemendid võivad olla muud loendid. Selliseid loendeid nimetatakse **mitmemõõtmeliseks loenditeks**.

Mitmemõõtmeliste loenditega töötamine ei erine põhimõtteliselt tööst näiteks numbrite või stringide loenditega. Mitmemõõtmelise loendi loomiseks eraldatakse nurksulgudes elemendid komadega:

```
my_list = [[0], [1, 2], [3, 4, 5]]
```

Muutuja `my_list` viitab loendile, mis koosneb muudest loenditest.

Kuna `my_list` pesastussügavus on kaks, nimetatakse seda loendit tavaliselt kahemõõtmeliseks loendiks. Praktikas töötatakse tavaliselt kahemõõtmeliste loenditega, harvemini kolmemõõtmeliste loenditega.

```
1. my_list = [[0], [1, 2], [3, 4, 5]]
2. print(my_list)
3. print(my_list[0])
4. print(my_list[1])
5. print(my_list[2])
6. print(len(my_list))
```

Tulemus:

```
[[0], [1, 2], [3, 4, 5]]
[0]
[1, 2]
[3, 4, 5]
3
```

- rida 1 loob loendi ja määrab selle muutujale `my_list`. Loendil on kolm elementi ja iga element on ka loend:
 - element `my_list[0]` on loend `[0]`;
 - element `my_list[1]` on loend `[1, 2]`;
 - element `my_list[2]` on loend `[3, 4, 5]`.
- rida 2 prindib kogu loendi `my_list`;
- rida 3 prindib elemendi `my_list[0]`;
- rida 4 prindib elemendi `my_list[1]`;

- rida 5 prindib elemendi `my_list[2]`;
- real 6 trükitakse loendis `my_list` olevate elementide arv, s.t number 3.

Loendi elemente saab ka indekseerida.

Funktsioonid `len()`, `max()`, `min()`

Sisseehitatud funktsioon `len()` tagastab loendi elementide arvu, mitte kõigi loendite elementide koguarvu.

Kui peab loendama pesastatud loendi elementide koguarvu, võib kasutada `for`-tsükli funktsiooniga `len()`

Funktsioonid `min()` ja `max()`. Kui nendele funktsioonidele edastatakse mitu loendit, tagastatakse üks edastatud loenditest täielikult. Sel juhul toimub võrdlus elementide kaupa: kõigepealt võrreldakse loendite esimesi elemente. Kui need ei ole võrdsed, siis funktsioon `min()` tagastab selle loendi, mille esimene element on väiksem, `max()` - vastupidi. Kui esimesed elemendid on võrdsed, siis võrreldakse teisi jne.

Mitmemõõtmeliste loendite loomine

Mitmemõõtmelise loendi loomiseks võib kasutada literaalset (sõnasõnalist) kirjutamisviisi - nurksulgudes komadega eraldatud elementide loetlemist:

```
my_list = [[0], [1, 2], [3, 4, 5]]
```

Mõnikord peab looma mitmemõõtmelise loendi, mis on täidetud teatud reegli - malli järgi . Näiteks loend pikkusega `n`, mis sisaldab loendeid pikkusega `m`, millest igaüks on täidetud nullidega.

Mitmemõõtmelise loendi elementide itereerimine ja kuvamine

Loendi elemendile juurdepääsu saamiseks määratakse selle elemendi indeks nurksulgudes. Kahemõõtmeliste loendite puhul tuleb määrata kaks indeksit (igaüks eraldi nurksulgudes), kolmemõõtmelise loendi puhul kolm indeksit jne.

Kui on vaja itereerida läbi kõik loendi elemendid (näiteks nende kuvamiseks ekraanil), kasutatakse tavaliselt *nested* tsükleid (tsükkel tsükli).

Ülesanne

1. Programmi sisendiks on arv `n`. Kirjutage programm, mis loob ja kuvab rida-realt loendi, mis koosneb `n` loendist. `[[1, 2, ..., n], [1, 2, ..., n], ..., [1, 2, ..., n]]`.
2. Programmi sisendiks on arv `n`. Kirjutage programm, mis loob ja kuvab ridade kaupa pesastatud loendi, mis koosneb `n` loendist. `[[1], [1, 2], [1, 2, 3], ..., [1, 2, ..., n]]`.