



TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Programmeerimise süvendatud algkursus ITI0140

2015



# Teemad

- Koodistiil
- Koodi kommenteerimine
- Eclipse IDE ja silumine
- Ülesanne



# Harjutamiseks

**<http://www.pyschools.com/>**

Pythoni keelespetsiifilised harjutused

**<http://codingbat.com/python>**

Lühikesed harjutused ülesannete lahendamise harjutamiseks (algoritmide loomine)

**<https://www.codecademy.com/tracks/python>**

Kursuse ülesehitusega (teemad eraldi plokkides)



# Koodistiil

**PEP8** Style Guide for Python Code

**<http://www.python.org/dev/peps/pep-0008/>**

Kui te tahate ise oma koodi PEP8 vastavust kontrollida, võite kasutada tööriista flake8

**<https://pypi.python.org/pypi/flake8>**



# Tähtsamad PEP8 ivad

Kasuta **4 tühikut ühe taande** kohta. Tabulaatorit ei ole lubatud kasutada.

Ära kirjuta ühele reale üle **79 tähemärgi**.

Lähtekood kirjuta **inglise keeles**. (Kunagi ei tea, kes sinu koodi tulevikus peab lugema – harjuta see tava endale sisse juba varakult)

**Muutujate ja funktsioonide nimed on kõik väiketähtedega ja eraldatud alakriipsudega (*example\_variable*).**

Täpikähti ei saa kasutada muutujanimeses Python 2-s (Python 3 toetab, aga ei soovita “rahvusvahelisuse” huvides)

*Defining Python Source Code Encodings*  
<http://www.python.org/dev/peps/pep-0263/>



# Docstring

**PEP257** Docstring Conventions

<https://www.python.org/dev/peps/pep-0257/>

Kui tahate ise enne Git-i panemist kontrollida, võite installida PEP257 tööriista: <https://pypi.python.org/pypi/pep257>

**Näide:**

```
def complex(real = 0.0, imag = 0.0):  
    """Form a complex number.  
  
    Keyword arguments:  
    real -- the real part (default 0.0)  
    imag -- the imaginary part (default 0.0)  
    """  
    if imag == 0.0 and real == 0.0:  
        return complex_zero  
    ...
```

See kommentaar  
ongi “docstring”





# Eclipse

Kasutame **Eclipse** ja **PyDev** lisapaketti

Kes tahab programmitöö visualiseerimist võib proovida kasutada:

**<http://thonny.cs.ut.ee/>**

(Soovitan meeles pidada just rekursioonil põhinevate ülesannete jaoks).



# Breakpoint

Topeltklikk või  
paremklikk ja  
“Add  
Breakpoint”

Kui nüüd F11  
(või Run –  
Debug) või  
sitika ikooniga  
nuppu  
vajutada, siis  
peatatakse  
programmi  
töö kui selle  
breakpointini  
jõutakse

```
1 """
2   HackerCup something >_<
3
4   @author: gent
5   """
6
7 def exchange_two_numbers_and_create_min_and_max_values(n):
8     r = []
9     for i in range(2):
10        PyDev breakpoint str(n)
11        s = sorted(l, reverse=True if i == 0 else False)
12        for j in range(len(l)):
13            if l[j] != s[-1]:
14                break
15
16        z = 0
17        if s[-(1 + j)] == "0" and j == 0 and len(l) > 1:
18            z = 1
19        l[(len(l) - 1 - l[::-1].index(s[-(1 + j)]))] = l[j + z]
20        l[j + z] = s[-(1 + j)]
21        r.append(str(" ".join(l)))
22    return " ".join(r)
23
24 import unittest
25 class MinMaxTests(unittest.TestCase):
26     def test_1(self):
27         self.assertEqual(exchange_two_numbers_and_create_min_and_max_values(10), "10 10")
28     def test_2(self):
29         self.assertEqual(exchange_two_numbers_and_create_min_and_max_values(7276), "2776 7762")
30     def test_3(self):
31         self.assertEqual(exchange_two_numbers_and_create_min_and_max_values(7267), "2767 7762")
32     def test_4(self):
33         self.assertEqual(exchange_two_numbers_and_create_min_and_max_values(7272), "2277 7722")
34     def test_5(self):
35         self.assertEqual(exchange_two_numbers_and_create_min_and_max_values(31524), "13524 51324")
36     def test_6(self):
37         self.assertEqual(exchange_two_numbers_and_create_min_and_max_values(897), "798 987")
38     def test_7(self):
39         self.assertEqual(exchange_two_numbers_and_create_min_and_max_values(123), "123 321")
40     def test_8(self):
41         self.assertEqual(exchange_two_numbers_and_create_min_and_max_values(5), "5 5")
42     def test_9(self):
```

Console

<terminated> F:\Workspace\Python33\swap.py

Ran 18 tests in 0.002s

OK



Resume (F8)

Step Into (F5) – Funktsiooni sisse astumine

Step Over (F6) – Funktsioonist üle astumine

Step Return (F7) – Funktsioonist väljumine

Terminate

Siin on näha muutujate väärtused

Debug - Python33/swap.py - Eclipse

File Edit Source Refactoring Refactor Navigate Search Project PyDev Run Window Help

Quick Access Java Prolog PyDev Debug

Debug

MainThread - pid4752\_seq1

- exchange\_two\_numbers\_and\_create\_min\_and\_max\_values [swap.py:10]
- test\_1 [swap.py:26]
- \_executeTestPart [case.py:384]
- run [case.py:439]
- \_\_call\_\_ [case.py:491]
- run [suite.py:105]
- \_\_call\_\_ [suite.py:67]
- run [suite.py:105]
- \_\_call\_\_ [suite.py:67]
- run [runner.py:168]
- runTests [main.py:265]
- \_\_init\_\_ [main.py:125]
- <module> [swap.py:68]
- execfile [pydev\_execfile.py:18]
- run [pydevd.py:1547]
- <module> [pydevd.py:2090]

Programmi järg

(x) Variables Breakpoints

Name	Value
Globals	Global variables
▶ i	int: 0
▶ n	int: 10
▶ r	list: []

```
1 """
2 Hackercup something ><
3
4 @author: gerit
5 """
6
7 def exchange_two_numbers_and_create_min_and_max_values(n):
8     r = []
9     for i in range(2):
10        l = list(str(n))
11        s = sorted(l, reverse=True if i == 0 else False)
12        for j in range(len(l)):
13            if l[j] != s[-1]:
14                break
15            z = 0
16            if s[-(1 + j)] == "0" and j == 0 and len(l) > 1:
17                z = 1
18            l[len(l) - 1 - l[::-1].index(s[-(1 + j)])] = l[j + z]
19            l[j + z] = s[-(1 + j)]
20            r.append(str("".join(l)))
21        return " ".join(r)
22
23 import unittest
24 class MinMaxTests(unittest.TestCase):
25     def test_1(self):
26         self.assertEqual(exchange_two_numbers_and_create_min_and_max_values(10), "10 10")
```

Console

swap.py

pydev debugger: starting (pid: 4752)

Outline

- exchange\_two\_numbers\_and\_create\_min\_and\_max\_values
- unittest
- MinMaxTests
- main
- \_\_main\_\_

Writable Insert 10:1



# Eclipse

Mõned kasulikud lühendid, mida meeles pidada

Ctrl+S

Salvesta fail

Ctrl+Shift+F

Format code (automaatselt parandab mõningaid vigu)

Ctrl+F11

Run

F11

Debug



# Programmi ülesehitus

```
"""  
Solve some problem using some clever algorithm  
  
@author: gert  
"""  
  
def some_function(argument):  
    """Solve a part of the bigger problem.  
  
    Arguments:  
    argument -- An argument for the function  
  
    Returns:  
    True if successful  
    """  
    return True  
  
if __name__ == "__main__":  
    some_function(True)
```

See konstruktsioon tagab, et teised Pythoni programmid saavad seda faili turvaliselt importida



# Ülesanne

Ülesanne on nähtaval

- <https://ained.ttu.ee>
- <https://courses.cs.ttu.ee/pages/ITI0140>