

Python objektų tipai ir sakiniai

Albertas Gimbutas

¹Matematikos ir informatikos institutas
Vilniaus universitetas

2014 pavasaris

- 1 Python interpretatorius
- 2 Objektų tipai
- 3 Sąlygos ir ciklo sakiniai
- 4 Funkcijos

Python interpretatorius

```
$ pip install ipython ipdb  
$ ipython
```

```
>>> var = 'egg'  
>>> dir(var)                # Objekto atributai ir metodai  
>>> help(var.replace)       # Objekto dokstringas  
>>> var?                     # Objekto aprašymas (tik ipython)  
>>> # Vykdymo pertraukimas  
>>> import ipdb; ipdb.set_trace()  
>>> # s - step (vykdo sekantį sakinį)  
>>> # n - next (vykdo iki sekančios eilutės)  
>>> # c - continue  
>>> # q - quit  
>>> # Redaguojant kontekstą neperdengti komandų
```

- python.org: Interpretatoriaus naudojimosi instrukcijos

Python virtuali aplinka

Virtuali aplinka leidžia susikurti nuo sistemos izoliuotą Python aplinką (ją susikurti ir naudoti galima ir neturint **sudo** teisių).

```
$ wget https://pypi.python.org/packages/source/v/  
virtualenv/virtualenv-1.9.tar.gz  
$ tar -xzf virtualenv-1.9.tar.gz  
$ python ~/virtualenv-1.9/virtualenv.py venv  
$ venv/bin/pip install pep8  
$ venv/bin/pep8 pirmoji_uzduotis.py
```

- Virtualios aplinkos dokumentacija

Python objektų tipai

Skaiciai (**int**, **float**)

Simbolių eilutės (**string**)

Sąrašai (**list**)

Sąrašai-konstantos (**tuple**)

Žodynai (**dict**)

Aibės (**set**)

Loginiai (**boolean**)

Neapibrėžtos reikšmės

Failai (**files**)

Kompleksiniai skaičiai (**complex**)

Ir kiti

```
123, 3.1415, 0xAF1, 0b101
'namas', "labas"
[1, 2], ['a', ['b', 1]]
(1, 2), ('a', ('b', 1))
{'suma': 9, 'x': [4, 5]}
{'a', 'b', 'c'}
True, False
None
open('duom.txt', 'w')
2.25 + 1.43j
funkcijos, klasės, moduliai
```

Operacijos su skaičiais

```
>>> (2 + 3.0) / 2 # + - * / - aritmetinės operacijos
2.5
>>> (2 + 3) // 2 # // % - sveikųjų sk. operacijos
2
>>> 2**6 # Kėlimas laipsniu
64
>>> round(3.1415, 2) # Apvalinimas
3.14
>>> abs(-3.14) # Modulis
3.14
>>> from math import sqrt, cos, sin, exp, log, pi, e
>>> sqrt(25) # Šaknis
5
>>> pi
3.141592653589793
```

- python.org skaičių dokumentacija

Atsitiktinių skaičių generavimas

```
>>> from random import seed, randint, random, uniform, gauss
>>> # pseudoatsitiktinių skaičių generatoriaus inicial.
>>> seed(777)
>>> randint(1, 1000)
230
>>> seed(777)
>>> randint(1, 1000)
230
>>> # Atsitiktinis dydis iš intervalo [0, 1)
>>> random()
0.44559617334521107
>>> # Atsitiktinis dydis iš intervalo [-1, 1)
>>> uniform(-1, 1)
-0.2628035012556791
>>> # Normaliojo (Gauso) pasiskirstymo su vidurkiu 0
>>> gauss(0, 1) # ir standartiniu nuokrypiu 1 at. dyd.
-0.11252053935604013
```

Operacijos su simbolių eilutėmis

```
>>> s = "spam eggs"
>>> len(s)
9
>>> # s[0] = 'S'
>>> s = 'S' + s[1:]
'Spam eggs'
>>> s.split()
['Spam', 'eggs']
>>> s.replace(' ', '-')
'Spam-eggs'
>>> ','.join(['vienas', 'du', 'trys'])
'vienas, du, trys'
>>> _.split(',')
['vienas', 'du', 'trys']
```

- python.org: Simbolių eilučių metodai

Simbolių eilučių formatavimas

```
>>> '%s %d: %.2f' % ('pavyzdys', 1, 3.1415)
'pavyzdys 1: 3.14'

>>> '{} {}: {:.2f}'.format('pavyzdys', '2', 3.1415)
'pavyzdys 2: 3.14'

>>> '{0} {1:.0f}: {1:.2f}'.format('pavyzdys', 3.1415)
'pavyzdys 3: 3.14'

>>> '{name} {value}'.format(name='pavyzdys', value=4)
'pavyzdys 4'
```

- python.org: Simbolių eilučių formatavimas

Atsitiktinumo operacijos su simbolių eilutėmis

```
>>> from string import letters, digits
>>> from random import choice, sample, shuffle
>>> choice(letters)
'N'
>>> sample(letters + digits, 6)
['5', '3', 'L', 'w', 'q', '4']
>>> a = _
>>> shuffle(a)
>>> a
['L', '5', 'q', '4', 'w', '3']
>>> ''.join(a)
'L5q4w3'
```

Sąrašai

```
>>> m = [2, 4, 1, 3]
>>> m.append(5)           # Prideda elementą į sąrašo galą
>>> m.sort()              # Surūšiuoja sąrašą
[1, 2, 3, 4, 5]
>>> m[1:4]
[2, 3, 4]
>>> m[-2:]
[4, 5]
>>> m[::2]                # sąrašas[nuo:iki:žingsnis]
[1, 3, 5]
>>> m.reverse()          # Apkeičia elementų eiliškumą
[5, 4, 3, 2, 1]
>>> m.extended([6, 7])    # Papildo sąrašą nuo galo
>>> # n = m               # Nukopijuoja sąrašo rodyklę
>>> n = m[:]              # Nukopijuoja sąrašą
>>> from copy import deepcopy
```

• python.org: Sąrašai (list)

Funkcija range()

```
>>> range(7)
[0, 1, 2, 3, 4, 5, 6]
>>> range(1,8)
[1, 2, 3, 4, 5, 6, 7]
>>> range(1, 7, 2)      # range(nuo, iki, žingsnis)
[1, 3, 5]
>>> from numpy import arange
>>> range(1, 3, 0.5)
array([ 1. ,  1.5,  2. ,  2.5])
```

- python.org: Sekos (list, tuple, range)

```
>>> set([3, 1, 2, 1, 3])
set([1, 2, 3])
>>> # <= - poaibis
>>> # | - sąjunga
>>> # & - sankirta
>>> # - - skirtumas
>>> # ^ - sąjungos ir sankirtos skirtumas
>>> s = set([1, 2, 3])
>>> s.add(4)
set([1, 2, 3, 4])
>>> s.update([5, 6])
set([1, 2, 3, 4, 5, 6])
>>> s.remove(1)
set([2, 3, 4, 5, 6])
```

- python.org: Aibės

```
>>> ids = {'tom': 123, 'bob': 242}
>>> ids = dict(tom=123, bob=242)
>>> ids['jon'] = 324
{'bob': 242, 'jon': 324, 'tom': 123}
>>> ids.has_key('tom')
True
>>> ids['tom']
123
>>> ids.get('Tom') # Nesukelia klaidos jei nerastas
>>> ids.keys()
['bob', 'jon', 'tom']
>>> ids.values()
>>> {x: x**2 for x in range(1, 5)}
{1: 1, 2: 4, 3: 9, 4: 16}
>>> ids.items()
[('bob', 242), ('jon', 324), ('tom', 123)]
```

Sąlygos sakinyys

```
>>> if x < 0:
...     x = 0
... elif x < 8:
...     x += 1
... else:
...     x = 10
```

```
>>> for v in [None, 0, 1]:
>>>     if not v:
>>>         print('bool({0}) == False'.format(v))
>>>     if v is None:
>>>         print(v, 'is None')
bool(None) == False
None is None
bool(0) == False
```

- python.org: Sąlygos sakinyys

```
>>> for w in words:
...     if len(w) > 2:
...         continue
...     else:
...         break
... else:
...     print 'loop finished not with break statement'

>>> [i for i in range(10) if i % 3]

>>> while i < 20:
...     i += 1
```

- python.org: ciklo sakiniai


```
>>> def f(a, b):  
...     return a + b  
>>> f(a=1, b=2)  
>>> def f(a, b, *args, **kwargs):  
...     print args, kwargs
```