Python ile Programlamaya Giriş

DERS 6: DÖNGÜLER

DR. HÜSEYİN BAHTİYAR

Döngü Tipleri
Döngü İçinde Sayma

```
zork = 0
print('Once', zork)
for thing in [9, 41, 12, 3, 74, 15] :
    zork = zork + 1
    print(zork, thing)
print('Sonra', zork)
```

```
$ python countloop.py
Once 0
1 9
2 41
3 12
4 3
5 74
6 15
Sonra 6
```

Döngünün dönüş sayısını saymak için 0 dan başlayan ve döngü içerisine girdiğinde artan bir değişken tanımlamamız gerekmektedir.

Döngüde toplama yapmak

```
zork = 0
print('Once', zork)
for thing in [9, 41, 12, 3, 74, 15] :
    zork = zork + thing
    print(zork, thing)
print('Sonra', zork)
```

```
$ python countloop.py
Once 0
9 9
50 41
62 12
65 3
139 74
154 15
Sonra 154
```

Döngüde toplama yapmak için 0 dan başlayan ve döngü içerisine girdiğinde toplama işlemi yaptığımız bir değişken tanımlamamız gerekmektedir.

Ortalama Bulmak

```
count = 0
sum = 0
print('once', count, sum)
for value in [9, 41, 12, 3, 74, 15] :
    count = count + 1
    sum = sum + value
    print(count, sum, value)
print('Sonra', count, sum, sum / count)
```

```
$ python averageloop.py
Once 0 0
1 9 9
2 50 41
3 62 12
4 65 3
5 139 74
6 154 15
Sonra 6 154 25.666
```

Ortalama hesaplarken sayma ve toplam sayma işlemlerini yapıp döngü sonunda sonuçları birbirine bölüyoruz

Döngü içinde Filtreleme

```
count = 0
print('Once')
for value in [9, 41, 12, 3, 74, 15] :
    if value >20 :
        print(“buyuk sayi”, value)
print('Sonra')
```

```
$ python search1.py
Once
Buyuk sayi 41
Buyuk sayi 74
Sonra
```

Döngüde if kullanarak istediğimiz değerleri (yakalayabilir) filtreleyebiliriz.

Boolean değişkeni ile aramak

```
found = False
print('Once', found)
for value in [9, 41, 12, 3, 74, 15] :
    if value == 3 :
        found = True
        print(found, value)
print('Sonra', found)
```

```
$ python search1.py
Once False
False 9
False 41
False 12
True 3
True 74
True 15
Sonra True
```

Eğer biz bildiğimiz bir değerin var bulunup bulunmadığını öğrenmek istiyorsak önce Yanlış (False) olan bulunduğunda ise Doğru (True) olarak değiştirebileceğimiz bir değişken kullanırız.

En küçük değeri bulmak

```
en_buyuk = -1
print('Once', en_buyuk)
for sayi in [9, 41, 12, 3, 74, 15] :
    if sayi > en_buyuk :
        en_buyuk = sayi
    print(en_buyuk, sayi)

print('Sonra', en_buyuk)
```

```
$ python largest.py
Once -1
9 9
41 41
41 12
41 3
74 74
74 15
Sonra 74
```

Bunu en küçük sayı için yapabilir miyiz?

En küçük değeri bulmak

```
en_kucuk = -1
print('Once', en_kucuk)
for sayi in [9, 41, 12, 3, 74, 15] :
    if sayi < en_kucuk :
        en_kucuk = sayi
    print(en_kucuk, sayi)

print('Sonra', en_kucuk)
```

- Değişken adını en_kucuk olarak değiştirdik
- if kısmındaki işareti > yerine < yaptık

En küçük değeri bulmak

```
en_kucuk = -1
print('Once', en_kucuk)
for sayi in [9, 41, 12, 3, 74, 15] :
    if sayi < en_kucuk :
        en_kucuk = sayi
    print(en_kucuk, sayi)

print('Sonra', en_kucuk)
```

Olmadı :(
Peki ne yapmalıyız?

```
$ python small bad.py
Once -1
-1 9
-1 41
-1 12
-1 3
-1 74
-1 15
Sonra -1
```

En küçük değeri bulmak

```
en_kucuk = None
print('Once', en_kucuk)
for sayi in [9, 41, 12, 3, 74, 15] :
    if en_kucuk is None :
        en_kucuk = sayi
    elif sayi < en_kucuk :
        en_kucuk = sayi
    print(en_kucuk, sayi)
print('Sonra', en_kucuk)
```

```
$ python smallest.py
Once
9 9
9 41
9 12
3 3
3 74
3 15
Sonra 3
```

Halen elimizde enkucuk isimli değişkenimiz var, döngüye ilk girerken enkucuk değişkenin değeri yok, bu sebeple ilk değeri enküçük olarak atıyoruz.

is ve is not Operatörleri

```
en_kucuk = None
print('Once', en_kucuk)
for sayi in [9, 41, 12, 3, 74, 15] :
    if en_kucuk is None :
        en_kucuk = sayi
    elif sayi < en_kucuk :
        en_kucuk = sayi
    print(en_kucuk, sayi)
print('Sonra', en_kucuk)
```

- Python daki is operatörü mantıksal işlemlerde kullanılabilir.
- ifadenin iki tarafındaki değişkenlerin de “aynı mı?” olduğunu kontrol eder
- == ile benzer işlemleri yaparken daha güçlü özellikleri vardır.
- is not operatörü de mantıksal bir operatördür.

Ödev

- Klavyeden “bitti” girilene kadar klavyeden sayı girmesini isteyen program yazınız.
  - Ayrıca bitti yerine farklı birşey yazıldığında hata versin.

Acknowledgements / Contributions

Initial Development: Charles Severance, University of Michigan School of Information

… Insert new Contributors and Translators here