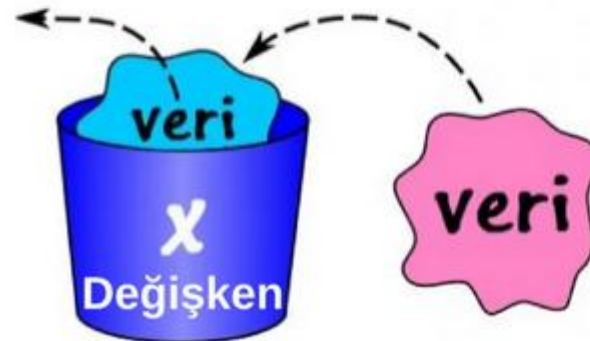


PYTHON DEĞİŞKENLER

Öğr. Gör. Halil ARSLAN

Python Değişkenler

- Değer taşıyan isimdir.
- İçinde veri tutan birimdir.
- Ram'de ayrılan bellek miktarıdır.
- Bir değişkene değer atama ifadesi eşit(=) kullanırız.
- Python'da değişkenleri önceden tanımlamak gerekmez.



Python Değişken Örnekleri

- `a=5` -> a'nın değeri 5 tir.
- `b=10` -> b'nin değeri 10 dur.
- `adi="Murat"` -> adi değişkeni Murat tır.
- `numara=2024` -> numara 2024 tür.
- `dil="Python"` -> dil'in değeri Python dur.
- `pi=3.14` -> pi bize 3.14 değeri verir.

Python'da Değişken İsimlendirme

- Değişkenler;
 - Sayı ile başlayamaz, (*2-Kisi=20 tl*)
 - Aritmetik işleçlerle başlayamaz, (*+fayda=6789*)
 - Harf veya `_` (alt_tire) ile başlamalıdır, (*adi, _adi*)
 - Belirli fonksiyona sahip adlar kullanılamaz, (*del, and,..*)
 - Kelimeler arasında boşluk bırakılmaz, (*ogrenci no*)
 - Büyük Harf ve küçük harf ayrımı Vardır. (*adi / Adi*)
- Tavsiyeler;
 - Çok kısa veya çok uzun olmamalıdır,
 - Mümkün olduğunca anlamlı olmalıdır,
 - Türkçe karakter kullanmak ilerde sorun olabilir.

Değişken İsimlendirme Yöntemleri

- Yazılım dünyasında değişken adlandırmada kullanılan yöntemler (Notasyon);
- **Pascal**
 - İlk harfleri büyük, devamı küçük harf.
- **Camel** (Deve)
 - İlk kelimenin ilk harfi küçük, geri kalan tüm kelimelerin ilk harfi büyük.
- **Macar** (Hungarian)
 - İlk 2 veya 3 harfte değişken tipi, sonrası anımsatıcı.
- **Uppercase** (Büyük Harf)
 - Tamamı büyük harf.
- **Underscore** (Altçizgili) [Dashes: Tire ile ayrılmış (page-not-found)]
 - Kelimeler altçizgili.
- **Abbreviations** (Kısaltmalar)
 - Kelimeler veya özellik adı kısaltılmış

Notasyon Örnekleri

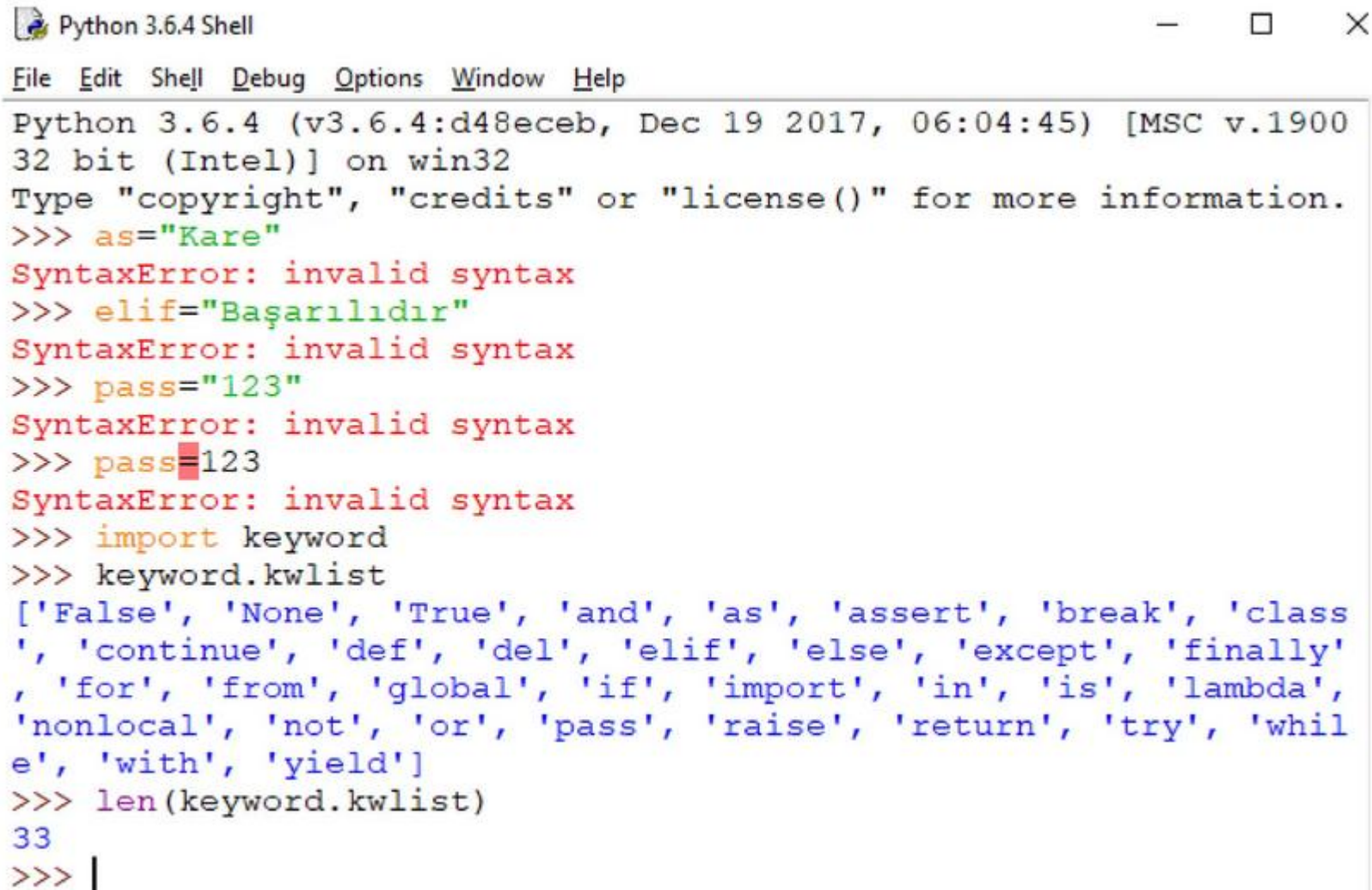
- **Pascal**
 - ToString, AdiSoyadi, TcKimlikNo, KategoriAdi,..
- **Camel (Deve)**
 - toString, adiSoyadi, tcKimlikNo, kategoriAdi,..
- **Macar (Hungarian)**
 - strString, txtAdiSoyadi, intTcKimlikNo, strKategoriAdi,..
- **Uppercase (Büyük Harf)**
 - STRING, ADISOYADI, TCKIMLIKNO, KATEGORIADI,..
- **Underscore (Altçizgili)**
 - String, adi_soyadi, tc_kimlik_no, kategori_adi,..
- **Abbreviations (Kısaltmalar)**
 - numStudent, maxValue

Python'da Kullanılmayan Kelimeler

- Özel anlam ifade eden kelimeler kullanılamaz.
 - ~~pass="123" veya pass=123~~
 - ~~elif="Başarılıdır"~~
 - ~~as="Kare"~~
- Keyword List (Anahtar Kelime Listesi) 36 adet kelimedir.

```
>>>import keyword
>>>keyword.kwlist
['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class',
'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for',
'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not',
'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
```

Keyword List



A screenshot of a Python 3.6.4 Shell window. The window title is "Python 3.6.4 Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area shows the following content:

```
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900
32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> as="Kare"
SyntaxError: invalid syntax
>>> elif="Başarılıdır"
SyntaxError: invalid syntax
>>> pass="123"
SyntaxError: invalid syntax
>>> pass=123
SyntaxError: invalid syntax
>>> import keyword
>>> keyword.kwlist
['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class',
',', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally',
',', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda',
'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'whil
e', 'with', 'yield']
>>> len(keyword.kwlist)
33
>>> |
```


EK BİLGİLER

Garbage Collector (Çöp Toplayıcı)

- Değişkene ilişkin atıf sona erince RAM'den silinmesi işlemidir.

```
>>> a=5
```

```
>>> print(a)
```

```
5
```

```
>>> a="Adana"
```

```
>>> print(a)
```

```
Adana
```



print() Fonksiyonu

```
>>> print (sep, end, file, flush)
```

- **sep:** Seperator(Ayırıcı, ayraç)
- **end:** Son, Varsayılanı (\n yeni satır veya satır başı)
- **file:** Dosya, (Yazma Nereye yapılacak) (*sys.stdout*)
- **flush:** True-False değerleri alır. Yazdırma işleminin tamponda bekletilmesini ayarlar.

print () Örnekleri

```
>>> print("P", "y", "t", "h", "o", "n", sep=".")
```

P.y.t.h.o.n

```
>>> print(*"Python", sep=".")
```

P.y.t.h.o.n

```
>>> print(*"Çukurova", sep=".")
```

Ç.u.k.u.r.o.v.a

Lütfen Deneyiniz..!

Kaçış Karakterleri / Dizileri

Python'da özel anlam taşıyan işaret veya karakterleri, sahip oldukları bu özel anlam dışında bir amaçla kullanmamızı sağlayan birtakım araçlardır.

- `\n` satır başı
- `\t` sekme
- `\u` Unicode
- `\x` Onaltılı Karakter
- ...