

Python Programlama Dili

DOCTOR FUN

6 Apr 2000



Copyright © 2000 David Farley, d-farley@metalab.unc.edu
<http://metalab.unc.edu/Dave/drfun.html>

This cartoon is made available on the Internet for personal viewing only. Opinions expressed herein are solely those of the author.

R. Emre Başar

emrebasar@member.fsf.org

5 Ocak 2005

Tarihçe

- Guido Van Rossum, 1991
- 1991 – 1995 SMC, Amsterdam
- 1995 – 2001 CNRI
- 2001 – 2005 PSF

Yılan?

Monty Python, BBC

*Programlamaya bakışınızı değiştirmeyen
bir programlama dili, öğrenmeye değer.*

Anonim (?)

Python'un Kökenleri

→ ~~C/C++~~

→ ABC

→ SmallTalk

→ LISP

→ OOP

→ Fonksiyonel

→ Prosedürel

Python Felsefesi

- Tao Of Python
- Herşey bir nesnedir
- Herşey taklit edilebilir
- Okunabilirlik önemlidir
- Piller Dahildir

Yazım Tarzı

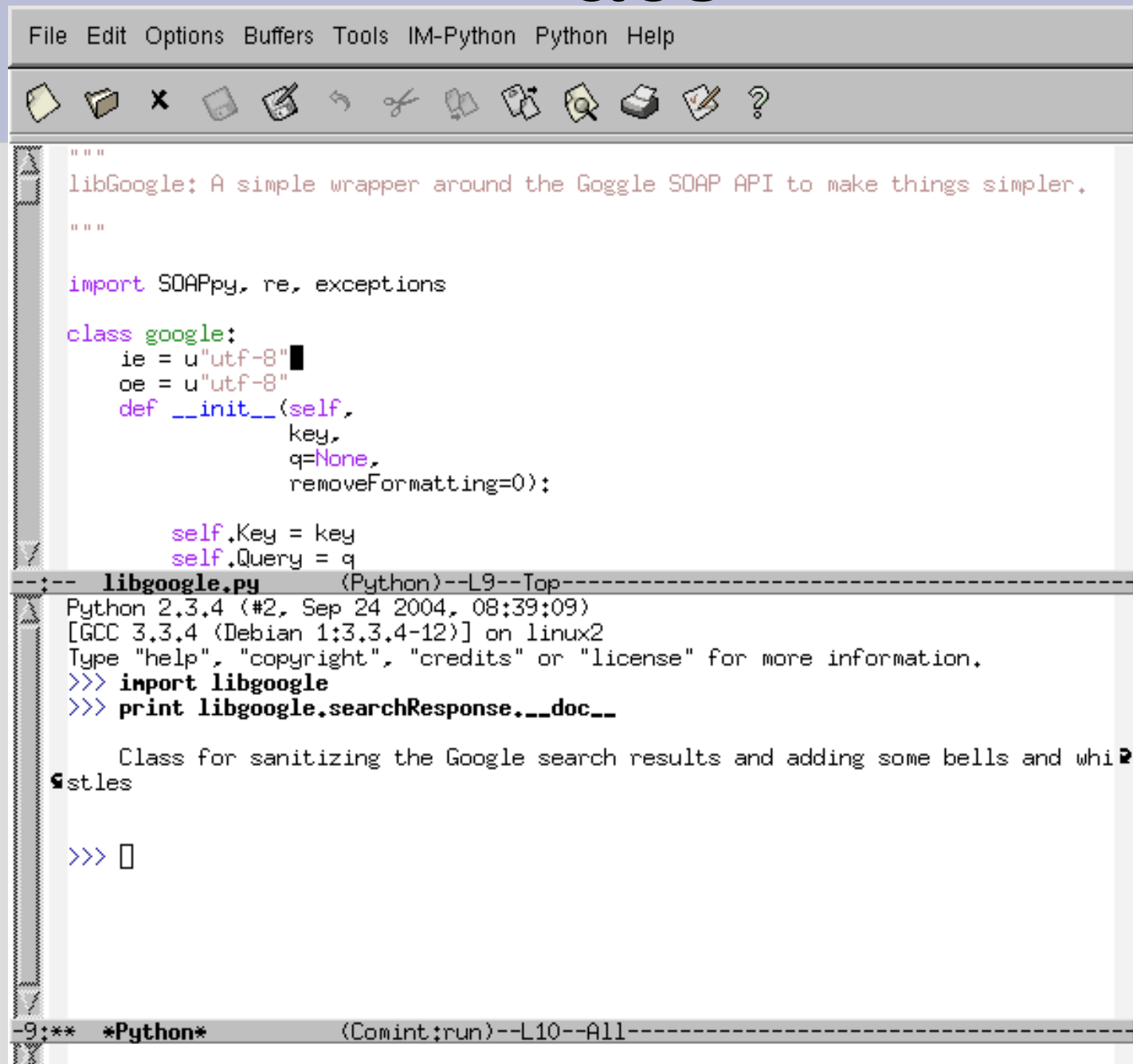
```
if blablabla:
    sunu_yap()
    bunu_yap(boyle_yap)
elif hede:
    hodo()
else:
    hicbiri_tutmazsa_bunu_yap(valla)

for foo in bar:
    print foo
```

IDE'ler

- **emacs** <http://www.gnu.org/software/emacs/>
- **Boa Constructor** <http://boa-creator.sf.net/>
- **DrPython** <http://drpython.sf.net/>
- **IDLE** <http://www.python.org/>
- **vim** <http://www.vim.org>

Emacs



The screenshot shows the Emacs text editor interface. The menu bar at the top includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'IM-Python', 'Python', and 'Help'. Below the menu bar is a toolbar with various icons for file operations and editing. The main window displays a Python script named 'libgoogle.py'. The script contains a docstring, imports, and a 'google' class with an '__init__' method. Below the script, the Emacs console shows the output of running the script, including the Python version, GCC version, and the result of importing 'libgoogle' and printing the docstring of 'searchResponse'.

```
File Edit Options Buffers Tools IM-Python Python Help

libGoogle: A simple wrapper around the Goggle SOAP API to make things simpler.

import SOAPpy, re, exceptions

class google:
    ie = u"utf-8"
    oe = u"utf-8"
    def __init__(self,
                  key,
                  q=None,
                  removeFormatting=0):

        self.Key = key
        self.Query = q

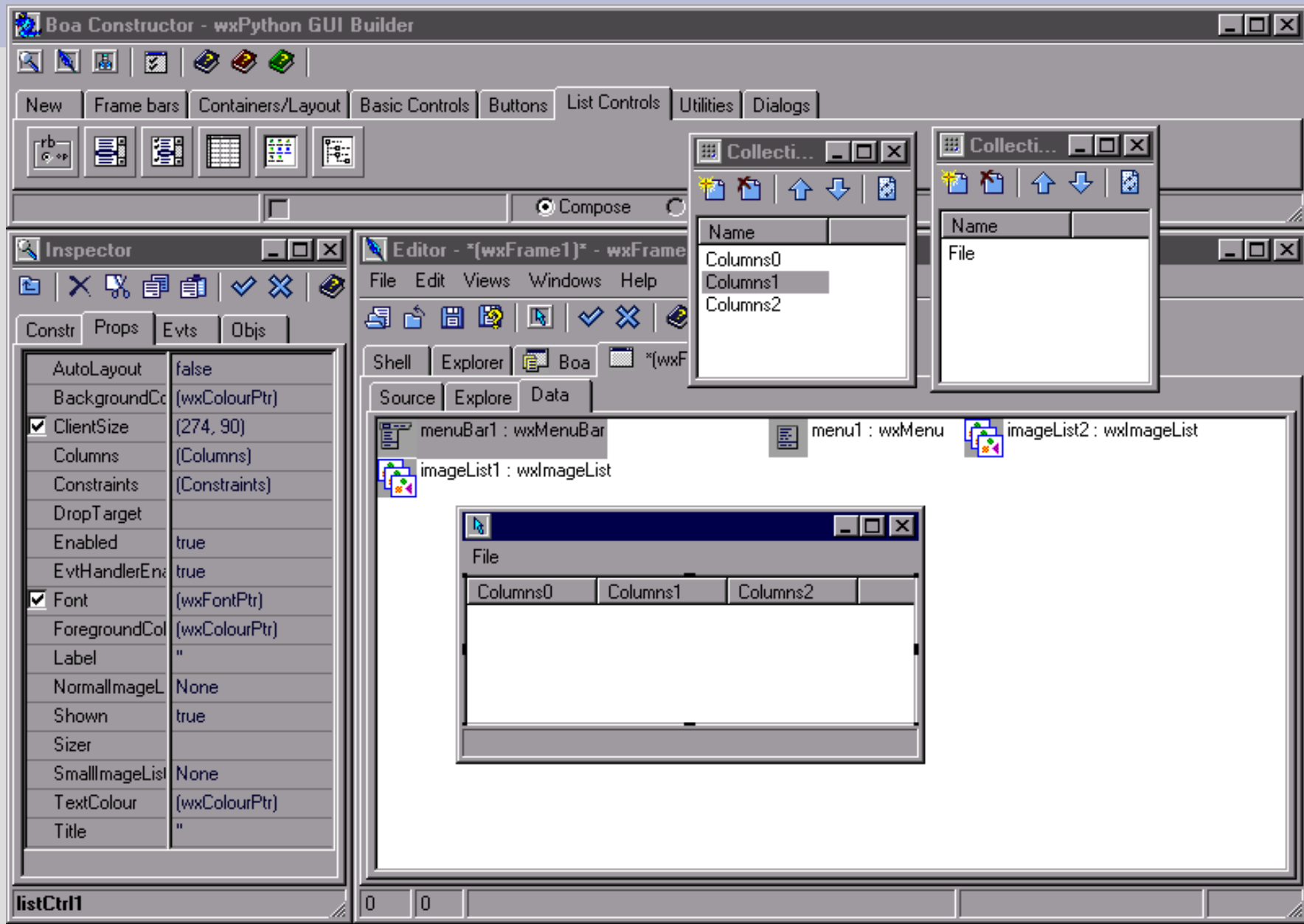
--:-- libgoogle.py (Python)--L9--Top-----
Python 2.3.4 (#2, Sep 24 2004, 08:39:09)
[GCC 3.3.4 (Debian 1:3.3.4-12)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import libgoogle
>>> print libgoogle.searchResponse.__doc__

Class for sanitizing the Google search results and adding some bells and whistles

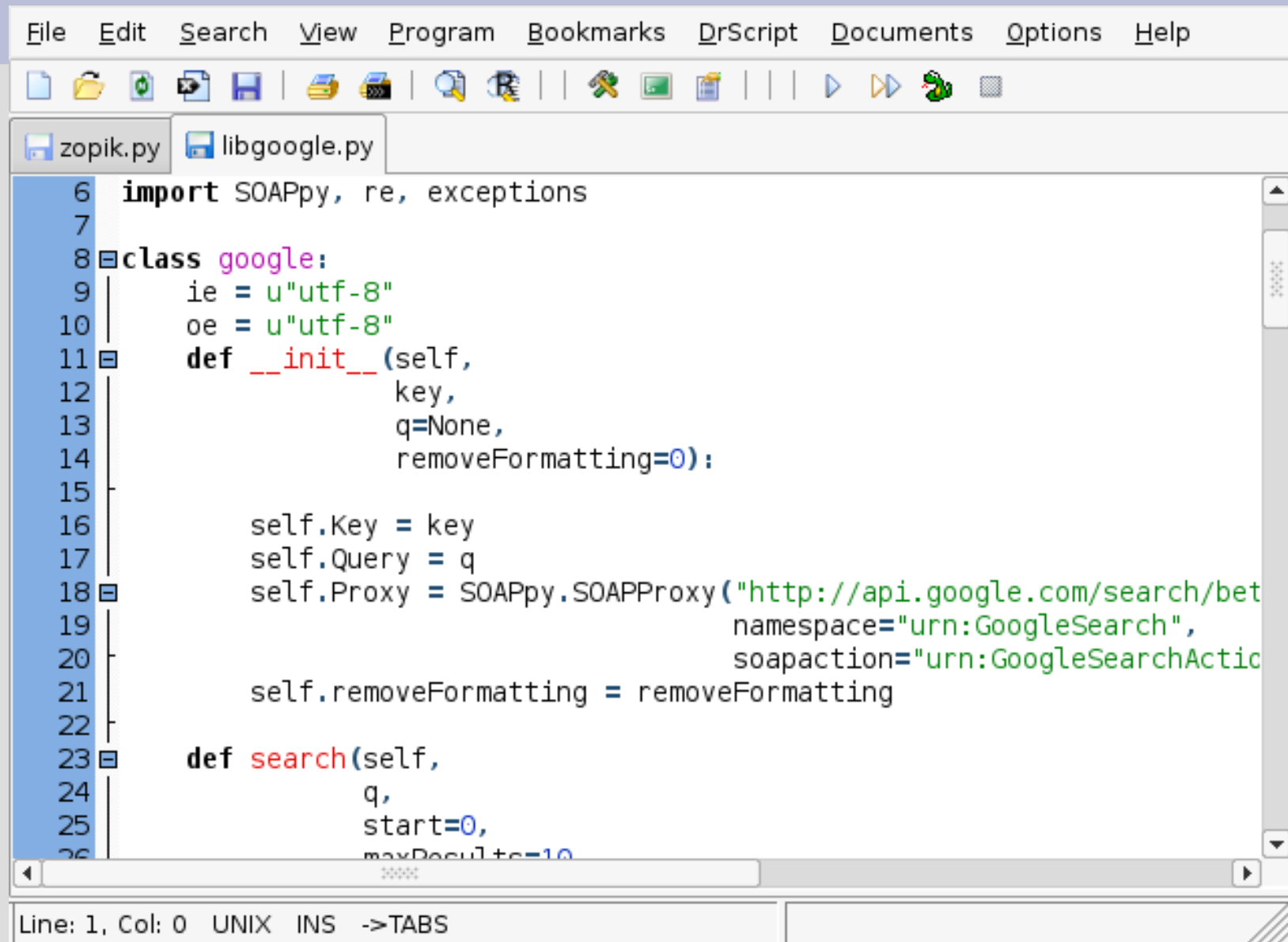
>>>
```

-9:** *Python* (Comint:run)--L10--All-----

Boa Constructor



DrPython



```
File Edit Search View Program Bookmarks DrScript Documents Options Help
zopik.py libgoogle.py
6 import SOAPpy, re, exceptions
7
8 class google:
9     ie = u"utf-8"
10    oe = u"utf-8"
11    def __init__(self,
12                key,
13                q=None,
14                removeFormatting=0):
15
16        self.Key = key
17        self.Query = q
18        self.Proxy = SOAPpy.SOAPProxy("http://api.google.com/search/bet
19                                     namespace="urn:GoogleSearch",
20                                     soapaction="urn:GoogleSearchActio
21
22        self.removeFormatting = removeFormatting
23    def search(self,
24              q,
25              start=0,
26              maxResults=10
```

Line: 1, Col: 0 UNIX INS ->TABS

Python'un Güzellikleri

- lambda formları
- map(), filter(), reduce()
- List Comprehensions
- Üreteçler (Generators)
- Yerel Fonksiyonlar
- İç bakış (Introspection)

lambda (λ)

- İsimsiz fonksiyonlar
- Kısa, LISP ile karşılaştırıldığında güçsüz.
- Fonksiyonel programlama'nın izi

```
lambda x: x*x
```

```
>>> def artirici_uret(kaclik):  
...     return lambda x: x + kaclik  
>>> beser_beser_artir = artirici_uret(5)  
>>> beser_beser_artir(10)  
15
```

map(), filter() ve reduce()

`map()` : Bir fonksiyonu listenin tüm elemanlarına uygular

```
map(lambda x: x*x, range(10))  
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

`filter()` : Bir listenin elemanlarını belirtilen kurala göre eler

```
filter(lambda x: x%2, range(10))  
[1, 3, 5, 7, 9]
```

`reduce()` : listenin elemanları üzerinde her elemanı bir önceki işlemin sonucuyla birlikte bir fonksiyona parametre şeklinde geçirerek dolaşır

```
reduce(lambda x,y: x+y, range(10), 0)  
45
```

List Comprehensions

- map() ve filter()'ın yerine geçer
- Kendine özgü yazım tarzı
- Okunabilirliği artırır

```
[x*x for x in range(10) if x%2]  
[1, 9, 25, 49, 81]
```

List Comprehensions Olmasaydı?

```
def tek_sayıların_kare_listesi():  
    kare_listesi = []  
    for x in range(10):  
        if x%2:  
            kare_listesi.append(x*x)  
    return kare_listesi
```

Üreteçler (Generators)

- Durumunu koruyan fonksiyonlar
- `return` yerine `yield`
- Bellek tasarrufu
- Verinin tek yönlü işlenmesine uygun
- Öncelikli hedef web uygulamaları

```
def jenerator():  
    yield 1  
    yield 2  
j = jenerator()  
j.next()  
1  
j.next()  
2  
j.next() --> StopIteration
```

Yerel Fonksiyonlar

- Lambda'nın adsız olmayanı
- İstenildiği kadar karmaşık olabilir
- Sadece ait olduğu fonksiyona görünür
- Fonksiyon kirliliğini önler

```
def esas_fonksiyon(x):  
    def alt_fonksiyon(n):  
        """Bir fonksiyon"""  
        return x * n  
    def docstring_yazdir(f):  
        return f.__doc__  
    print docstring_yazdir(alt_fonksiyon)  
    return alt_fonksiyon
```


İç Bakış (Introspection)

- `dir()`
- `globals()`, `locals()`
- `hasattr()`, `getattr()`, `setattr()`
- Çalışma anında kodun durumunu inceleyebilmeyi sağlar
- Kod kendini o an içerisinde bulunduğu duruma göre biçimlendirebilir
- `import` vb... modüllerin yardımıyla sınırsız esneklik

```
print '\n---\n'.join([getattr(sys, f).__doc__  
                        for f in dir(sys)  
                        if hasattr(getattr(sys, f),  
                                   '__call__')])
```

Özel Adlar

- Veri tiplerinin taklidini sağlayan özel fonksiyonlar
- Nesneler hakkında bilgi sağlayan özel değişkenler
- `'__'` ile başlar `'__'` ile biterler
- `__add__`, `__mul__`, `__div__`
- `__cmp__`, `__call__`
- `__getattr__`, `__hasattr__`
- `__getitem__`, `__setitem__`
- `__doc__`, `__name__`, `__file__`, `__module__`
- Tam liste: <http://docs.python.org/ref/specialnames.html>

Python ile geliştirilmiş uygulamalar

- Zope
- BitTorrent
- Portege
- pyblosxom

Web Uygulamalarında Python

- Zope
- Twisted
- mod_python
- cherrypy

Diğer Python İmplementasyonları

→ CPython

- C ile geliştirilmiş
- Orjinal Python

→ JPython

- Java ile geliştiriliyor
- Java ile Python entegrasyonu
- Çift taraflı

→ IronPython

- .Net ile entegre python yorumlayıcısı
- C# ile geliştiriliyor
- .Net kütüphanesine erişebiliyor
- Microsoft.Net ve Mono altında çalışıyor
- Henüz yeni olmasına rağmen hızlı

Kaynaklar

Python Dökümantasyonu

<http://www.python.org/doc/>

Dive Into Python

<http://www.diveintopython.org/>

Mustafa Başer, Python

Pusula Yayınları, 2002

Dinçer Aydın, Python Kılavuzu

<http://www.belgeler.org/uygulamalar/python-tutorial.html>

Sorular?

Teşekkürler