



pythonTM
ile Python gibi kod yazma

Gürer Özen
gurer @ pardus.org.tr

Liste
[1, 2, 3, 4]

?

Demet
(1, 2, 3, 4)

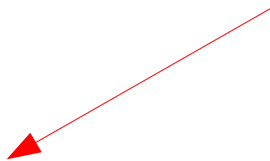
Tek elemanlı demet

(1,)

```
int find_cell(char *text, int *row, int *column)
{
    ...
    if (found) {
        *row = x;
        *column = y;
        return 1;
    } else
        return 0;
}
```

```
def find_cell(text):
    ...
    if found:
        return (row, column)
    else:
        return None
```

Birden fazla değeri
döndürmek için demet
idealdir!



Liste içinde demet!

```
adaylar = [ ( "Ali", 24 ), ( "Zeynep", 22 ), ( "Ekin", 90 ) ]
```

```
for aday in adaylar:  
    if aday[1] > 20 in aday[1] < 25:  
        print "Adı", aday[0], "yaşı", aday[1]
```

Dönüşte demet!

```
for aday, yas in adaylar:  
    if yas in range(21, 25):  
        print "Adı", aday, "yaşı", yas
```

Karşılaştırmada demet!

Metin nasıl birleştirilir?

“merhaba ” + “dünya”

“%s %s” % (“merhaba”, “dünya”)

string.join(("merhaba ", "dünya"))

“ ”.join(("merhaba", "dünya"))

```
isimler = ( "ali", "barış", "can", "didem")
```

```
metin = ""
```

```
for isim in isimler:
```

```
    metin += ", %s" % isim
```

```
metin = metin.join(isimler)
```

```
metin == " ".join( metin.split() )
```

Temel döngü yapısı

```
for (int i = 0; i < 100; i++) {  
    ...  
}
```

```
i = 0  
while i < 100:  
    ...  
    i += 1
```

```
for i in range(100):  
    ...
```

```
a = (1, 2, 3, 4, 5)
```

```
def cift_mi(x):  
    return x & 1 == 0
```

```
def kare(x):  
    return x * x
```

```
b = []  
for x in a:  
    if cift_mi(x):  
        b.append(x)
```

```
b = filter(cift_mi, a)
```

```
b = []  
for x in a:  
    b.append(a * a)
```

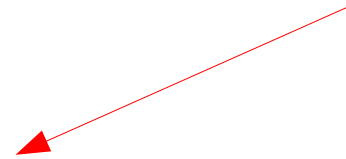
```
b = map(kare, a)
```

```
d = map( kare, filter( cift_mi, a ) )
```

QListBox

```
dns = []
item = self.dns_widget.firstChild()
while item:
    dns.append(str(item.text()))
    item = item.next()
```

API yetersiz!



```
class Iter:
    def __init__(self, box):
        self.item = box.firstChild()
    def next(self):
        item = self.item
        if not item:
            raise StopIteration
        self.item = item.next()
        return item
```

```
class IterableListBox(QListBox):
    def __iter__(self):
        return Iter(self)

self.dns = IterableListBox()

dns = list(self.dns)
```

İç içe bir kod

```
for yol, dizinler, dosyalar in os.walk("/home/koray"):
    for dosya in dosyalar:
        if dosya.endswith(".key"):
            if sha1sum(yol, dosya) != db_get_sha1(yol, dosya):
                dosya_kurtar(yol, dosya)
```

Bellek harcayan bir kod

```
def get_key_files(path):
    tmp = []
    for yol, dizinler, dosyalar in os.walk(path):
        if dosya.endswith(".key"):
            tmp.append((yol, dosya))
    return tmp

for dosya, yol in get_key_files("/home/koray"):
    if sha1sum(yol, dosya) != db_get_sha1(yol, dosya):
        dosya_kurtar(yol, dosya)
```

Üreteç kullanan bir kod

```
def get_key_files(path):  
    for yol, dizinler, dosyalar in os.walk(path):  
        if dosya.endswith(".key"):  
            yield (yol, dosya)
```

```
for dosya, yol in get_key_files("/home/koray"):  
    if sha1sum(yol, dosya) != db_get_sha1(yol, dosya):  
        dosya_kurtar(yol, dosya)
```

switch ... case ... yok bu Python'da?

```
switch(x - 2) {  
    case 0:  
        return y + 1;  
    case 1:  
        return y*y + 5*y - 9  
    case 2:  
        return y - (y+1) / 3  
    default:  
        return x*y  
}
```

```
t = x - 2  
if t == 0:  
    return y + 1  
elif t == 1:  
    return y*y + 5*y - 9  
elif t == 2:  
    return y - (y+1) / 3  
else:  
    return x*y
```

Ya da var!

```
operations = {  
    0: lambda: y + 1,  
    1: lambda: y*y + 5*y - 9  
    2: lambda: y - (y + 1) / 3  
}
```

```
func = operations.get(x - 2, lambda: x*y)  
func()
```

Pythonic olmayan bir API örneği:

```
from xml.dom.minidom import getDOMImplementation

impl = getDOMImplementation()

doc = impl.createDocument(None, "Etiket", None)
top = doc.documentElement
text = doc.createTextNode("Karakter verisi")
top.appendChild(text)
```

Pythonic olan başka bir tasarım:

```
import piksemel

doc = piksemel.newDocument("Etiket")
doc.insertData("Karakter verisi")
```

API ilk uyarlama:

```
def komut_oku(socket):
```

```
    ...
```

```
    return (istemci, komut, deger1, deger2)
```

```
t = komut_oku(s)
```

```
print t[0]
```

API gelişmesini sürdürür:

```
class Cevap:
    def __init__(self, istemci, komut ,deger1, deger2):
        self.istemci = istemci
        self.komut = komut
        self.deger1 = deger1
        self.deger2 = deger2

    def __getitem__(self, key):
        return (self.istemci, self.komut, self.deger1, self.deger2)[int(key)]

def komut_oku(soket):
    ...
    return Cevap(istemci, komut, deger1, deger2)

t = komut_oku(soket)
print t[0]
print t.istemci
```

```
class Cevap:
    def __init__(self, istemci, komut ,deger1, deger2):
        self.istemci = istemci
        self.komut = komut
        self.deger1 = deger1
        self.deger2 = deger2
```

Tekrarlı bir kısım!



```
class Struct:
    def __init__(self, **args):
        self.__dict__.update(args)
```

```
class Cevap(Struct):
    pass
```

```
Cevap(istemci=istemci, komut=komut, ...)
```

Standart tipleri de geliřtirebiliriz:

```
a = { "en": "Hello world", "tr": "Merhaba dünya" }
```

```
metin = a.get(language(), None)
if not metin:
    a.get("en")
print metin
```

```
class Metin(dict):
    def __str__(self):
        txt = self.get(language(), None)
        if not txt:
            txt = self.get("en")
        return txt
```

```
a = Metin(a)
```

```
print a
```

Değişken isimleri önemli, çünkü tip bilgisi yok!

```
if aday == "Ali Veli":
```

```
...
```

```
if aday > 30:
```

```
...
```

```
print aday.aday
```

```
if aday_ismi == "Ali Veli":
```

```
...
```

```
if aday_adi > 30:
```

```
...
```

```
print aday.isim
```

Sakın ha!

```
l = 5
```

```
l = 8
```

```
O = 4
```

```
l - l + 0
```

dosya.py (Javadan etkilenmiş)

```
TGZ, ZIP, NORMAL = range(3)
```

```
class Dosya:
    @staticmethod
    def tip(ad):
        if ad.endswith(".tar.gz"):
            return TGZ
        elif ad.endswith(".zip"):
            return ZIP
        else:
            return NORMAL
```

```
...
```

dosya.py (Pythonic)

```
TGZ, ZIP, NORMAL = range(3)
```

```
def tip(ad):
    if ad.endswith(".tar.gz"):
        return TGZ
    elif ad.endswith(".zip"):
        return ZIP
    else:
        return NORMAL
```

```
class Dosya:
```

```
...
```

Boş except çok tehlikeli bir yapı!

```
try:  
    asdf  
except:  
    pass
```

Doğru yanlış kime göre...

```
dogru_ahmet = True
```

Berbat

```
if dogru_ahmet is True:
```

```
...
```



Singleton:

```
if deger is None:
```

```
...
```

Kötü

```
if dogru_ahmet == True:
```

```
...
```

Pythonic

```
if dogru_ahmet:
```

```
...
```

Sorularınız...