

lstlistingchapter

# Metaohjelmointi Python-kielellä

Mikko Koho

Seminaarityö

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Helsinki, 14. marraskuuta 2014

Tiedekunta — Fakultet — Faculty		Laitos — Institution — Department	
Matemaattis-luonnontieteellinen		Tietojenkäsittelytieteen laitos	
Tekijä — Författare — Author			
Mikko Koho			
Työn nimi — Arbetets titel — Title			
Metaohjelmointi Python-kielellä			
Oppiaine — Läroämne — Subject			
Tietojenkäsittelytiede			
Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages	
Seminaarityö	14. marraskuuta 2014	5	
Tiivistelmä — Referat — Abstract			
<p>Tiivistelmä.</p> <p>ACM Computing Classification System (CCS):</p>			
Avainsanat — Nyckelord — Keywords			
Python, metaohjelmointi			
Säilytyspaikka — Förvaringsställe — Where deposited			
Muita tietoja — Övriga uppgifter — Additional information			

# Sisältö

<b>1 Johdanto</b>	<b>1</b>
<b>2 Python yleisesti</b>	<b>1</b>
2.1 Pythonin syntaksi . . . . .	1
2.2 Operaattorit . . . . .	2
2.3 Muuttujat . . . . .	2
2.4 Muuttujatyyppejä . . . . .	2
2.5 Iteroitavat . . . . .	2
<b>3 Python-kielen metaohjelmointimaisia komponentteja</b>	<b>3</b>
3.1 Introspektio . . . . .	3
3.2 Metaluokat . . . . .	4
3.3 Generaattorit . . . . .	4
3.4 Lausekkeet . . . . .	4
<b>4 Python-metaohjelmointi laajemmin</b>	<b>4</b>
4.1 Reflektio . . . . .	4
4.2 Ohjelman ajonaikainen muokkaus . . . . .	4
<b>5 Yhteenveto</b>	<b>4</b>
<b>Lähteet</b>	<b>5</b>

# 1 Johdanto

TODO: Johdanto. (Metaohjelmointi, Python)

## 2 Python yleisesti

Ensimmäinen Python-kielen versio julkaistu 1991 [TODO: lähde]. Python-kielistä on nykyään käytössä eri versioita. Python 2.7 on edelleen melko suosittu vaikka versio 3 on julkaistu jo 2008. Versio 3 ei ole yhteensopiva aiempien versioiden kanssa. Version 2:n suosion taustalla on se, että monet suositut kirjastot ja sovelluskehikset eivät ole siirtyneet versioon 3. Tämän tekstin esimerkit toimivat sekä Python 2.7:llä että Python 3:lla, ellei toisin mainita.

Pythonin suosio on kasvanut tasaisesti ja se on nykyään käytetyin kieli ohjelmoinnin perusteiden opetukseen Yhdysvaltojen yliopistoissa [Guo14].

Python-ohjelmakoodia voidaan kääntää useilla eri kääntäjillä [Mar06]. Käytetyin kääntäjä on CPython (Classic Python), joka kääntää alkuperäisen koodin Python-tavukoodiksi. Tavukoodia ajetaan C-kielillä toteutetulla virtuaalikoneella [Mar06]. Standardikirjasto on toteutettu osittain C:llä ja osittain Pythonilla.

Muita suosittuja kääntäjiä ovat Java-tavukoodiksi kääntävä Jython sekä IronPython, joka kääntää Python-koodia .NET-ympäristön käyttämäksi CIL-tavukoodiksi. PyPy on Python-kielillä toteutettu useissa eri ympäristöissä toimiva suoraan konekielille koodia kääntävä ajonaikainen (just-in-time) kääntäjä. PyPy on toteutettu RPython-kielillä, joka on Python-kielen osajoukko.

Tässä luvussa käydään läpi Python-kielen perusteet.

### 2.1 Pythonin syntaksi

Python ohjelma koostuu loogisista riveistä, jotka ovat yhden tai useamman ”fyysisen” rivin mittaisia. [Mar06]. Loogisten rivien päättämiseen ei käytetä mitään merkkiä. Rivien sisennyksen perusteella erotetaan ohjelmakoodin lohkot toisistaan. Suositeltu tapa sisentää on käyttää ensimmäisen tason sisentämiseen 4 välilyöntiä ja seuraavaan 8 ja niin edelleen [VWC13].

## 2.2 Operaattorit

## 2.3 Muuttujat

Python on dynaamisesti tyyppitetty kieli eli muuttujien arvon tyyppiä ei tarvitse eksplisiittisesti määrittää vaan tyyppi määräytyy sen perusteella minkälainen arvo muuttujaan sijoitetaan. Muuttujan tyyppiä voi myös vaihtaa sijoittamalla siihen uuden eri tyyppisen arvon. Listaus 2.3 sisältää yksinkertaisen esimerkin Python-kielen syntaksista.

```
# -*- coding: utf-8 -*-  
  
import sys  
  
a = 'Hello world!'  
print(a)  
# Hello world!  
  
a = len(a)  
  
print(a)  
# 12
```

Listing 1: Yksinkertainen esimerkki Python-kielen syntaksista.

Pythonissa kaikki arvot ovat olioita. Olion tyyppi määrittää mitä metodeja ja ominaisuuksia olio tarjoaa. Osa olioista on muuttumattomia (immutable) ja osa muutettavina (mutable).

## 2.4 Muuttujatyyppejä

## 2.5 Iteroitavat

Listakehitelmä (list comprehension), joukkokehitelmä (set comprehension) ja sanakirjakehitelmä (dictionary comprehension) ovat tapoja luoda lista-, joukko- tai sanakirjaolioita.

Esimerkki listakehitelmän käytöstä sekä joistain Python-kielen funktioista

on listauksessa 2.5. Funktio `range` palauttaa Python 2:ssa listan ja Python 3:ssa generaattoriolion, jota voidaan käyttää listan tapaan. Funktio `all` tarkastaa kaikkien listan (tai muun iteroitavan olion) totuusarvon. Pythonissa kaikki oliot voidaan evaluoida totuusarvoina, jolloin lukujen tapauksessa aina luku "0" evaluoituu epätodeksi ja muut luvut todeksi.

```
print(range(10))
# [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

print(range(0, 30, 3))
# [0, 3, 6, 9, 12, 15, 18, 21, 24, 27]

print( [x**2 for x in range(11)] )
# [0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

some_list = [1, 4, 7, 'foo', 12, 'bar']
print( [x for x in some_list if str(x).isalpha()] )
# ['foo', 'bar']

# Alkulukuja
print(
    [x for x in range(2, 50) if all( [x % y for y in range(2, x-1)] )]
)
# [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]
```

Listing 2: Esimerkki funktion `range` käytöstä ja listakehitelmistä.

## 3 Python-kielen metaohjelmointimaisia komponentteja

### 3.1 Introspektio

Introspektiolla tarkoitetaan tietojen hakemista muistissa olevista olioista, moduuleista ja funktioista [Pil04].

TODO: esimerkki kaikista perusjutuista: `type`, `dir()`, `__doc__`  
reflektio (`dir()`, `__luokat__`).  
[Pil04]

### **3.2 Metaluokat**

### **3.3 Generaattorit**

Yield on metaohjelmointia.

### **3.4 Lausekkeet**

Kuorruttajat(?) (decorator) [Dub05].  
Osittainsovellus (partial application).

## **4 Python-metaohjelmointi laajemmin**

Exec. Eval. Compile.

Käännösaikainen metaohjelmointi. Löytyykö template-kieliä?

### **4.1 Reflektio**

In computer science, reflection is the ability of a computer program to examine (see type introspection) and modify the structure and behavior (specifically the values, meta-data, properties and functions) of the program at runtime.

### **4.2 Ohjelman ajonaikainen muokkaus**

metodien korvaaminen toisilla osoittimilla muuttamalla. Monkey patching.

## **5 Yhteenveto**

Yhteenveto.



## Lähteet

- [Dub05] Dubois, P. F.: *A nest of Pythons*. Computing in Science & Engineering, 7(6):81–84, 2005.
- [Guo14] Guo, P.: *Python is Now the Most Popular Introductory Teaching Language at Top U.S. Universities*. Communications of The ACM Blog, heinäkuu 2014. <http://cacm.acm.org/blogs/blog-cacm/176450/fulltext> [ 06.11.2014 ].
- [Mar06] Martelli, A.: *Python in a Nutshell*. O'Reilly Media, Inc., 2006.
- [Pil04] Pilgrim, M.: *Dive Into Python*, toukokuu 2004. <http://www.diveintopython.net/> [ 06.11.2014 ].
- [VWC13] Van Rossum, G., Warsaw, B. ja Coghlan, N.: *PEP 8 – Style guide for python code*. 2013. <http://www.python.org/dev/peps/pep-0008> [ 11.11.2014 ].