ANR 413 Bilgisayar Programlama

Python

#1

Serdar ARITAN

Biyomekanik Araştırma Grubu,
Spor Bilimleri Fakültesi
Hacettepe Üniversitesi
Ankara

Python; nesne yönelimli, yorumlanabilen, birimsel (modüler) ve etkileşimli bir programlama dilidir.

Python, Guido Van Rossum adlı Hollandalı bir programcı tarafından 90'lı yılların başında geliştirilmeye başlanmış bir programlama dilidir. Zannedildiğinin aksine bu programlama dilinin adı piton yılanından gelmez... Guido Van Rossum bu programlama dilini, "The Monty Python" adlı bir İngiliz komedi grubunun, "Monty Python's Flying Circus" adlı gösterisinden esinlenerek adlandırmıştır.

PYTHON PROGRAMLAMA DİLİNİN ÖZELLİKLERİ

- Nesneye yönelik bir programlama dilidir.
- Özgürdür.
- Derlenebilir
- Taşınabilirdir.
- Güçlü ve hızlıdır.
- Yazılımı kolay ve sadedir.
- Ticari uygulamalar geliştirilebilir.

- Söz dizimi (Syntax) yapısı

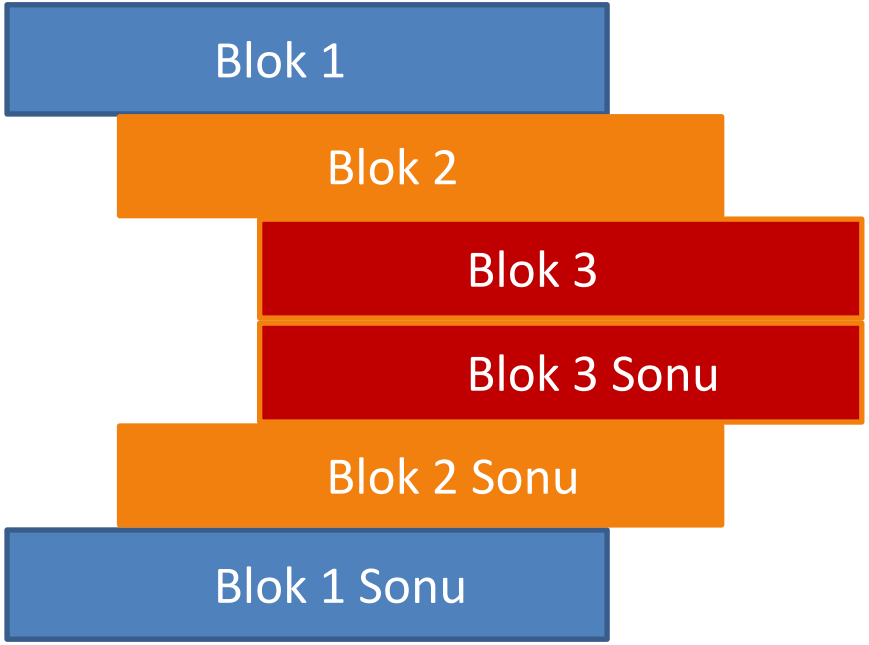
```
// C++ ile yazılan
#include "iostream"
using namespace std;
int main(){
Cout<“Merhaba Dünya”<<endl;
return 0;}
```

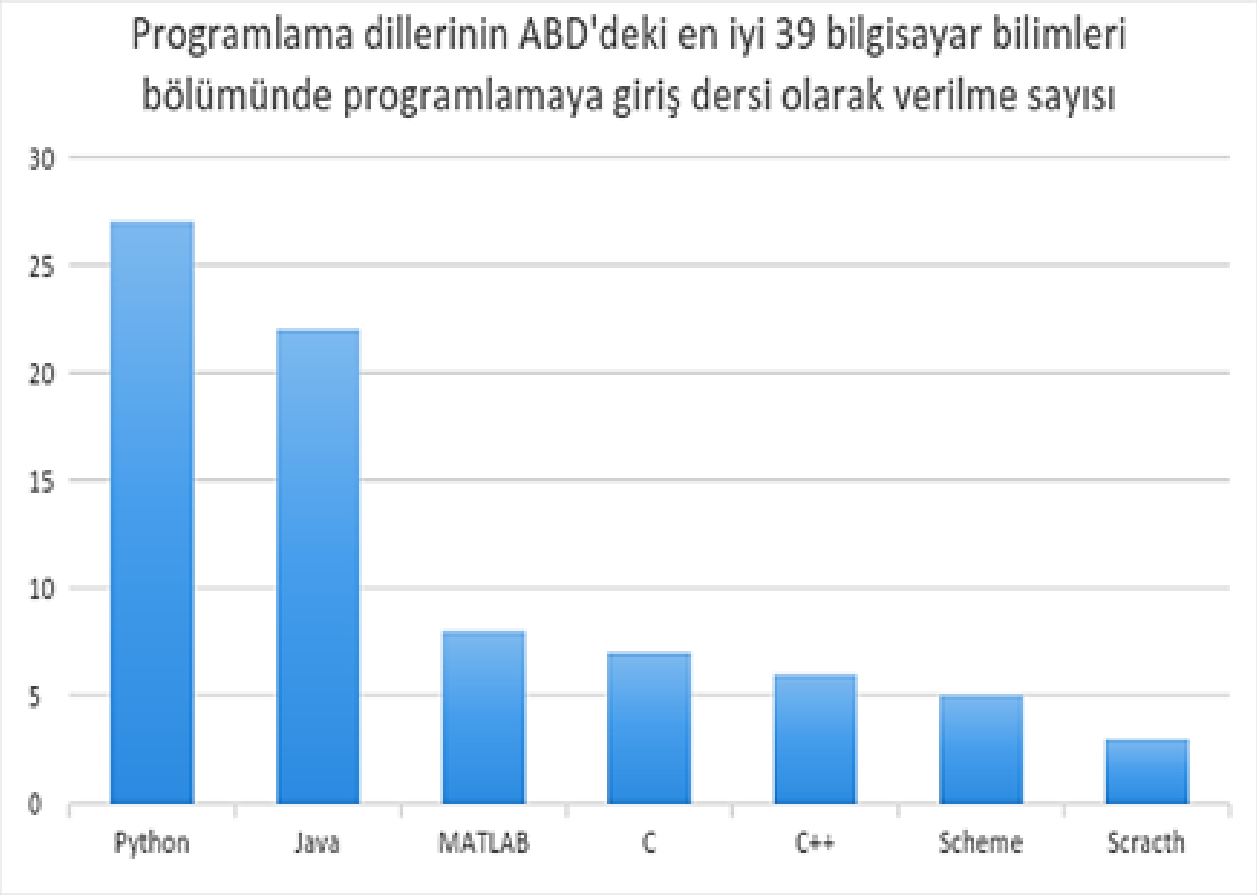
```
// Java ile yazılan
class jmerhaba {
public static void main(String args[]){
System.out.println(“Merhaba Dünya”);} }
```

```
# Python ile yazılan
print (“Merhaba dünya”)
```

```
// C# ile yazılan
public class Merhaba{
public static void Main() {
System.Console.WriteLine(“Merhaba dünya”);}}
```

- Girintileme (indentation)




Programlama dillerinin ABD'deki en iyi 39 bilgisayar bilimleri bölümünde programlamaya giriş dersi olarak verilme sayısı
30
25
20
15
10
5
0
Python
Java
MATLAB
C
C++
Scheme
Scracth

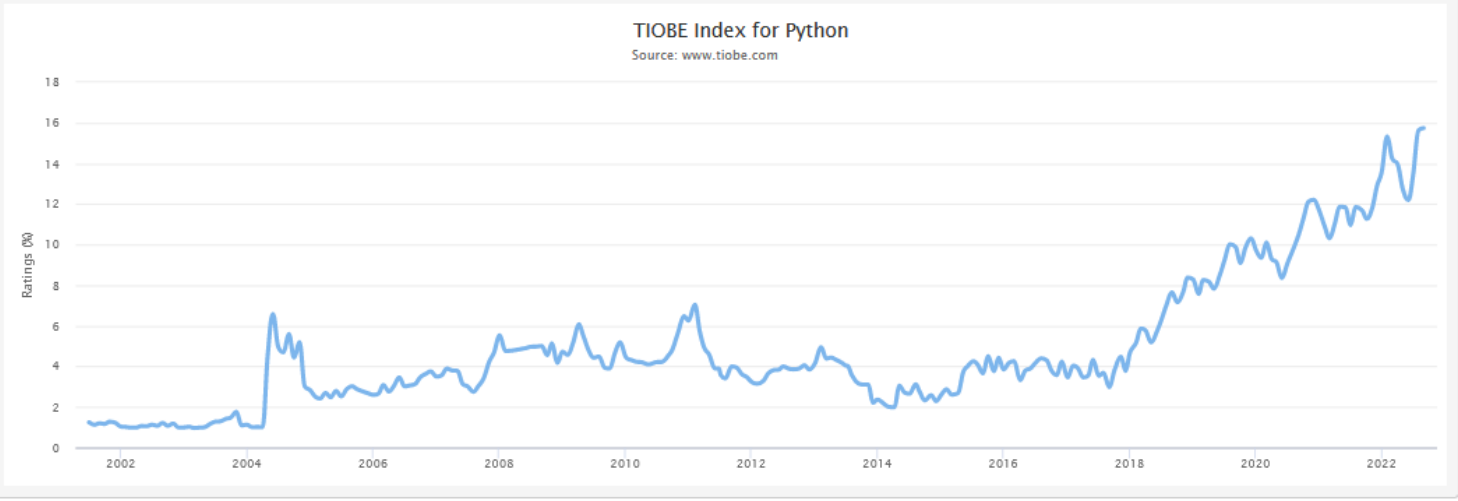
The Python Programming Language

Some information about Python:

Highest Position (since 2001): #1 in Sep 2022

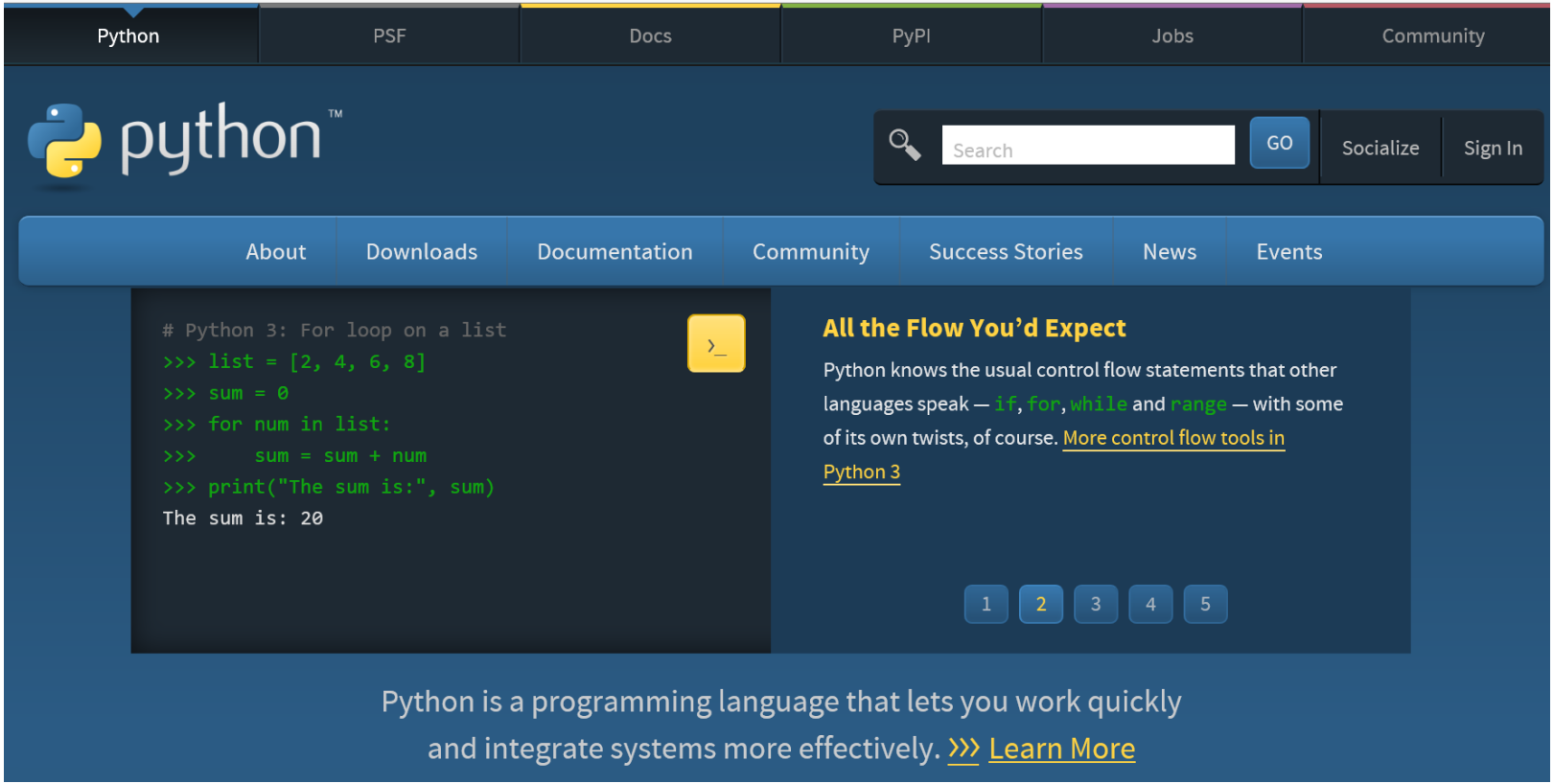
Lowest Position (since 2001): #13 in Feb 2003

Language of the Year: 2007, 2010, 2018, 2020, 2021



https://www.tiobe.com/tiobe-index/

http://www.python.org/



https://docs.python.org/3/

Download
Download these documents

Docs by version
Python 3.12 (in development)
Python 3.11 (pre-release)
Python 3.10 (stable)
Python 3.9 (security-fixes)
Python 3.8 (security-fixes)
Python 3.7 (security-fixes)
Python 3.6 (EOL)
Python 3.5 (EOL)
Python 2.7 (EOL)
All versions

Other resources
PEP Index
Beginner's Guide
Book List
Audio/Visual Talks
Python Developer's Guide

Python 3.10.7 documentation

Welcome! This is the official documentation for Python 3.10.7.

Parts of the documentation:

What's new in Python 3.10?
or all "What's new" documents since 2.0

Tutorial
start here

Library Reference
keep this under your pillow

Language Reference
describes syntax and language elements

Python Setup and Usage
how to use Python on different platforms

Python HOWTOs
in-depth documents on specific topics

Installing Python Modules
installing from the Python Package Index & other sources

Distributing Python Modules
publishing modules for installation by others

Extending and Embedding
tutorial for C/C++ programmers

Python/C API
reference for C/C++ programmers

FAQs
frequently asked questions (with answers!)

PYTHON DİLBİLGİSİ

Diğer programlama dillerinde olduğu gibi Python'da da kod yazımında uyulması gereken kurallar vardır. Bir dili kullanabilmek için bu kuralları iyi bilmek ve uygulamak gereklidir

- Python BÜYÜK-küçük harf ayrımı yapar.
- Komutların sonlarında ; (noktalı virgül) ve benzeri işleçler konulmaz.
- Pythonda kod yazarken girintiler kullanmak zorundasınız.
- Pythonda değişkenleri tanımlama yoktur.

Python un Ana Felsefesi PEP20 de özetlenmiştir (The Zen of Python)",

- **Beautiful** is better than **ugly**.
- **Explicit** is better than **implicit**.
- **Simple** is better than **complex**.
- **Complex** is better than **complicated**.
- **Readability** counts.

```
Try;
>>> import this
```

PYTHON OPERATÖRLERİ

Aritmetik Operatörler

Operatör	İşlem	Örnek
=	Atama Operatörü	A=4, b='Oku'
+	Toplama	A=4+86
-	Çıkarma	A=86-52
*	Çarpma	A=9*86
/	Bölme	A=86/12
**	Kuvvet alma	A=8**3 (8'in 3.kuvveti)

PYTHON OPERATÖRLERİ

Mantıksal Operatörler

Operatör	İşlem	Örnek
and	Ve işlemi	a == 3 **and** b == 12
or	Veya İşlemi	a == 5 **or** b == 43
not	Değil İşlemi	**not** a % 2 == 0

PYTHON OPERATÖRLERİ

Karşılaştırma Operatörler

Operatör	İşlem	Örnek
>	Büyüklük	A>B (A, B' den büyüktür)
>=	Büyüklük ya da eşitlik	A>=B (A, B' ye eşit veya büyük)
<	Küçüklük	A<B (A, B' den küçüktür)
<=	Küçüklük ya da eşitlik	A<=B (A, B' den küçük veya eşit)
==	Eşitlik	A=B (A, B' ye eşit)
<>	Eşit değil	A<>B (A, B' ye eşit değil)

PYTHON PROGRAMLAMA DİLİNDE DEĞİŞKENLER

- Diğer dillerde de olduğu gibi bir değişken RAKAM ile başlayamaz. İlk karakter bir harf veya _ (altçizgi) olmak zorundadır.

- Harf, Rakam ve _ (alt çizgi) haricinde bir karakter içeremez (örn: $, #,*, ? veya boşluk gibi).

- Aksi belirtilmedikçe tüm değişkenler yerel olarak algılanırlar.

PYTHON PROGRAMLAMA DİLİNDE DEĞİŞKENLER

Python programlarımızda geçici olarak veri saklamak için oluşturduğumuz alanlara değişken denir.

Değişkenler, değerleri saklamak için ayrılmış bellek konumlarıdır. Bir değişken oluşturduğunuzda hafızada yer ayırmanız anlamına gelir. Bir değişken değer türüne bağlı olarak belleğe yerleştirilir ve bellekten okunurken değişkenin değer türüne bağlı okuma yapılır. Bu nedenle, değişkenlere farklı veri türleri atayarak, bu değişkenlere tamsayı, ondalık veya karakter kaydedebilirsiniz. Bu bölüm, değişken kavramına ve Python değişken tiplerine ayrılmıştır.

PYTHON PROGRAMLAMA DİLİNDE DEĞİŞKENLER

Python değişkenleri, bellek alanını ayırmak için açık bildirime gerek duymaz. Başka bir deyişle C, C# ve Java dillerinde olduğu gibi değişkenin tipini önceden tanımlamaya gerek yoktur. Bu açıdan Javascript ve PHP dillerindeki tanımlamaya benzerdir. Bu konu öğrenciler tarafından değişken tipi yokmuş gibi algılanmaktadır!!! Bu yaklaşım yanlıştır. Değer tipi belirleme programlama dili yorumlaması sırasında atanan ilk değere göre belirlenir.

Başka bir deyişle tip belirleme atama işlemi olduğunda otomatik olarak belirlenir. Değer atamak için çoğu programlama dilindeki gibi eşittir (=) işareti kullanılır.

```
>>>ad="serdar"
>>>soyad='arıtan'
>>>=a=9
>>>b="9"
```

PYTHON PROGRAMLAMA DİLİNDE DEĞİŞKENLER

Python değişkenleri, bellek alanını ayırmak için açık bildirime gerek duymaz. Başka bir deyişle C, C# ve Java dillerinde olduğu gibi değişkenin tipini önceden tanımlamaya gerek yoktur. Bu açıdan Javascript ve PHP dillerindeki tanımlamaya benzerdir. Bu konu öğrenciler tarafından değişken tipi yokmuş gibi algılanmaktadır!!! Bu yaklaşım yanlıştır. Değer tipi belirleme programlama dili yorumlaması sırasında atanan ilk değere göre belirlenir.

Başka bir deyişle tip belirleme atama işlemi olduğunda otomatik olarak belirlenir. Değer atamak için çoğu programlama dilindeki gibi eşittir (=) işareti kullanılır.

```
>>>ad="serdar"
>>>soyad='arıtan'
>>>=a=9
>>>b="9"
```

PYTHON PROGRAMLAMA DİLİNDE DEĞİŞKENLER

```
>>>sayac = 100        # Tamsayı atama (int)
>>>km    = 1000.0     # Virgüllü sayı atama (float)
>>>ad     = "Çorlu"   # Katar (String)
>>>print(sayac)
>>>print(km)
>>>print(ad)
```

Bir değişken tanımlayıp bir değer atadığımızda arka planda işletim sistemi tarafından belirlenen bellek adresleri uygulamamıza tahsis edilir. Bellek büyüklüğü tanımlanan veri türünün boyutu ile alakalıdır.

Örnekteki, sayac değişkeni bilgisayarımızın belleğinde xxxx adresinden başlayıp 32 bitlik yer kaplar. Başka bir deyişle 4 byte'lık yer kaplamıştır. Bu 32 bitlik yer içinde 100 sayısı tutulmaktadır. Biz sayı değişkenini çağırdığımızda xxxx adrese gidilip 32 bitlik okuma yapılır ve karşımıza bellekteki sayı çıkar.

PYTHON PROGRAMLAMA DİLİNDE DEĞİŞKENLER

Python, aynı anda birden fazla değişkene tek bir değer atamanıza izin verir. Örneğin,

```
>>> a = b = c = 413
```

Burada, 413 değeriyle bir tamsayı nesnesi oluşturulur ve üç değişkenin tümü aynı bellek konumuna atanır. Birden çok değişkene birden çok nesne atayabilirsiniz. Örneğin,

```
>>>d, e, f = 413, 1, " Bilgisayar Programlama "
```

Burada, 413 ve 1 değerlerine sahip iki tamsayı nesnesi, sırasıyla d ve e değişkenlerine atanır ve “Bilgisayar Programlama” değerine sahip bir dize nesnesi, değişken f’ye atanır. Değerleri görüntüleyecek olursak,

PYTHON PROGRAMLAMA DİLİNDE DEĞİŞKENLER

Değerleri görüntüleyecek olursak,

```
print(a)
print(b)
print(c)
print(d)
print(e)
print(f)
```

Şimdi bu değişkenlerin nesne adreslerine bakalım. Bunun için `id(değişken adı)` fonksiyonu kullanacağız.

PYTHON PROGRAMLAMA DİLİNDE DEĞİŞKENLER

Şimdi bu değişkenlerin nesne adreslerine bakalım. Bunun için, **`id(değişken adı)`** fonksiyonu kullanacağız.

```
print(id(a))
print(id(b))
print(id(c))
print(id(d))
print(id(e))
print(id(f))
```

Çıktı olarak a, b, c değişkenleri aynı bellek adresini kullandıklarını görülür. Burası değişken konusunun en önemli kısımlarından biridir. Nesneye yönelik dillerde değer tabanlı atama ve adres tabanlı atama olmak üzere iki konu vardır. Adres tabanlı bir atama yapılınca iki değişken biri değişse de diğer değişkende etkilenir. Aslında değişiklik yapılan adres aynıdır.

PYTHON PROGRAMLAMA DİLİNDE DEĞİŞKENLER

sol taraf = sağ taraf

LHS = RHS

```
>>> width = 10                    <enter>
>>> length = 5                    <enter>
>>>
>>> print(width)                  <enter>
10
>>> print(length)                 <enter>
5
>>> print('width')                <enter>
width
>>> print(width)                  <enter>
10
>>> 25 = age          <enter>
SyntaxError: can't assign to literal
```

PYTHON PROGRAMLAMA DİLİNDE DEĞİŞKENLER

Bellekte saklanan veriler birçok türde olabilir. Örneğin, bir kişinin yaşı sayısal bir değer olarak saklanır ve onun adresi karakter katarı olarak saklanır. Python'un beş standart veri türü vardır.

Sayılar (Numbers)
Katar (String)
Liste (List)
Tuple
Dictionary

PYTHON PROGRAMLAMA DİLİNDE DEĞİŞKENLER

Sayılar : Sayı veri türleri sayısal değerleri saklar. Değişkene ilk değer atandığında nesne oluşur. Bu aşamada uygun sayı veri tipi tespit edilir ve nesne oluşturulur.

```
>>>sayi_1 = 10
>>>sayi_2 = 20
>>>sayi_3 = 30
```

int (signed integers): 100
float (floating point real values): 100.232
complex (karmaşık sayılar): 3.14j

Nesne olduğu için çalışma anında bellekten silinebilir. Bu işlem için, del ifadesi kullanılır. Örneğin,

```
del sayi_1
del sayi_2, sayi_3
```

Belleği daha etkin kullanımı için gereksiz değişkenlerinizi del komutu silebilirsiniz.

PYTHON PROGRAMLAMA DİLİNDE DEĞİŞKENLER

Katarlar (Strings) : Python'daki katarlar, tırnak işaretlerinde temsil edilen bitişik bir karakter kümesi olarak tanımlanır. Python, tek veya çift tırnak çiftlerine izin verir. Artı (+) işareti, katar birleştirme yaparken yıldız işareti (*) tekrarlama sağlar. Python, alt küme ulaşımı çok daha kolaylaştırmaktadır. [] ve [:] işleçleri kullanılarak alt katarlara ulaşılabilinir. Örneğin,

```
str = 'Merhaba Python!'

print(str)          # Tüm katarı yazar
print(str[0])       # İlk karakteri yazar
print(str[2:5])     # 3 ile 5. karakter arasını yazar.
print(str[2:])      # 3. karakterden başlayarak son karaktere kadar yazar.
print(str * 2)      # 2 kere yazar
print(str + " yeni katar") # Birleştirme işlemi yapar
```

PYTHON PROGRAMLAMA DİLİNDE DEĞİŞKENLER

Listeler (Lists): Listeler, Python'un bileşik veri tipidir. Bir liste, virgülle ayrılmış ve köşeli parantez ([]) içine alınmış öğeler içerir. Aslında listeler birçok programlama dilinde gördüğümüz diziler (arrays) benzerdir. Fakat, Python daha esnek bir yapı sunar. Bir liste içine birden farklı veri tipi tanımlayabilirsiniz.

Listede saklanan değerlere, listenin başlangıcında 0'dan başlayan ve -1'in sonuna kadar gider. ([] ve [:]) işleçleri kullanılarak erişilebilir. Artı (+) işareti liste birleştirme ve yıldız işareti (*) tekrarlama işlevi olur. Örneğin,

```
list1 = [ 'abcd', 786 , 2.23, 'Ankara', 70.2 ]
list2 = [413, 'Merhaba']
```

PYTHON PROGRAMLAMA DİLİNDE DEĞİŞKENLER

```
print(list1)           # Tüm listeyi yazar
print(list1[0])        # İlk öğeyi yazar
print(list1[1:3])      # 2. öğe ile 3. öğeyi yazar
print(list1[2:])       # 3. öğeden sonrasını yazar
print(list2 * 2)       # İki kere yazar
print(list1 + list2) #  İki listeyi birleştirir
```

PYTHON PROGRAMLAMA DİLİNDE DEĞİŞKENLER

Tuples : Bir tuple, listeye benzer bir veri türüdür. Bir tuple, virgülle ayrılmış bir dizi değerden oluşur. Ancak listelerden farklı olarak, köşeli parantez yerine parantezler (( )) içine alınır. Listelere göre farkı sadece okunabilir (read-only) modda olmasıdır. Öğe sayısı arttırılamaz ve güncellenemezler. Örneğin,,

```
tuple1 = ('abcd', 786 , 2.23, 'Ankara', 70.2)
tuple2 = (413, 'Merhaba')
print(tuple1)           # Tüm tuple öğelerini yazar
print(tuple1[0])        # İlk öğeyi yazar
print(tuple1[1:3])      # 2. öğe ile 3. öğeyi yazar
print(tuple1[2:])       # 3. öğeden sonrasını yazar
print(tuple2 * 2)       # İki kere yazar
print(tuple1 + tuple2)  # İki tupleyi birleştirir
```

PYTHON PROGRAMLAMA DİLİNDE DEĞİŞKENLER

Tuples : Güncelleme işlemi deneyelim

```
tuple = ( 'abcd', 786 , 2.23, 'Ankara', 70.2  )
list = [ 'abcd', 786 , 2.23, 'Ankara', 70.2  ]
tuple[2] = 1000     # Tuple güncellenemez. Hatalı satır.
list[2] = 1000      # List güncellenebilir.
```

PYTHON PROGRAMLAMA DİLİNDE DEĞİŞKENLER

Sözlük (Dictionary) : Python'un sözlükleri bir çeşit hash tablo türüdür. Birçok programlama dilinde (C#, Java, Perl) bulunan ve anahtar-değer çiftlerinden oluşan ilişkisel diziler ya da hash gibi çalışırlar. Bir sözlük anahtarı hemen hemen her Python tipi olabilir, ancak genellikle sayılar veya dizelerdir. Diğer taraftan değerler, herhangi bir rastgele Python nesnesi olabilir. Sözlüklere küme parantezleri ({}) ile çevrelenir ve değerler köşeli parantezler ([]) kullanılarak atanabilir ve erişilebilir.

Sözlüklerin en büyük avantajı anahtar üzerinden aramayı hızlandırmasıdır. Dizi üzerinde arama için tüm öğeler kontrol edilmelidir. Bunun için bir döngüye ve if kontrolüne ihtiyaç vardır.

PYTHON PROGRAMLAMA DİLİNDE DEĞİŞKENLER

Sözlük (Dictionary) : Bunun için bir döngüye ve if kontrolüne ihtiyaç vardır.

```
dict1 = {} #boş bir dictinary
dict1['one'] = "Anahtar değeri string olarak one - Değer kısmı"
dict1[2]     = "Anahtar değeri int olarak 2 - Değer kısmı"
```

PYTHON PROGRAMLAMA DİLİNDE DEĞİŞKENLER

```
>>> type(1)
<class 'int'>
>>> type(200000)
<class 'int'>
>>> type(99999999999999)
<class 'int'>
>>> type(1.0)
<class 'float'>
>>> type(5j)
<class 'complex'>
```

PYTHON PROGRAMLAMA DİLİNDE DEĞİŞKENLER

Data type Cat <class 'cat'>

PYTHON PROGRAMLAMA DİLİNDE DEĞİŞKENLER

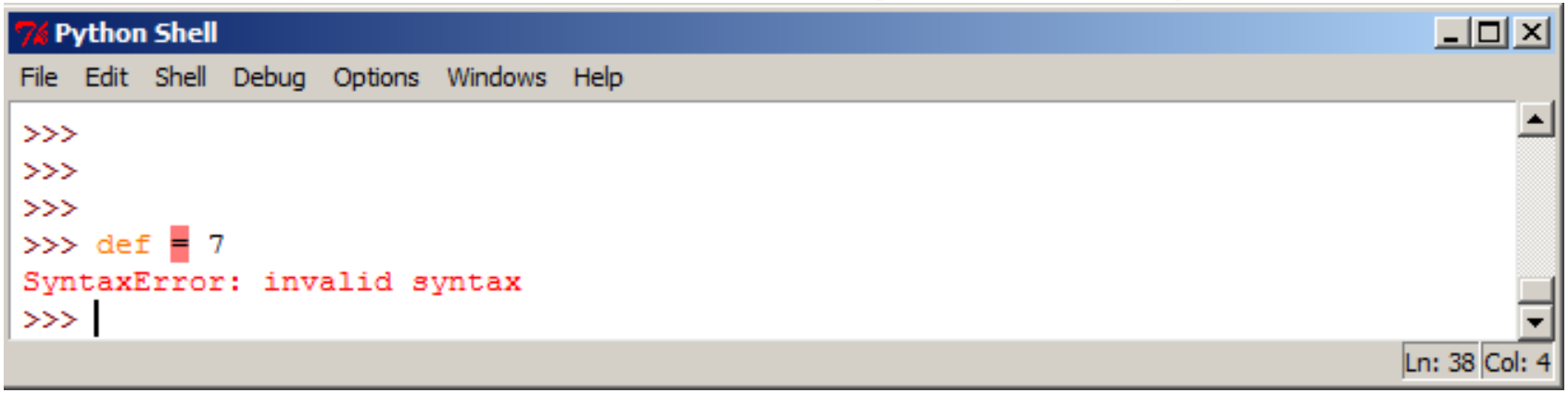
Key Words in Python

Aşağıdaki kelimeler Python tarafından kullanılmaktadır. Bu kelimeleri değişken adı olarak kullanmayın.

and	continue	except	global	lambda	pass	while
as	def	False	if	None	raise	with
assert	del	finally	import	nonlocal	return	yield
break	elif	for	in	not	True	
class	else	from	is	or	try	

PYTHON PROGRAMLAMA DİLİNDE DEĞİŞKENLER

Key Words in Python



"SyntaxError: invalid syntax..." anlamı , "Hey, senin ne dediğini anlamıyorum veya sen Python dilinden konuşmuyorsun.!"