

Python 101

Ders 2:

+ Koşullar

+ Döngüler

Eğitmen : Şevket Aşurgan

İletişim: sevketasurgan@gmail.com

Koşullar(Conditions)

Programların belirli koşullara göre farklı davranışlar sergilemesini sağlar.

Koşullar, "if", "else" ve "elif" anahtar kelimeleri ile belirtilir.

Bir koşul, "True" ya da "False" olarak değerlendirilir.

Koşullar(Conditions)

```
x = 10  
y = 5  
  
if x > y:  
    print("x, y'den büyüktür")  
else:  
    print("x, y'den küçüktür veya eşittir")
```

Koşul Blokları

Koşul ifadesi "True" ise "if" bloğu çalıştırılır.



```
graph TD; A[Koşul ifadesi "True" ise "if" bloğu çalıştırılır.] --> B[Koşul ifadesi "False" ise "if" bloğu atlanır ve "else" bloğu çalıştırılır.]; B --> C[Birden fazla koşul için "elif" blokları kullanılabilir.];
```

Koşul ifadesi "False" ise "if" bloğu atlanır ve "else" bloğu çalıştırılır.

Birden fazla koşul için "elif" blokları kullanılabilir.

Mantıksal Operatörler

"and" : İki koşulun da doğru olması durumunda "True" döndürür.

"or" : İki koşuldan en az birinin doğru olması durumunda "True" döndürür.

"not" : Koşulun tersini döndürür.

Karşılaştırma Operatörleri

- "==" : Eşitlik kontrolü
- "!=" : Eşitlik dışı kontrolü
- ">" : Büyüklük kontrolü
- ">=" : Büyük veya eşitlik kontrolü
- "<" : Küçüklük kontrolü
- "<=" : Küçük veya eşitlik kontrolü

Örnek Kullanım

```
if x > 0:  
    print("x, 0'dan büyüktür.")  
elif x < 0:  
    print("x, 0'dan küçüktür.")  
else:  
    print("x, 0'a eşittir.")
```

Indentation(Girintileme)

Python'da indentation (girintileme), kod bloklarının kapsamını belirtmek için kullanılır. Bu özellik özellikle koşullarda önemlidir, çünkü belirli bir koşul altında çalıştırılacak olan kod, o koşulun altında girintili olarak yazılır.

Örnek Kullanım

```
x = 10 y = 5
if x > y:
    print("x y'den büyüktür")
else:
    print("y x'ten büyüktür")
```

Indentation(Girintileme)

Girintileme Python'da hayati öneme sahiptir ve kodun tümünde tutarlı olmalıdır. Yanlış girintileme hatalara veya beklenmedik davranışlara neden olabilir.

Yanlış Kullanım

```
if x > y:  
    print("x y'den büyüktür")  
else:  
    print("y x'ten büyüktür")
```

Döngüler

Python'da döngüler, belirli bir kod bloğunun belirli bir koşul veya döngü değişkeni altında tekrar tekrar çalıştırılmasını sağlar.

Döngüler

1. While Döngüsü

- While döngüsü, belirli bir koşul doğru olduğu sürece tekrarlanan bir döngüdür. Döngü, koşul yanlış olduğunda durur. Bu döngü, bir şeyi belirli bir sayıda kez yapmak yerine, belirli bir koşul gerçekleşene kadar tekrar etmek istediğimiz zaman kullanışlıdır.

Örnek Kullanım

```
x = 0
while x < 5:
    print(x)
    x += 1
```

Döngüler

- While döngüsü, doğru koşul yanlış olduğunda durur. Bu nedenle, koşul yanlış olana kadar döngü devam eder. Bu nedenle, bir koşul yanlış yazılırsa, döngü sonsuza kadar devam edebilir. Bu durumda, döngüyü sonlandırmak için Ctrl + C tuşlarına basmanız gerekir.

Döngüler

2. For Döngüsü

- For döngüsü, bir dizi, liste, demet vb. gibi bir veri yapısı üzerinde yineleyen bir döngüdür. For döngüsü, her yinelemede bir sonraki öğeyi alır ve öğe sayısı kadar yineleme yapar.

Örnek Kullanım

```
fruits = ["elma", "armut", "çilek"]  
for fruit in fruits:  
    print(fruit)
```

Döngüler

- For döngüsü ayrıca `range()` fonksiyonu ile birlikte kullanılabilir. `range()` fonksiyonu, belirli bir aralıktaki sayıları içeren bir dizi oluşturur.

Örnek Kullanım

```
for i in range(1, 6):  
    print(i)
```

sevketasurgan@gmail.com
