



Python Uygulama Kodları 1

İfade ve Kavram	Açıklama
Bilgisayar Veriyi Nasıl Saklar?	Bilgisayar veriyi hafızada saklar. Her bir değişken için hafızada belirli bir alan ayrılır ve bu alan her seferinde tek bir değer saklayabilir. Kullanıcı, var olan değer yerine yeni bir değer atadığında eski değer silinir. Hafızada bu konumlar geçicidir. Programın çalışması bittiğinde ya da bilgisayar kapatıldığında bu veriler silinir. Verilerin daha sonra tekrar kullanılması gerekiyorsa sabit disk gibi kalıcı bir konuma kaydedilmeleri gerekir. Bu şekilde kaydedilen verilere dosya adı verilir.
Değişken Nedir?	<p>Yazılan programlar bilgisayarın belleği üzerinde çalışır program içerisinde kullanılan her veri için bellekte bir yer ayrılır. Her bir bellek bölgesinin bir adresi vardır. Ancak adreslerle işlem yapmak zordur. Bu nedenle bellekteki verilere erişmek için, verinin tutulduğu bellek bölgesine program içerisinde sembolik isimler verilir. Bir bellek bölgesine işaret eden bu sembolik isimlere değişken adı verilir.</p> <p>Değişkenler programın akışı içerisinde farklı zamanda farklı değerler alabilir. Örneğin yandaki atamalardan sonra a=15 şeklinde bir atama daha gerçekleştirilirse, bellekteki a hücrenin değeri değiştirilerek 15'e ayarlanır.</p>
Değişken Adlandırma Kuralları	<p>1-Değişken adları sayı ile başlamaz.</p> <p>3_kilo_elma = "10 tl" ✗ kilo_elma_3 = "10 tl" ✓</p> <p>2-Değişken adları özel sembol içermez (_ altçizgi hariç)</p> <p>gelir?= "500 TL" ✗ kullanici_adi= "admin" ✓</p> <p>3- Değişken adlarında boşluk olmaz.</p> <p>kullanici adi = "admin" ✗ kulllanici_adi = "admin" ✓</p> <p>4- Değişken adlarında bazı özel anlam ifade eden kelimeler kullanılmaz.</p> <p>Not: Python'da özel anlam ifade eden kelimeleri görmek için aşağıdaki kodları yazın.</p> <pre>['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']</pre> <p>5-Python dili büyük küçük harf duyarlı bir dildir.</p> <pre>sayi=10 Sayi=20 print(sayi+30) 40</pre> <p>6-Değişkenler adlandırılırken İngiliz alfabesindeki karakterler kullanılmalıdır.</p>

BELLEK	
a	3
b	3214.5987
c	Ali

```
a=3  
b=3214.5987  
c="Ali"  
a=15  
print(a)  
15
```



Python Uygulama Kodları 1

Değişkenlere Değer Atama

Not: Değişkenlere değer ataması yaparken = operatörü kullanılır.

Örnek:

```
x=3
```

Not: Bir değere iki veya daha fazla değişken atayabiliriz.

Örnek:

```
>>>a=b=c=4 → a,b ve c değişkenlerinin hepsine birden 4 değeri atandı.
```

```
>>>print(a*b+c) → 4*4+4
```

```
20
```

Not: Bir değişkene defalarca farklı değerler atayabiliriz.

Örnek:

```
>>>x = 10
```

```
>>>x = 20
```

```
>>>x = 30 → Değişkenin en son değeri
```

```
>>>print(x) → Değişkenin en son değeri geçerlidir.
```

```
30
```

Çoklu atama:

Örnek:

```
>>> x, y, z = 4,3,2 → x değişkenine 4, y değişkenine 3, z değişkenine ise 2 değeri atandı.
```

```
>>> print(x*y-z) → 4*3-2
```

```
10
```

Değişken Takası:

Örnek:

```
>>> x,y,z=4,3,2
```

```
>>> x,y,z=y,z,x → x artık y, y artık z, z artık x oldu.
```

```
>>> print(z**y*x) → 4**2*3
```

```
48
```

Değişken İptali: Değişken iptali için komut penceresini kapatıp açabiliriz ya da del komutunu kullanabiliriz.

Örnek:

```
>>>a=2
```

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

```
2
```

```
>>>del a → a değişkeni artık yok yani iptal edildi.
```

```
>>>print(a) → a değişkenini iptal ettiğimiz için kod hata verir. ✘
```



Python Uygulama Kodları 1

ÖRNEK 1.

```
a=6
print(a)
b=3.14159265359
print(b)
c="Merhaba Dünya"
print(c)
```

ÖRNEK 2.

```
3boy=177
boy3=189
print(boy3)
```

ÖRNEK 3.

```
ad-soyad="Ali Can"
ad_soyad="Ali Can"
print(ad_soyad)
```

ÖRNEK 4.

```
x=y=z=3
print(x*y+z)
```

ÖRNEK 5.

```
k,l,m=4,5,2
print(k*l+m)
```

ÖRNEK 6.

```
d,e,f=1,2,3
d,e,f=f,d,e
print(f-d*e)
```

ÖRNEK 7.

```
m="Ali YILMAZ"
print(m)
del m
print(m)
```