

ANR 413 Bilgisayar Programlama

Python

#7

Serdar ARITAN

Biyomekanik Araştırma Grubu,
Spor Bilimleri Fakültesi
Hacettepe Üniversitesi
Ankara





Python'da hataları yakalamak için **try**, **except** ve **finally** kod bloklarından faydalanırız.

- **try**, bir kod bloğunu hatalara karşı denetler.
- **except**, kod bloğundaki bir hata durumunda işlemler yapmayı sağlar.
- **finally**, **try-except** sonucu fark etmeksizin işlem hata tespit işlemi sonrasında çalıştırılan kod bloğudur.

Normalde uygulamamızda bir hata meydana geldiğinde Python hata mesajı yazdırır ve **uygulamayı durdurur**. `try` kodu ile bir kod bloğunu hatalara karşı denetleme imkanı buluruz. Aşağıdaki kod çalıştırıldığı zaman `x` tanımsız olduğu için uygulamamız hata verecektir. Bu hata içeren kodu `try:` bloğu içinde yazdık. Hata verdiği zaman `except:` bloğu çalıştırılacak ve oradaki mesaj görüntülenecektir.

`try:`

```
    print(x)
```

`except:`

```
    print("Bir hata oluştu")
```

Bu durumda **hatayı yakalamış** ve ona göre işlem yapmış olduk. **Uygulamamız halen çalışıyor**. Eğer `try-except` bloklarını kullanmasaydık hata nedeniyle uygulama sona erecekti.



IDLE Shell 3.10.0

File Edit Shell Debug Options Window Help

```
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [
MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more i
nformation.
>>> try:
...     print(x)
... except:
...     print("Bir hata oluřtu")
...
Bir hata oluřtu
>>> print(x)
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    print(x)
NameError: name 'x' is not defined
>>> |
```

Ln: 14 Col: 0



Hatalar farklı farklı nedenlerden dolayı meydana gelebilir. Bu durumda hatanın nedenini tespit etmeye yönelik birden fazla **except** bloğu yazabiliriz. İstisnalar her zaman farklı bir nedenden kaynaklanan hataları yakalamaya yönelik çalışacaktır.

```
try:  
    print(x)  
except NameError:  
    print("x tanımlanmamış")  
except:  
    print("Başka bir hata meydana geldi")
```



```
IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18)
[MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more
information.
>>> try:
...     print(x)
... except NameError:
...     print("x tanımlanmamış")
... except:
...     print("Başka bir hata meydana geldi")
...
x tanımlanmamış
>>> try:
...     print(1/0)
... except NameError:
...     print("x tanımlanmamış")
... except:
...     print("Başka bir hata meydana geldi")
...
Başka bir hata meydana geldi
>>> |
```

Ln: 19 Col: 0

```
sayi1 = int(input("Birinci Sayı :"))  
sayi2 = int(input("İkinci Sayı :"))
```

```
try:
```

```
    sonuc = sayi1/sayi2  
    print("Sonuc :",sonuc)
```

```
except ZeroDivisionError:
```

```
    print("Bir sayıyı sıfıra bölemezsiniz lütfen sıfır dışında bir sayı girin")
```

```
Birinci Sayı :5
```

```
İkinci Sayı :0
```

```
Bir sayıyı sıfıra bölemezsiniz lütfen sıfır dışında bir sayı girin
```

Başka bir durumda, kullanıcı sayı dışında bir karakter girerse o zaman ne farklı bir istisna ile karşılaşacağız;

```
Birinci Sayı :8
```

```
İkinci Sayı :d
```

```
Traceback (most recent call last):
```

```
File
```

```
"C:/Lectures/ANR_413_Bilgisayar_Programlama/hata_1.py",  
line 2, in <module>
```

```
sayi2 = int(input("İkinci Sayı :"))
```

```
ValueError: invalid literal for int() with base 10: 'd'
```

```
sayi1 = int(input("Birinci Sayı :"))  
sayi2 = int(input("İkinci Sayı :"))
```

```
try:  
    sonuc = sayi1/sayi2  
    print("Sonuc :",sonuc)  
  
except (ZeroDivisionError,ValueError) :  
    print("Lütfen girdiğiniz değerlere dikkat edin")
```

```
Birinci Sayı :8  
İkinci Sayı :d  
Lütfen girdiğiniz değerlere dikkat edin
```

Burada bir belirsizlik var kullanıcı acaba hangi hatayı yaptım diye düşecektir. Bu yüzden hatları ayrı bloklar içine almamız doğru olacaktır.

```
sayi1 = int(input("Birinci Sayı :"))  
sayi2 = int(input("İkinci Sayı :"))
```

```
try:  
    sonuc = sayi1/sayi2  
    print("Sonuc :",sonuc)  
  
except ZeroDivisionError:  
    print("Lütfen ikinci sayıya sıfırdan farklı bir değer girin.")  
  
except ValueError:  
    print("Lütfen sayısal bir karakter girin.")
```

Bütün olası hataları yani istisnaları nereden bileceğiz? Hepsini tekrar girmek zorunda mıyız. acaba!

```
sayi1 = int(input("Birinci Sayı :"))  
sayi2 = int(input("İkinci Sayı :"))
```

```
try:  
    sonuc = sayi1/sayi2  
    print("Sonuc :",sonuc)  
  
except ZeroDivisionError:  
    print("Lütfen ikinci sayıya sıfırdan farklı bir değer girin.")  
  
except ValueError:  
    print("Lütfen sayısal bir karakter girin.")  
  
except:  
    print("Beklenmeyen bir hata oluştu")
```

Hata yakalama mekanizmanı olan VKİ hesabı yapan bir program yazınız; VKİ hesaplaması ve durum bilgisi fonksiyon olarak yazılacak.

```
Lütfen Boy Uzunluğunuzu Giriniz [m]: 1.83
Lütfen Vücut Ağırlığınızı Giriniz [kg]: 104
Vücut kitle indeksiniz 31.05497327480665 ve siz ' Şişman ' kategorisindesiniz.
Lütfen Boy Uzunluğunuzu Giriniz [m]: -1.83
Lütfen Vücut Ağırlığınızı Giriniz [kg]: 104
Lütfen negatif sayı girmeyiniz.
Lütfen Boy Uzunluğunuzu Giriniz [m]: 0
Lütfen Vücut Ağırlığınızı Giriniz [kg]: 10
Lütfen boy için sıfır girmeyiniz.
Lütfen Boy Uzunluğunuzu Giriniz [m]: 1.83
Lütfen Vücut Ağırlığınızı Giriniz [kg]: yüzdört
Lütfen sayısal bir karakter girin.
```