

PYTHON FOR HACKERS

Mazlum Ağar (info@mazlumagar.com)

20 Şubat

Hackercamp 2016

EDUC Bilgisayar mühendisliği, Sakarya Üniversitesi

WORK Siber güvenlik, PRODAFT

IDEA Özgür yazılım ve açık kaynak tutkunu

MEMB Octosec

SOCIAL @mazlumagar

PYTHON

1990 yılında Guido van Rossum tarafından geliştirilmeye başlanmıştır. İsmi **Monty Python's Flying Circus** isimli gösteriden almıştır. Sadece bir programlama dili olmanın ötesine geçen nadir dillerdendir. Özellikle **Hacker** kültürünün **Perl**'den sonra sahiplenmesiyle ayrıcalıklı bir duruma gelmiştir.

- Okunabilir söz dizimi(syntax)
- Gelişmiş kütüphane desteği
- Kolay öğrenme
- Aktif bir topluluk
- Taşınabilir uygulamalar
- Hızlı geliştirme

UÇMAK MI İSTİYORSUN?



YOU'RE FLYING!
HOW?



I LEARNED IT LAST
NIGHT! EVERYTHING
IS SO SIMPLE!

HELLO WORLD IS JUST
`print "Hello, world!"`

I DUNNO...
DYNAMIC TYPING?
WHITESPACE?

COME JOIN US!
PROGRAMMING
IS FUN AGAIN!
IT'S A WHOLE
NEW WORLD
UP HERE!



BUT HOW ARE
YOU FLYING?

I JUST TYPED
`import antigravity`
THAT'S IT?

... I ALSO SAMPLED
EVERYTHING IN THE
MEDICINE CABINET
FOR COMPARISON.



BUT I THINK THIS
IS THE PYTHON.

- Web sitesi
- Oyun
- Günlük araçlar
- Mobil uygulama
- Bilimsel çalışma ...

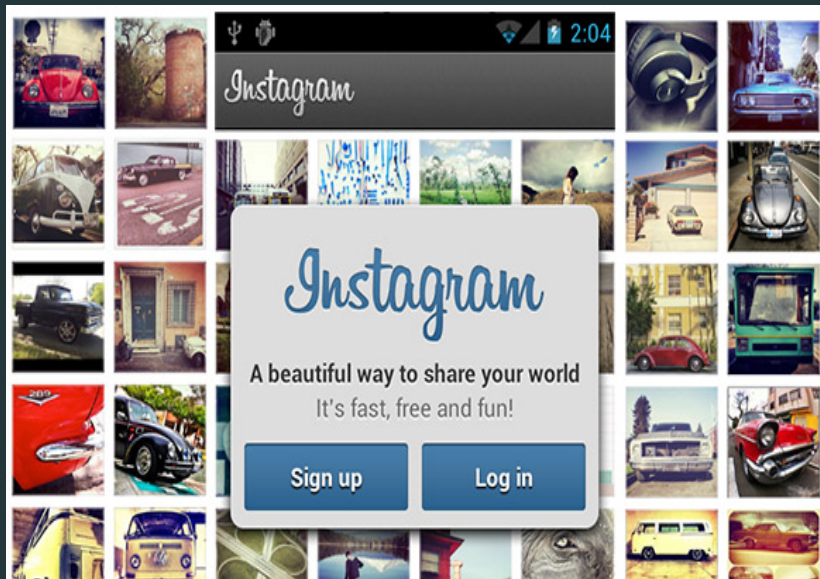
KIM KULLANIYOR?

- Google, Yahoo, NASA
- Dropbox, Disqus, Mozilla
- Friendfeed, Reddit, Eventbrite
- Walt Disney, Battlefield 2, Civilization 4
- Nokia, IBM, CIA ...

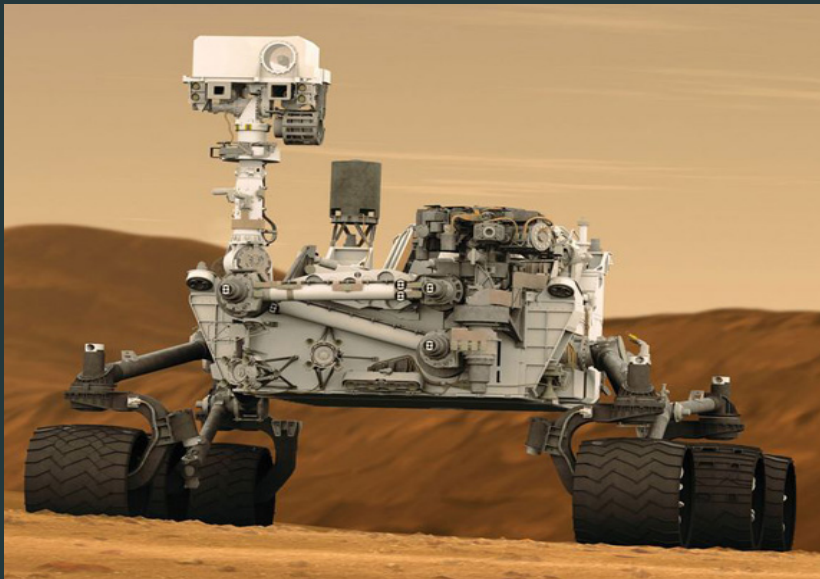
KARAYIP KORSANLARI



INSTAGRAM



YOK ARTIK?



Python'un kendi yapıları ve veri tiplerini okunabilir, temiz bir şekilde kullanarak oluşturulan ifadelere **Pythonic** denilir.

Herkes **Python** yazabilir ama sadece çok az kişi **Pythonic** kod yazabilir!

Şu anda **Python** ile uğraşmak isteyen birçok kişi hangi sürümü kullanması gerektiği konusunda kafaları karışık. Kafa karışıklığına gerek yok, ikisi de **Python**!

ENVIRONMENT

Vim python.vim, python-mode, jedi ...

Emacs auto-complete, jedi.el ...

Sublime Text anaconda, djaneiro ...

PyCharm Full Python IDE ...

```

5 d./g/index c/box.yml
p p - previous diff of selected and current
o <cr> - revert to selected state

Q [10] 3 minutes ago
o [9] 3 minutes ago
o [8] 3 minutes ago
o [7] 6 days ago
o [6] 6 days ago
o [5] 6 days ago
o [4] 6 days ago
o [3] 6 days ago
o [2] 6 days ago
o [1] 6 days ago
o [0] Original

Gundo > Undo tree
+++ 9 2013-01-25 02:40:12 am
+++ 10 2013-01-25 02:48:14 am
00 -23,12 +23,7 00
+ google_analytics/
+ google_cse/
+ mediagenerator/
+ filters/
+ _init_.pyc
+ concat.pyc
+ cssurl.pyc
+ sass.pyc
+ yuicompressor.pyc
+ filters/
+ generators/
+ management/
+ templatetags/

Gundo > Diff preview
:GStatus

+ blog/
+ caching/
+ dbindexer/
+ Disqus/
+ django/
+ djangoengine/
+ djangotoolbox/
+ docutils/
+ events/
+ google_analytics/
+ google_cse/
+ mediagenerator/
+ filters/
+ generators/
+ management/
+ templatetags/
+ _init_.pyc
+ base.pyc
+ middleware.pyc
+ settings.pyc
+ utils.pyc
+ minicms/
+ object_permission_backend/
+ permission_backend_norel/
+ plog/
+ profiles/
+ programmi/
+ pygments/
+ redirects/
+ replay/
+ templates/
+ admin/
+ replay/
+ change_form.html
+ submit_line.html
+ _init_.py
+ _init_.pyc
+ admin.py
+ admin.pyc
+ models.py
+ models.pyc
+ robots.py
+ script/
+ search/
+ simplesocial/
+ staticstuff/
+ templates/
+ urlrouter/
+ _init_.py
+ _init_.pyc
+ _generated_media_names.py
+ _generated_media_names.pyc
+ migrate.py
+ app.yaml
+ backup.sh
+ blank.html
+ cleanhtmlwriter.py
+ cleanhtmlwriter.pyc

1 # Git branch master
2 # Untracked files:
3 # (use "git add <file>..." to include in what will be committed)
4 #
5 # .dropbox.js.swp
6 nothing added to commit but untracked files present (use "git add" to track)

N p master > dropbox.js/git/index - 100% unix < utf-8 < gitcommit 16% % 1:
131 xhr.onreadystatechange = function(evt) {
132     if (this.readyState == 4 && this.status == 200) {
133         successCallback(evt);
134     }
135 };
136 xhr.overrideMimeType(data.mimeType);
137 xhr.setRequestHeader('Content-Type', data.mimeType);
138 xhr.sendAsBinary(evt.target.result);
139 }
140 reader.readAsBinaryString(data.fileObj);
141 };
142 c._getRoot = function() {
143     return this.session.accessType == 'dropbox' ? 'dropbox' : 'sandbox';
144 };
145 c._getPath = function(params) {
146     if (!params || !params.path) {
147         throw new Error("Must provide a path to get metadata from");
148     }
149     var path = params.path;
150     delete params.path;
151     return path;
152 };
153 c._getFileUrl = function(prefix, params) {
154     var root = this._getRoot();
155     var path = this._getPath(params);
156     return [prefix, root, path].join('/');
157 };
158 c.accountInfo = function(successCallback, errorCallback) {
159     this.api('account/info', {}, successCallback, errorCallback);
160 };
161 c.metadata = function(params, successCallback, errorCallback) {
162     var url = this._getFileUrl('metadata', params);
163     this.api(url, params, successCallback, errorCallback);
164 };
165 c.putFile = function(params, successCallback, errorCallback) {
166     params.overwrite = params.overwrite || false;
167     params.mimeType = params.fileObj.type || 'text/plain';
168 }

```



```

justthis.lan : /Applications/GoogleAppEngineLauncher.app/Contents/Resources/GoogleAppEngine-default.bundle/Contents/Resources/google_appengine/google_appengine...

GoogleAppEngine-default.bundle
├── Contents
│   ├── MacOS
│   └── Resources
│       ├── google_appengine
│       │   ├── demos
│       │   └── google
│       │       ├── appengine
│       │       │   ├── api
│       │       │   ├── base
│       │       │   ├── datastore
│       │       │   ├── ext
│       │       └── admin
│       ├── google_appengine
│       │   ├── api
│       │   ├── base
│       │   ├── datastore
│       │   ├── ext
│       └── admin
└── google_appengine/google_appengine/ext/db

W-0 ...rces/google_appengine/google_appengine/ext/db
├── idjangoforms.py
├── __init__.py
└── __init__.pyc

W-0 ...rces/google_appengine/google_appengine/ext/db
├── Parents
├── Variables
├── validate (self,value)
├── Misc
├── DateTimeProperty : class
├── Parents
├── Variables
├── __init__ (self,verbose_name,auto_now,auto_now_add,now,now_add,now_callable,now_callable_kwargs,default_value(self),get_value_for_datastore (self,model_instance),now(),validate (self,value))
├── Misc
├── models.py
├── __init__.py
└── __init__.pyc

W-0 History
Variables: auto_now_add

data_type = Blob

class DateTimeProperty(Property):
    """The base class of all of our date/time properties.

    We handle common operations, like converting between time tuples and
    datetime instances.
    """
    def __init__(self, verbose_name=None, auto_now=False, auto_now_add=False,
        **kwargs):
        """Construct a DateTimeProperty

        Args:
            verbose_name: Verbose name is always first parameter.
            auto_now: Date/time property is updated with the current time every time
                it is saved to the datastore. Useful for properties that want to track
                the modification time of an instance.
            auto_now_add: Date/time is set to the when its instance is created.
                Useful for properties that record the creation time of an entity.
        """
        super(DateTimeProperty, self).__init__(verbose_name, **kwargs)
        self.auto_now = auto_now
        self.auto_now_add = auto_now_add

    def validate(self, value):
        """Validate datetime.

        Returns:
            A valid value.

        Raises:
            ValueError if property is not instance of 'datetime'.
        """
        value = super(DateTimeProperty, self).validate(value)
        if value and not isinstance(value, self.data_type):
            raise ValueError("Property %s must be a %s" %
                (self.name, self.data_type.__name__))
        return value

    def default_value(self):
        """Default value for datetime.

        Returns:
            value of now() as appropriate to the date-time instance if auto_now
            or auto_now_add is set, else user configured default value implementation.
        """
        if self.auto_now or self.auto_now_add:
            return self.now()
        return self.default_value

W-0 ...rces/google_appengine/google_appengine/ext/db
├── Parents
├── Variables
├── validate (self,value)
├── Misc
├── DateTimeProperty : class
├── Parents
├── Variables
├── __init__ (self,verbose_name,auto_now,auto_now_add,now,now_add,now_callable,now_callable_kwargs,default_value(self),get_value_for_datastore (self,model_instance),now(),validate (self,value))
├── Misc
├── models.py
├── __init__.py
└── __init__.pyc

W-0 History
Variables: auto_now_add

```


Python'un kendi paket yöneticisi:

```
apt-get install python-pip  
yum install python-pip  
pacman -S python2-pip (python-pip)
```

Temel komutları:

```
pip install <paket_adi>  
pip search <paket_adi>  
pip uninstall <paket_adi>  
pip install -r requirements.txt
```

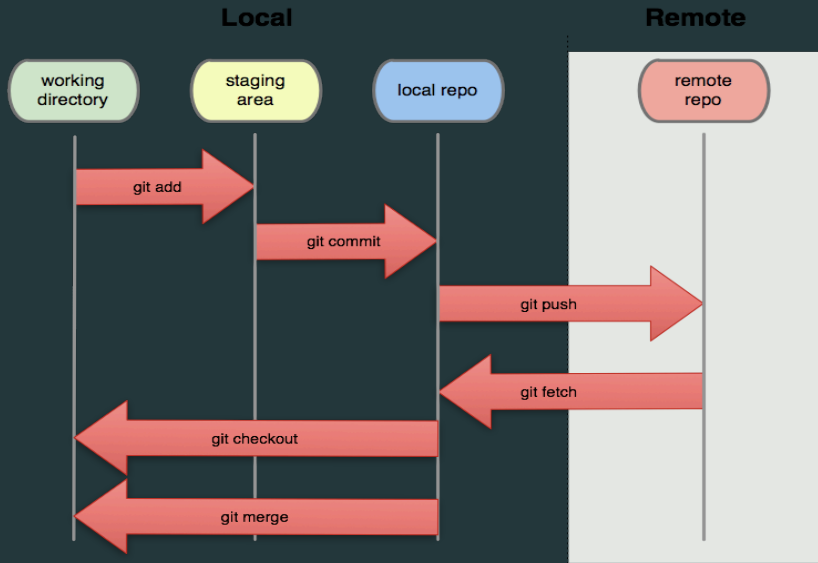
Başındaki **i** interaktif kelimesinden geliyor. Geliştirme yaparken kolaylık sağlıyor.

```
pip install ipython
```

Sürüm kontrol ve kaynak kod yönetim yazılımıdır. Her ne kadar fazla kodumuz olmasa da kullanmaya alışmak gereklidir.

```
apt-get install git  
yum install git  
pacman -S git
```

GIT ŞEMASI!



Temel komutlar:

```
git init
git add <dosya_adi>
git status
git log
git checkout -- <dosya_adi>
git branch <dal>
git checkout <dal>
git branch -D <dal>
git pull <repo>
git push <repo>
```

Git depolarımızı ücretsiz olarak barındırma hizmeti sağlıyor. Bilinen birçok büyük proje aktif olarak **Github** depolarında geliştiriliyor.

```
git config --global user.name "isim"  
git config --global user.email "e-posta"
```

HTTPS kullanacaksak depoları çekerken **kullanıcı adı** ve **şifre** girmemizi ister. **SSH** kullanacaksak **SSH** keyi oluşturup **Github** içerisinde tanıtmalıyız.

GITHUB ÇOK GÜZEL GELSENIZE!



twbs / bootstrap

Watch

4,465

Star

75,817

Fork

28,847

The most popular HTML, CSS, and JavaScript framework for developing responsive, mobile first projects on the web. <http://getbootstrap.com>

Code



Facebook



We work hard to contribute our work back to the web, mobile, big data, & infrastructure communities. NB: members must have two-factor auth.

Menlo Park, California

<https://code.facebook.com/projects/>

Filters

Find a repository...

mysql-5.6

C++ ★ 204 🍴 83

Facebook's branch of the Oracle MySQL v5.6 database

Updated 12 hours ago

rocksdb

C++ ★ 2,909 🍴 541

A library that provides an embeddable, persistent key-value store for fast storage.

Updated 12 hours ago

Python bağımlılıkları için izole bir ortam oluşturuyor. Yüklediğiniz paketler sistemin genelini etkilemiyor. Paketlerin farklı sürümleri ile çalışma imkanı sağlıyor.

```
pip install virtualenv
```

Temel komutlar:

```
cd <klasor_adi>
virtualenv <sanal_ortam_adi>
virtualenv -p /usr/bin/python2.7 <sanal_ortam_adi>
source <sanal_ortam_adi>/bin/activate
deactivate
```

Birleşmenin tam zamanı!

```
pip freeze > requirements.txt
pip install -r requirements.txt
```

FUNDAMENTALS

Python2:

```
print "hello world"
```

Python3:

```
print("hello world")
```

Oluşturduğunuz her `py` uzantılı dosyalar birer Python modülüdür. Ancak doğrudan çalışmasını da sağlayabilirsiniz. Bir önceki slaytta yazan satırları `hello.py` olarak kaydedebilirsiniz.

```
python hello.py
```

*nix ortamında çalıştırılabilir dosya yapmak için, dosyanın en başına aşağıdaki satır eklenir.

```
#!/usr/bin/env python
```

Son olarak:

```
chmod +x hello  
./hello
```

Python'da her şey bir nesnedir. Bu nesneler kimliklerini belirten bir `id` ve değerini tutan `value` sahibidir.

```
sdu = 3.14159265538979323846  
id(sdu)
```

Sahip oldukları değerler `değiştirilebilir(mutable)` ve `değiştirilemez(immutable)` olabilir.

Mutable değerler Üzerinde değişiklik yaptığınızda **id** değeri değişmez. Ancak **Immutable** değerler üzerindeki değişiklikler **id** değerini değiştirir.

```
hacktrick = 3.14159265538979323846
id(hacktrick)
hacktrick += 1.0
id(hacktrick)
```

```
sepet = ["elma", "armut", "kiraz"]
id(sepet)
sepet.append('karpuz')
id(sepet)
```

```
en_buyuk = "besiktas"
id(en_buyuk)
en_buyuk += "tir"
id(en_buyuk)
```

Değişken isimlendirmeleri önemlidir.

```
ahmet = 1  
osman = 1.0  
altan = "ohooneleryazilirburaya"  
ceyda_osman_kemal = []  
ayse_ezgi_riza = {}
```


Matematiksel işlemler için diğer dillerden bilinen $+$, $-$, $/$, $*$ operatörleri aynen geçerlidir. Farklı olarak üs alma işlemi için $**$ ve $\%$ vardır.

```
ahmet = 5 + 6  
ahmet += 2  
ahmet = 3 * 5  
ahmet = 2**5  
ahmet = ahmet % 5  
ahmet %= 5  
ahmet *= 10
```

Tam sayı bölme işlemi diğer dillerde olduğu gibi karıştırılabilir.

```
ahmet = 5 / 2  
ahmet = 5 / 2.0  
ahmet = 5.0 / 2
```

Tam sayı olarak bellekte tutulan en büyük değerin üstüne çıkarsanız, artık o sayıyı **long** tipinde saklamaya başlarsınız.

```
import sys  
ahmet = sys.maxint  
type(ahmet)  
ahmet += 1  
type(ahmet)
```

Metinler karakter dizisi olarak tutulur. Python'da bu karakter dizileri `str` tipindedir. Ayrıca stringler `immutable` bir veri tipidir.

```
ahmet = "iste bu bir string"  
ahmet += "tir."
```

Python'da stringleri tek tırnak, çift tırnak veya üç tırnak ile yazabilirsiniz. Kullanım sebepleri çeşitli sorunlar ile karşılaşınca daha rahat anlaşılır.

```
ahmet = "iste bu bir string"  
ahmet = 'ama bu da bir string'  
ahmet = """vallaha bu da string"""
```

Stringler üzerinde çeşitli formatlama işlemleri yapılabilir.

```
ahmet = "%s %s" %('bu bir', 'string')  
ahmet = "{0} {1}".format('bu bir', 'string')
```

Veri tiplerinin hangi metodlara sahip olduğunu ezberlemenize gerek yok. Python içerisinde tanımlı gelen `help`, `dir` fonksiyonları size yardımcı olacaktır.

```
ahmet = "bu string"  
melda = 3.14159165538979323846  
dir(ahmet)  
dir(melda)  
help(ahmet.upper)  
help(ahmet.title)  
help(melda.hex)
```

`dir` komutunu kullandığınız zaman iki tane alt çizgi ile başlayan(double underscore, dunder) metodlar olduğunu da göreceksiniz. Onlar normalde doğrudan kullanmayacağınız ancak başka bir nesne ile etkileşime girdiğinde kullanacağınız metodlardır.

```
ahmet = 2  
melda = 3  
ceyda = ahmet.__add__(melda)
```

Python'da yorumlar # ile başlar. C'de olduğu gibi bir kerede çok satır içeren yorum satırını doğrudan oluşturmakta kullanılmaz. Ancak o amaç için yapılmış başka bir yorum türü vardır.

Python'da ilerlemenin en temel şartı sahip olduğu veri tiplerine hakim olmaktır.

NoneType:

```
ahmet = None
```

Boolean:

```
ahmet = True  
melda = False
```


Python içerisinde veri tiplerini kıyaslamak için `>`, `<`, `>=`, `<=`, `==`, `!=` operatörlerini kullanabilirsiniz.

```
"ahmet" > "mehmet"  
"ahmet" == "ahmet"  
546 != 457  
x = 3.14159265538979323846  
2 < x < 5  
isinstance(x, float)  
isinstance("ahmet", str)
```

Mantıksal işlemler için `and` ve `or` kullanılır.

Farklı tür verileri birarada tutmak için Python `liste(list)`, `sözlük(dict)`, `küme(set)` ve `demet(tuple)` veri tiplerine sahiptir.

```
meyveler = ["elma", "armut"]  
elma = list("elma")
```

```
meyveler = {"elma": "kirmizi", "armut": "sari"}  
meyveler = dict()
```

```
meyveler = {"elma", "armut", "kiraz", "elma"}  
meyveler = set()
```

```
meyveler = ("elma", )  
meyveler = ("elma", "armut")  
meyveler = tuple()
```

Python sözlükleri veri yapısı olarak **hashmap** veya **ilişkili dizi** olarak da bilinir.

```
meyveler = {"elma": "kirmizi", "armut": "sari"}  
meyveler["elma"] = "yesil"
```

in ifadesi nesnenin **__contains__** metodunu kullanmaktadır.

```
print "armut" in meyveler  
print "kiraz" in meyveler
```

get metodu ile erişim yapılabilir ayrıca **del** metodu ile de silinebilir.

```
meyveler["kiraz"]  
meyveler.get("kiraz", "yok ki")  
del meyveler["kiraz"]
```

```
ahmet = "hihahoo"  
meyveler = ["elma", "kiraz", "armut"]  
turgut = "elma,kiraz,armut"  
print ahmet[0]  
print ahmet[-1]  
print meyveler[0:1]  
print meyveler[:1]  
print meyveler[:-1]  
print meyveler[::-1]  
print ahmet[::-1]  
print len(ahmet)  
print turgut.split(',')  
print len(turgut.split(',')) == len(meyveler)  
ahmet.sort()  
print ahmet.sorted()  
elma, kiraz, armut = meyveler
```

```
x = int(raw_input("gir bakim: "))  
if x < 0:  
    print "negatif"  
elif x < 10:  
    print "0 ile 10 arasinda"  
else:  
    print "ne bicim sayi bu?"
```

For:

```
meyveler = ["elma", "armut", "kiraz", "muz"]  
for meyve in meyveler:  
    print meyve, len(meyve)
```

While:

```
meyveler = ["elma", "armut", "kiraz", "muz"]  
i = 0  
while i < len(meyveler):  
    print meyveler[i], len(meyveler[i])
```

Diğer dillerde olan **break** ve **continue** deyimleri Python içerisinde de vardır.

```
sayilar = [2, 3, 1, 5, -6, 7, -8, -2, -1]
for sayi in sayilar:
    if sayi < 0:
        break

for sayi in sayilar:
    if sayi < 0:
        continue

for sayi in range(100):
    pass

for mk in meyveler.keys():
    ...
for mv in meyveler.values():
    ...
for mi in meyveler.items():
    ...
```

Python'da fonksiyonları `def` deyimi ile birlikte tanımlarız. Dikkat edilmesi gereken en önemli nokta 4 boşluk bırakılıp bırakılmadığıdır. Ayrıca fonksiyonların ilk satırı `docstring` olarak isimlendirilir. Buraya yazdıklarınız `help` metodu ile okunabilir.

```
def ekranci():  
    """ ekrana acayip seyler basar """  
    print "acayip bisi"  
  
def eklemeci(a, b, c, d, e=23):  
    """ verdigin her seyi ekler birbirine """  
    return a + b + c + d + e  
  
ekranci()  
eklemeci(1, 2, 3)  
eklemeci(1, 2, 3, 4)
```



```
l = "elma armut kiraz karpuz muz"
t_m = [m for m in l.split() if len(m) > 3]
t_s = [[m.upper(), m.lower(), len(m)] \
        for m in l.split()]
t_l = [s for s in range(0, 100) if s % 2]
t_k = {s for s in range(0, 100) if s % 2}
i_s = {s:s+1 for s in range(0, 100) if s % 2}
```

```
tek_mi = lambda x: x % 2 == 1
arttirsana = lambda x: x + 1
sayilar = ["1", "532", "45", "12356"]
sayilar = map(int, sayilar)
sayilar = map(lambda x: x + 1, sayilar)
sayilar_b = map(lambda x: x % 2 == 1, sayilar)
sayilar_t = filter(lambda x: x % 2 == 1, sayilar)
toplam = reduce(lambda x, y: x+ y, range(10))
```

Aslında her Python kodu içeren dosya bir modüldür. Modüller içerisinde fonksiyon, sınıf tanımlamaları ile çalıştırılabilir satırlar yer alır. Her modül sadece **import** edildiğinde çalıştırılır. Modüller içerisinde **private** sembol tabloları da bulunur.

```
import modul
modul.fonksiyon()
modul.__name__
fonksiyon = modul.fonksiyon()
fonksiyon()
from modul import fonksiyon1, fonksiyon2
fonksiyon1()
fonksiyon2()
from modul import *
```

```
class Hayvan(object):  
    """ Hayvan sinifi """  
    def __init__(self, isim):  
        """ Yapici metodumuz """  
        self.isim = isim  
    def konus(self):  
        """ Konusturan metodumuz """  
        print "Ben hayvanim: ", self.isim  
  
sari_kiz = Hayvan("sari kiz")  
prens = Hayvan("prens")  
  
sari_kiz.konus()  
prens.konus()
```

```
class Okuz(Hayvan):  
    def konus(self):  
        """ Konusturan metodumuz """  
        print "Ben okuzum: ", self.isim  
  
sari_kiz = Okuz("sari kiz")  
sari_kiz.konus()
```

```
dosya = open("dosya.txt", "r")
for satir in dosya:
    print satir
dosya.close()
```

```
dosya = open("dosya.txt", "w")
dosya.write("yaz beni")
dosya.close()
```

```
yazdigin = raw_input("Yaz bakim: ")
```

```
import sys
print sys.argv
```

```
try:
    lay_lay_lom = 123456 / 0
except ZeroDivisionError:
    print "oha 0'a bolmeye calistin oha!"
else:
    print "yok ya 0'a bolmedi"

...
def cok_iyi_fonksiyon(a, b):
    if b != 0:
        return a/b
    else:
        raise CokIlginchHata
...
```

ORNEKLER

```
def get_password(file_name):  
    with open(file_name) as fp:  
        while True:  
            data = fp.readline()  
            if not data:  
                break  
            yield data  
  
passwords = get_password('rockyou.txt')  
for password in passwords:  
    print password
```

WORDLIST READER

```
import argparse
import sys

if __name__ == "__main__":
    parser = argparse.ArgumentParser("tool",
                                     description="My Perfect Tool")
    parser.add_argument("--server", "-s",
                        help="Run the server", action="store_true")
    parser.add_argument("--port", "-p",
                        help="Set the server port number",
                        type=int, default=8000)

    args = parser.parse_args()

    if len(sys.argv) == 1:
        parser.print_help()
    else:
        pass
```

```
[genel_ayarlar]
site_adi = http://www.google.com
kullanici_adi = root
parola = g00gl3

okuyucu = SafeConfigParser()
okuyucu.read('ayarlar.conf')

print okuyucu.get('genel_ayarlar', 'site_adi')
```

```
import logging
LOG_DOSYASI = "logcu.out"
logging.basicConfig(filename=LOG_DOSYASI,
                    level=logging.DEBUG)
logging.debug("neler oldu neler")
```

```
def nmap_handler(host):  
    nmap_params = ['nmap', '-sV', '-n', '--open',  
                  host, '-oX', '-']  
    process = subprocess.Popen(nmap_params,  
                               stdout=subprocess.PIPE)  
    result, _ = process.communicate()  
    return result
```

```
import pickle

s = {"veri": "veri2", "veri2": 1234}
t = pickle.dumps(s)

a = pickle.loads(t)
```

```
from xml.dom import minidom

xml = minidom.parseString(result)
ports = xml.lastChild.getElementsByTagName('port')
hosts = xml.lastChild.getElementsByTagName('hosts')[0]
host_status = hosts.attributes['up'].value

for port in ports:
    port_id = port.attributes['portid'].value
    protocol = port.attributes['protocol'].value
```

```
import re
import requests

mail_r = re.compile("[\w\.-]+@[\w\.-]+")
content = requests.get("http://site.com").content
mails = mail_r.findall(content)
```



```
import socket

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_address = ('localhost', 1337)
sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
sock.bind(server_address)
sock.listen(5)
```

- Multi thread telnet cracker, socket
- Interactive tools, cmd
- Pcap file read, write, scapy
- Icmp data leak, scapy
- SSH botnet, fabric
- Non-blocking io, gevent
- Multi process downloader, multiprocessing

PYTHONIC

```
temp = ahmet  
ahmet = mehmet  
mehmet = temp  
  
ahmet, mehmet = mehmet, ahmet
```

```
sonuc = None
if 'dizin' in ayarlar:
    sonuc = ayarlar['dizin']
else:
    sonuc = on_tanimli

sonuc = ayarlar.get('dizin', on_tanimli)
```

```
sonuc_listesi = ["True", "False", "Yok"]  
sonuc_stringi = ""  
for sonuc in sonuc_listesi:  
    sonuc_stringi += sonuc  
  
sonuc_listesi = ["True", "False", "Yok"]  
sonuc_stringi = "".join(sonuc_listesi)
```

```
dosya = open("dosya.txt", "r")
for satir in dosya:
    uber_sonik_islem(satir)
dosya.close()

with open("dosya.txt", "r") as dosya:
    for satir in dosya:
        uber_sonik_islem(satir)
```

```
if i == "S" or i == "D" or i == "U":  
    uber_sonik_islem()  
  
if i in ("S", "D", "U"):  
    uber_sonik_islem()
```



```
ss = range(1, 100)
l = []
for s in ss:
    if asal_mi(s):
        l.append(s+5)

ss = range(1, 100)
l = [s+5 for s in ss if asal_mi(ss)]
```

```
i = 0
for eleman in listem:
    print i, eleman
    i += 1

for i, eleman in enumerate(listem):
    print i, eleman
```

```
def liste_olusturucu(arg):  
    listem = []  
    for i in uber_sonik_islem(arg):  
        listem.append(i)  
    return listem
```

```
def liste_olusturucu(arg):  
    for i in uber_sonik_islem(arg):  
        yield i
```

```
sayilar = [1, 5, 2, 4, 3]
for i in range(len(sayilar)):
    sayilar[i] += 3

map(lambda x: x+3, sayilar)
```

```
toplam = 10
for i in (1, 2, 3):
    toplam = toplam ** i

reduce(lambda x, y: x**y, [1, 2, 3], 10)
```

```
sayilar = [5, 6, 3, 4, 7, 9, 1]
yeni_sayilar = []
for i in range(len(sayilar)):
    if sayilar[i] % 2 == 0:
        yeni_sayilar.append(sayilar[i])

filter(lambda x: x%2 == 0, sayilar)
```

```
map(lambda x: x+3, listem)  
[i+3 for i in listem]
```

```
filter(lambda x: x%2 == 0, listem)  
[i for i in listem if i % 2 == 0]
```

```
pip install requests
```

```
import requests  
istek = requests.get("http://blabla.com")  
istek.text  
istek.headers  
istek.status_code  
istek.ok
```



```
import pdb  
pdb.set_trace()
```

h help

s step into

n next

c continue

w where

l list

```
from bottle import route, run, template

@route('/selam/<isim>')
def index(isim):
    cevap = '<b>Merhaba {{isim}}</b>!'
    return template(cevap, isim=isim)

run(host='localhost', port=8080)
```

SORULAR?