



PROGRAMLAMA TEMELLERİ

Öğr. Gör. Erhan AKAGÜNDÜZ

HATA YAKALAMA İŞLEMLERİ

Hata Nedir?

- ❑ Programlar, -özellikle başlangıç seviyesinde- genellikle en iyi duruma göre yazılır.
- ❑ Yani tüm yazılım ve donanım kaynaklarının beklenen şekilde çalışacağı ve kullanıcıların programı yazılımcının ondan beklediği şekilde kullanacağı varsayılır.
- ❑ Benzer şekilde bir hesaplama işleminin her durumda doğru çalışması beklenir.
- ❑ Ancak özellikle profesyonel seviyede, bu asla olmaması gereken bir durumdur.

HATA YAKALAMA İŞLEMLERİ

Hata Nedir? devamı

- ❑ Programcının bütün iyi niyetli yaklaşımına rağmen işler her zaman istenildiği gibi gitmeyebilir.
- ❑ Bu nedenle iyi bir programcı; yazılım, donanım ve kullanıcı kaynaklı birçok hatayla karşılaşacağını bilmeli ve bunlara yönelik önlemlerini almalıdır.
- ❑ Programın asla kendi kontrolü dışında sonlanmasına izin vermemelidir.
- ❑ En kötü durumda bile kullanıcıların anlayabileceği hata mesajları vererek programı sonlandırmalıdır.

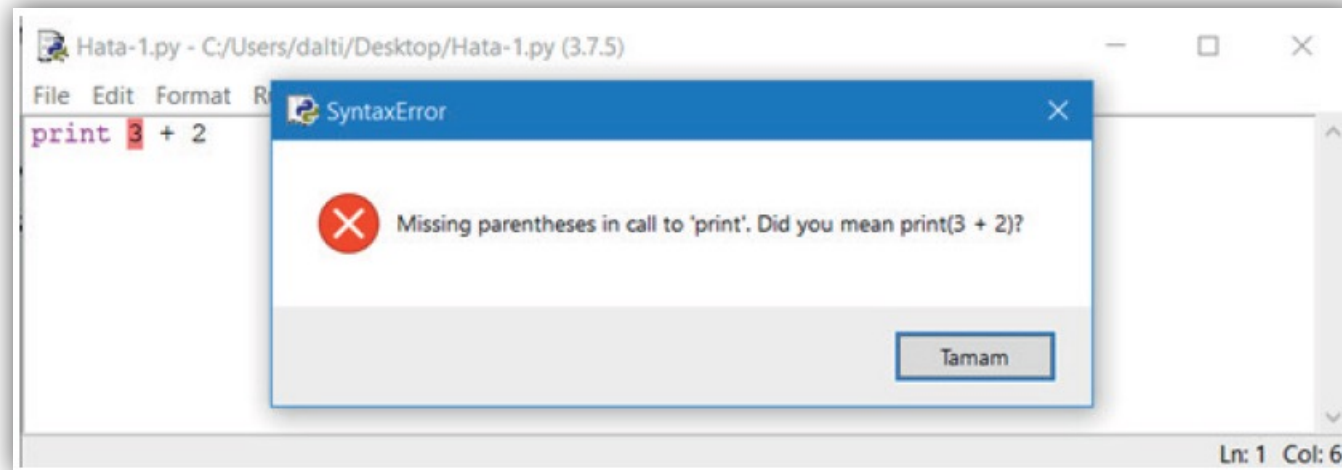
HATA TÜRLERİ

- ❑ Hata kavramının sadece program dışı parametrelerde oluşabileceği düşünülmemelidir.
- ❑ Hataları üçe ayırmak mümkündür:
 1. Programcı hataları / Yazım hataları (Syntax errors)
 2. Mantıksal hatalar (Bugs)
 3. İstisnai hatalar (Exceptions)

HATA TÜRLERİ

Programcı Hataları/Yazım Hataları

- ❑ Programı yazan kişiden kaynaklanan hatalardır. Çoğunlukla dikkatsizlik sonucu oluşur.



HATA TÜRLERİ

Mantıksal hatalar (Bugs)

- ❑ Bu hata türünün, tespiti ve çözülmesi daha zordur.
- ❑ Çünkü program hata vermeden çalıştığı hâlde, programda hesaplanan sonuçlar yanlıştır.
- ❑ Özellikle programdaki satır sayısı arttıkça bu tür hataların tespiti de zorlaşmaktadır.
- ❑ Bu tür hatalara **bug** (böcek), ilgili hatayı bulup düzeltme işlemine de **debug** (ayıklama) denir.

HATA TÜRLERİ

İstisnai Hatalar

- ❑ Bu tür hatalar, programın çalışması esnasında oluşan, aslında gerçekleşmesi beklenmeyen hatalardır.
- ❑ Örneğin; kullanıcıdan bir sayı alıp karesini ekrana yazan bir program yazılsın ama kullanıcı tam sayı yerine «a» harfini yazsın.

HATA YAKALAMA

- ❑ Python programlama dilinde hata yakalama **try-except** blokları aracılığıyla yapılmaktadır.

```
print("Program başladı")
try:
    sayi = int(input("Bir sayı yazınız: "))
    karesi = sayi * sayi
    print("Sayının karesi:", karesi)
except:
    print("Bir hata oluştu !!!")
print("Program sona erdi!")
```

```
Program başladı
Bir sayı yazınız: a
Bir hata oluştu !!!
Program sona erdi!
```

```
print("Program başladı")
try:
    sayi = int(input("Bir sayı yazınız: "))
    karesi = sayi * sayi
    print("Sayının karesi:", karesi)
except:
    print("Bir hata oluştu !!!")
print("Program sona erdi!")
```

```
Program başladı
Bir sayı yazınız: 9
Sayının karesi: 81
Program sona erdi!
```


HATA YAKALAMA

“as” İfadesi ile Orijinal Hata Mesajı Gösterme

- ❑ except bloğunda istenilen hata mesajı gösterilebildiği gibi Python tarafından oluşturulan orijinal hata mesajı da gösterilebilir.
- ❑ Bunun için **as** deyimi kullanılır.

```
try:
    sayi1 = int(input("1. sayı: "))
    sayi2 = int(input("2. sayı: "))
    toplam = sayi1 + sayi2
    print("Toplam:", toplam)
except ValueError as hata:
    print("Sayı girmediniz!")
    print("Orijinal hata mesajı:", hata)
```

```
1. sayı: a
Sayı girmediniz!
Orijinal hata mesajı: invalid literal for int() with base 10: 'a'
```