

Sąrašai **list**

Sąrašas – duomenų rinkinys.

Tuščias sąrašas: saraso_pavadinimas = []

Sąrašas su duomenimis: saraso_pavadinimas = [reikšmė, reikšmė, reikšmė]

1.

```
skaiciai = [1,2,3]
print (skaiciai)
```

```
=====
[1, 2, 3]
```

2.

```
vardai = ['Mikas', 'Marius', 'Martynas']
print (vardai)
```

```
['Mikas', 'Marius', 'Martynas']
```

```
-----

vardai = [
    'Mikas',
    'Marius',
    'Martynas'
]
print(vardai)
```

```
['Mikas', 'Marius', 'Martynas']
```

3.

```
duomenys = ['Mikas',5,8.5]
print (duomenys)
```

```
=====
['Mikas', 5, 8.5]
```

4.

Sąrašo papildymui naudojama **append()**, **extend()**

<pre>duomenys = ['Mikas',5,8.5] print (duomenys) duomenys.append('Jonas') print (duomenys) duomenys.extend([5,7.5]) print (duomenys)</pre> <pre>=====</pre> <pre>['Mikas', 5, 8.5] ['Mikas', 5, 8.5, 'Jonas'] ['Mikas', 5, 8.5, 'Jonas', 5, 7.5]</pre>	<pre>duomenys = ['Mikas',5,8.5] print (duomenys) duomenys.append('Jonas') print (duomenys) papildyti = [5, 7.5] duomenys.extend(papildyti) print (duomenys)</pre> <pre>=====</pre> <pre>['Mikas', 5, 8.5] ['Mikas', 5, 8.5, 'Jonas'] ['Mikas', 5, 8.5, 'Jonas', 5, 7.5]</pre>
--	---

5.

Norint sužinoti kiek sąrašo elementų naudojama **len()**.

```
duomenys = ['Mikas',5,8.5]
print (len(duomenys))
```

```
====
3
```

```
miestas = 'Vilnius'
print (len(miestas))
```

```
====
7
```

6.

Indeksavimas

```
pazymiai = [7, 8, 5, 9, 10, 6]
```

Elementas iš sąrašo	7	8	5	9	10	6
----------------------------	----------	----------	----------	----------	-----------	----------

Priskirtas indeksas	0	1	2	3	4	5
---------------------	---	---	---	---	---	---

```
medziai = ['Gluosnis', 'Ąžuolas', 'Liepa', 'Berželis', 'Obelis']
```

Elementas iš sąrašo	'Gluosnis'	'Ąžuolas'	'Liepa'	'Berželis'	'Obelis'
---------------------	------------	-----------	---------	------------	----------

Priskirtas indeksas	0	1	2	3	4
---------------------	---	---	---	---	---

Elementas iš sąrašo	'Gluosnis'	'Ąžuolas'	'Liepa'	'Berželis'	'Obelis'
---------------------	------------	-----------	---------	------------	----------

Priskirtas indeksas (nuo galo)	-5	-4	-3	-2	-1
--------------------------------	----	----	----	----	----

7.

```
duomenys = ['Mikas',5,8.5]
print (duomenys[0])
print (duomenys[1])
print (duomenys[2])
```

```
=====
```

```
Mikas
5
8.5
```

```
duomenys = ['Mikas',5,8.5]
print ('Mokinys: ',duomenys[0])
print ('Klasė: ',duomenys[1])
print ('Pažymių vidurkis: ',duomenys[2])
```

```
=====
```

```
Mokinys: Mikas
Klasė: 5
Pažymių vidurkis: 8.5
|
```

8.

```
duomenys = [2,3,4,5]
print(sum(duomenys))
```

```
=====
```

```
14
```

9.

Sąrašo elementų pakeitimas

```
vardai = ['Mikas', 'Marius', 'Martynas']
print(vardai)
print()
vardai[0]= 'Ona'
vardai[1] = 'Sima'
vardai[2]= 'Roma'
print (vardai)
```

```
=====
['Mikas', 'Marius', 'Martynas']

['Ona', 'Sima', 'Roma']
|
```

10.

Sąrašo papildymui į norimą vietą (pagal nurodytą indeksą) naudojama **insert()**

```
vardai = ['Mikas', 'Marius', 'Martynas']
print (vardai)
vardai.insert(0, 'Ona')
print (vardai)
```

```
['Mikas', 'Marius', 'Martynas']
['Ona', 'Mikas', 'Marius', 'Martynas']
|
```

11.

Elemento išėmimui iš sąrašo naudojama **pop()**

```
vardai = ['Mikas', 'Marius', 'Martynas']
print (vardai)
vardai.pop()
print (vardai)
```

```
['Mikas', 'Marius', 'Martynas']
['Mikas', 'Marius']
|
```

```
-----

vardai = ['Mikas', 'Marius', 'Martynas']
print (vardai)
# trina iš galo
vardai.pop()
vardai.pop()
print (vardai)
```

```
['Mikas', 'Marius', 'Martynas']
['Mikas']
```

```
vardai = ['Mikas', 'Marius', 'Martynas']
print (vardai)
# trina nurodyto vietos (pagal indeksą)
vardai.pop(1)
print (vardai)

['Mikas', 'Marius', 'Martynas']
['Mikas', 'Martynas']
.
```

12.

Išvalo sąrašą **clear()**

```
vardai = ['Mikas', 'Marius', 'Martynas']
print (vardai)

vardai.clear()
print (vardai)

['Mikas', 'Marius', 'Martynas']
[]
.
```

13.

Elemento vietą sąraše nurodo **index()**

```
vardai = ['Mikas', 'Marius', 'Martynas']
print (vardai)

print ('Elemento Marius indeksas sąraše:')
print (vardai.index('Marius'))

['Mikas', 'Marius', 'Martynas']
Elemento Marius indeksas sąraše:
1
.
```

14.

```
vardai = ['Mikas', 'Marius', 'Martynas', 'Mindaugas', 'Motiejus']
print (vardai)

# ar sąrašė yra toksai elementas
print ('Marius'in vardai)

print ('Ona'in vardai)

print ('k'in 'Mikas')

['Mikas', 'Marius', 'Martynas', 'Mindaugas', 'Motiejus']
True
False
True
.
```

```
vardai = ['Mikas', 'Marius', 'Martynas', 'Mindaugas', 'Motiejus']

if 'Mikas' in vardai:
    print ('Mokinys Mikas yra sąrašuose')
else:
    print ('Mokinio Miko nėra sąrašuose')

if 'Petras' in vardai:
    print ('Mokinys Petras yra sąrašuose')
else:
    print ('Mokinio Petro nėra sąrašuose')

Mokinys Mikas yra sąrašuose
Mokinio Petro nėra sąrašuose
|
```

15.

Suskaičiuoti kiek kartų elementas pasikartoja sąrašė naudojamas **count()**

```
vardai = ['Mikas', 'Marius', 'Mikas', 'Mindaugas', 'Mikas']
print (vardai)

print (vardai.count('Mikas'))

print (vardai.count('Ona'))

['Mikas', 'Marius', 'Mikas', 'Mindaugas', 'Mikas']
3
0
.
```

16.

Sąrašo elementų rūšiavimui ir rikiavimui naudojamas **sort()**, **reverse()**. Sukuriamas naujas sąrašas, bet senasis sąrašas nepakeičiamas.

<code>skaiciai = [5, 3, 1, 12]</code>	<code>skaiciai = [5, 3, 1, 12]</code>
<code>print (skaiciai)</code>	<code>print (skaiciai)</code>
<code>print ()</code>	<code>print ()</code>
<code>skaiciai.sort()</code>	<code>skaiciai.reverse()</code>
<code>print(skaiciai)</code>	<code>print(skaiciai)</code>
<code>skaiciai.reverse()</code>	<code>skaiciai.sort()</code>
<code>print (skaiciai)</code>	<code>print (skaiciai)</code>
=====	=====
<code>[5, 3, 1, 12]</code>	<code>[5, 3, 1, 12]</code>
<code>[1, 3, 5, 12]</code>	<code>[12, 1, 3, 5]</code>
<code>[12, 5, 3, 1]</code>	<code>[1, 3, 5, 12]</code>

```

daiktai = ['knyga', 'skirtukas', 'pieštukas', 'rašalas']
print (daiktai)
print ()

daiktai.sort()
print (daiktai)
print ()

print(sorted(daiktai))
print(daiktai)

```

`['knyga', 'skirtukas', 'pieštukas', 'rašalas']`

`['knyga', 'pieštukas', 'rašalas', 'skirtukas']`

`['knyga', 'pieštukas', 'rašalas', 'skirtukas']`

`['knyga', 'pieštukas', 'rašalas', 'skirtukas']`

17.

Sąrašo karpymas **list slicing**

Nurodomi režiiai:

```
sąrašo_pavadinimas[pradžia:pabaiga]
sąrašo_pavadinimas[pradžia:pabaiga:žingsnis]
```

```
vardai = ['Mikas', 'Rimas', 'Jonas', 'Simas', 'Petras', 'Antanas', 'Jokūbas']
print (vardai[3])
print (vardai[0:3])
print (vardai[2:5])
print ()

print (vardai[:5])
print (vardai[5:])
print ()

print (vardai[0:1])
print (vardai[:1])
print (vardai[0:2])
print (vardai[:2])
print (vardai[:-1])
print ()

atrinkti = vardai[2:6:2]
print (atrinkti)

Simas
['Mikas', 'Rimas', 'Jonas']
['Jonas', 'Simas', 'Petras']

['Mikas', 'Rimas', 'Jonas', 'Simas', 'Petras']
['Antanas', 'Jokūbas']

['Mikas', 'Rimas', 'Jonas', 'Simas', 'Petras', 'Antanas', 'Jokūbas']
['Mikas', 'Rimas', 'Jonas', 'Simas', 'Petras', 'Antanas', 'Jokūbas']
['Mikas', 'Jonas', 'Petras', 'Jokūbas']
['Mikas', 'Jonas', 'Petras', 'Jokūbas']
['Jokūbas', 'Antanas', 'Petras', 'Simas', 'Jonas', 'Rimas', 'Mikas']

['Jonas', 'Petras']
|
```

18.

Sąrašo apjungimui, skaidymui naudojami **join**, **split**

```
'<skirtukas tarp kiekvieno elemento>'.join(<sąrašas>)
```

```
<teksto eilutė>.split(<skirtukas pagal ką skelti>)
```

```
tekstas = ['Mano', 'vardas', 'Mikas']
sakinyys = ' '.join(tekstas)
print (sakinyys)
```

```
=====
Mano vardas Mikas
|
```

```
-----

tekstas = 'Mano vardas Jonas'
sakinyys = tekstas.split()
print (sakinyys)
```

```
=====
['Mano', 'vardas', 'Jonas']
```

```
-----

sakinyys = 'Mano /vardas:Jonas'
print (sakinyys.split())
print (sakinyys.split('/'))
print (sakinyys.split(':'))
```

```
===== ]
['Mano', '/vardas:Jonas']
['Mano ', 'vardas:Jonas']
['Mano /vardas', 'Jonas']
```