

Programavimo kalba **Python**

antroji paskaita

Marius Gedminas
<mgedmin@b4net.lt>

<http://mg.b4net.lt/python/>





Sintaksė



Vienas sakinytis vienoje eilutėje

```
x = 2
```

```
y = 3
```

```
z = x + y
```

```
print x, "+", y, "=", z
```



Blokai skiriami postūmio gyliu

```
x = int(raw_input("Įveskite x: "))  
y = int(raw_input("Įveskite y: "))  
if x == y:  
    print "Koks sutapimas!"  
    print "x == y"  
    print "kas galėjo pagalvoti?"
```

Daugumai tai atrodo keista
po 20 minučių įpranti



Priskyrimo sakinyys

`x = 42`

`y = 'abc'`

`z = str(x) + y`

`i = j = k = 0`

`foo, bar = bar, foo`



Reiškiniai

`-b + c * d`

`(x ** 2 + y ** 2) ** 0.5`

`'Mano' + '\n' + "vardas"`

`len('lia lia')`

`'kva kva'.upper()`



Reiškiniai sąrašai

`[1, 2, 3, 4]`

`[1, 2] + [3, 4]`

`a_list[5]`

`a_list[i:j]`

`a_list[:3]`

`a_list[-3:]`



Reiškiniai žodynai

```
{'jonas': 2, 'petras': 3}
```

```
balai['jonas']
```

```
balai.get('petraitis', 0)
```

```
'jonas' in balai
```

```
'petras' not in balai
```

balai['jonas', 'matematika']

balai['jonas', 'informatika']

raktas == ('jonas', 'matematika')

balai[raktas]



Sąrašai ir žodynai

```
balai = []  
balai.append(4)  
balai += [5, 6, 1, 2, 3]  
print max(balai)  
balai.sort()  
print ', '.join(balai[-3:])
```

```
balai = {}
```

```
balai['jonas'] = 5
```

```
balai['petras'] = 6
```

```
print balai.keys()
```

```
print balai.values()
```

```
balai['aš'] = balai.get('aš', 0) + 1
```



Išvedimas į ekraną

```
print 'Labas!'
print 'Mano', 'vardas'
print 'x =', x
print 'x =',
print x
```



Skaitymas iš klaviatūros

```
x = raw_input("Įvesk ką nors: ")
```

```
# Nedarykite taip:
```

```
y = input("Sugadink ką nors: ")
```



Darbas su failais


```
input = open('input.txt')
output = open('output.txt', 'w')
for line in input:
    print >> output, line
input.close()
output.close()
```



Sąlyginis sakinyys

```
if 1 <= x <= n:  
    print "valio"  
else:  
    print "ne valio"
```

```
cmd = raw_input("> ").strip()
if cmd == 'shout':
    print "Valio!!!!!"
elif cmd == 'wait':
    time.sleep(1)
else:
    print "Huh?"
```



Ciklai

```
while True:  
    cmd = raw_input("> ").strip()  
    if cmd == 'quit':  
        break
```

```
for line in file('/etc/passwd'):  
    print line,
```

```
for i in range(0, 1000, 10):  
    print i,
```




Funkcijos

```
def print_password(username):  
    for line in file('/etc/passwd'):  
        fields = line.split(':')  
        if fields[0] == username:  
            print fields[1]
```



Moduliai

```
import sys
```

```
print >> sys.stderr, "Klaida!"
```

```
# mano_modulis.py
import time

def visatos_prasme():
    time.sleep(236677140000000)
    return 42
```

```
# mano programa
```

```
from mano_modulis import \  
    visatos_prasme  
print visatos_prasme()
```



Kuo skirias modulis nuo programos?

```
# mano_modulis.py
import time

def visatos_prasme():
    time.sleep(236677140000000)
    return 42

if __name__ == '__main__':
    print "Testuojų..."
    assert visatos_prasme() == 42
```




Paketai



```
from zope.interface import Interface
```



Katalogas su `__init__.py` failu



Klasės

```
from math import sqrt  
  
class Vector(object):  
  
    def __init__(self, x, y):  
        self.x = x  
        self.y = y  
  
    def length(self):  
        return sqrt(self.x**2 +  
                    self.y**2)
```



Objektai

```
zero = Vector(0, 0)  
assert zero.length() == 0.0
```



Viskas yra objektai

Skaičiai

Simbolių eilutės

Įvairios kitos reikšmės

Funkcijos

Moduliai

Klasės

Bet kokį objektą galima priskirti
kintamajam

```
zero = Vector(0, 0)
```

```
one = Vector(1, 0)
```

```
fn1 = zero.length
```

```
fn2 = one.length
```

```
print fn1(), fn2()
```



uff...

laikas pailsėti