

Errors_Exceptions(Xətalər və İstisnalar)

May 23, 2022

0.1 Errors and Exceptions(Xətalər və İstisnalar)

Python proqramlaşdırma dilində yazılan kod nümunələrində az-çox hər kəs səhv edərək öyrənir. Python dilində bu səhvlər əsasən iki qismdə 1. sintaksis səhvlər 2. istisnalar olaraq iki yerə ayrılır. Məsələn input funksiyası bilirikki istifadəçidən aldığı parametri sətir tipində saxlayır və biz bunu riyazi əməllərdə istifadə etdiyimiz zaman məcburi həqiqi və ya tam ədəd tipinə çeviririk. Və istifadəçi xüsusi işarələr yazma bilməsi ehtimalı olduğundan biz xəta ilə üzləşərik və bu da proqramın sonlandırılması deməkdir. Bunun üçün biz proqramı davamlı çalışdırıb çıxış əmri əlavə edərək həmçinin baş verəcək xətalara özəlləşdirərək istifadəçiyə bildiriş göndərərək yenidən ədəd tipində dəyər yazmasını tələb edə bilərik. Bu bəhsdə məhz biz bu xətalara tiplərini və özəlləşdirmə üsullarını nəzərdən keçirəcəyik. Misallardan bu daha aydın olacaq

```
[2]: var = input('dəd yazın:')
      print('*'*50)
      print('Cavab:', var+12)
```

dəd yazın:12

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-2-15798c81f55c> in <module>()
      1 var = input('dəd yazın:')
      2 print('*'*50)
----> 3 print('Cavab:', var+12)

TypeError: must be str, not int
```

Nə oldu! xəta aldım. TypeError: must be str, not int sətiri xəta sətiri olub xəta tipi TypeError adlanır. sətir tipi ilə tam ədəd tipini toplaya bilmərik.

Bu tip xətalardan başqa python dilində digər xətalara da mövcuddur. Məsələn

```
[3]: 1/0
```

```
-----
ZeroDivisionError                        Traceback (most recent call last)
<ipython-input-3-9e1622b385b6> in <module>()
1/0
```

```
----> 1 1/0
```

```
ZeroDivisionError: division by zero
```

```
[4]: 4 + num*3
```

```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-4-d5d588e5d103> in <module>()  
----> 1 4 + num*3
```

```
NameError: name 'num' is not defined
```

Və digərləri.Bu xətalrı əzbərləməyinizə ehtiyac yoxdur.Bir neçə xətalara dair nümunələr hazır-
ladığımdan dolayı mövzunu mənimsəyəcəksiniz.

```
[5]: print("""  
  
try:  
    kodlar.....  
except:  
    print(istifadəçiyə mesaj göndər.....)  
  
""")
```

```
try:  
    kodlar...  
except:  
    print(istifadəçiyə mesaj göndər...)
```

Yuxarıda nümunə üçün xəta və istisnaların yazılış qaydasını göstərdim.Prinsip etibarı ilə iki hissədən
try bloku daxilində çalışacaq kodlarımız və müstəsna halda baş verəcək xəta,digəri exception bloku
daxilinə baş verəcək xətalrı istifadəçiyə bildiriş formasında göndərmək.İrəliləyən dərslərdə digər
yazılış sintaksislərini göstərəcəm

```
[23]: def Fahrenheit(temprature=input('Celsius:')):  
    """  
    funksiya selsi cinsindən temperaturu farnheytt cinsinə çevirir  
    """  
    F = (temprature * 9/5) + 32  
    return ('Converted :{:}.2f}F'.format(F))  
Fahrenheit()
```

Celsius:12

```
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-23-2822a7fea8f7> in <module>()  
      5     F = (temprature * 9/5) + 32  
      6     return ('Converted {:.2f}F'.format(F))  
----> 7 Fahrenheit()  
      8  
  
<ipython-input-23-2822a7fea8f7> in Fahrenheit(temprature)  
      3     funksiya selsi cinsindən temperaturu farnheyte cinsinə çevirir  
      4     """  
----> 5     F = (temprature * 9/5) + 32  
      6     return ('Converted {:.2f}F'.format(F))  
      7 Fahrenheit()  
  
TypeError: unsupported operand type(s) for /: 'str' and 'int'
```

Yuxarıda funksiya tərtib etdik və xəta aldığımız. Bilirikki input funksiyasından geri döən dəyərlər sətir tipində(string) olduğu üçün hesablama doğru aparılmadı

```
[24]: def Fahrenheit(temprature=input('Celsius:')):  
      """  
      funksiya selsi cinsindən temperaturu farnheyte cinsinə çevirir  
      """  
      try:  
          F = (temprature * 9/5) + 32  
          return ('Converted {:.2f}F'.format(F))  
      except:  
          print('Xəta')  
      Fahrenheit()
```

Celsius:12

Xəta

Nəticədə funksiya daxilində try bloku hazırladıq.Və ardından except blokuna əgər xəta baş verərsə print() funksiyası ilə istifadəçiyə bildiriş göndərdik

və ya

```
[30]: def Fahrenheit(temprature):  
      """  
      funksiya selsi cinsindən temperaturu farnheyte cinsinə çevirir  
      """  
      try:  
          F = (int(temprature) * 9/5) + 32  
          return ('Converted {:.2f}F'.format(F))
```

```

except ValueError:
    print('Xəta.....')
Fahrenheit(input('Celsius:'))

```

Celsius:!!!!--as
Xəta...

[31]: 1/0

```

-----
ZeroDivisionError                                Traceback (most recent call last)
<ipython-input-31-9e1622b385b6> in <module>()
----> 1 1/0

ZeroDivisionError: division by zero

```

Yuxarıdakı xətanın tipi ZeroDivisionError -dur. Misal olaraq hesablayıcı proqram hazırlasanız istədiyi 0-a bölmə əməlini yazarsa xəta ilə nəticələnib proqram sonlandırılacaq. Biz öncədən bu xətanı qabaqlaya bilərik

```

[51]: def toplama(a,b):
        """
        funksiya toplama əməlini yerinə yetirir
        """
        return ('Cavab:{}'.format(a+b))
def çıxma(a,b):
        """
        funksiya çıxma əməlini yerinə yetirir
        """
        return ('Cavab:{}'.format(a-b))
def vurma(a,b):
        """
        funksiya vurma əməlini yerinə yetirir
        """
        return ('Cavab:{}'.format(a*b))
def bölme(a,b):
        """
        funksiya bölme əməlini yerinə yetirir
        """
        return ('Cavab:{}'.format(a/b))

```

Yuxarıda 4 sayda riyazi əməliyyatı hesablayacaq funksiya hazırladıq

```

[52]: emeliyyat = input('məliyyatı seçin:')
a = input('Birinci ədəd:')
b = input('İkinci ədəd:')
try:

```

```

if emeliyyat == 'toplama':
    print(toplama(float(a),float(b)))
elif emeliyyat == 'çıxma':
    print(çıxma(float(a),float(b)))
elif emeliyyat == 'vurma':
    print(vurma(float(a),float(b)))
elif emeliyyat == 'bölmə':
    print(bölmə(float(a),float(b)))
except ZeroDivisionError:
    print('0-a bölmə xətası')

```

```

məliyyatı seçin:toplama
Birinci ədəd:12
İkinci ədəd:3
Cavab:15.0

```

```

[53]: emeliyyat = input('məliyyatı seçin:')
a = input('Birinci ədəd:')
b = input('İkinci ədəd:')
try:
    if emeliyyat == 'toplama':
        print(toplama(float(a),float(b)))
    elif emeliyyat == 'çıxma':
        print(çıxma(float(a),float(b)))
    elif emeliyyat == 'vurma':
        print(vurma(float(a),float(b)))
    elif emeliyyat == 'bölmə':
        print(bölmə(float(a),float(b)))
except ZeroDivisionError:
    print('0-a bölmə xətası')

```

```

məliyyatı seçin:bölmə
Birinci ədəd:3
İkinci ədəd:0
0-a bölmə xətası

```

[]:

0.1.1 except as xəta

Yuxarıdakı nümunələrdə hər bir xətanı əldə edə bilirdik. Bu hissədə isə həm xətaları cəmləşdirməyi həm də xətanı bir simvola mənimsəyib çap edə bilərik.

```
[6]: try:
    num=int(input('bir rəqəm yazın:'))
    num1=int(input('bir rəqəm yazın:'))
    print('Result:',num/num1)
except ZeroDivisionError as err:
    print('Xəta oldu,xəta tipi: {}'.format(err))
```

```
bir rəqəm yazın:12
bir rəqəm yazın:4
Result: 3.0
```

```
[7]: try:
    num=int(input('bir rəqəm yazın:'))
    num1=int(input('bir rəqəm yazın:'))
    print('Result:',num/num1)
except ZeroDivisionError as err:
    print('Xəta oldu,xəta tipi: {}'.format(err))
```

```
bir rəqəm yazın:4
bir rəqəm yazın:0
Xəta oldu,xəta tipi: division by zero
```

Yuxarıdakı nümunədə xətanı err ifadəsinə mənimsətdik

Bundan başqa xətaları toplu şəkildə əldə etmək üçün

```
[8]: try:
    num=int(input('bir rəqəm yazın:'))
    num1=int(input('bir rəqəm yazın:'))
    print('Result:',num/num1)
except (ValueError,ZeroDivisionError) as err:
    print('Xəta oldu,xəta tipi: {}'.format(err))
```

```
bir rəqəm yazın:
Xəta oldu,xəta tipi: invalid literal for int() with base 10: ''
```

Heç bir ifadə yazmadan enter düyməsini sıxdıqda ValueError xətası aldığımız üçün except bloku işə düşərək err ifadəsi ilə xətanı əldə edirik daha sonra print() funksiyası ilə ekrana çap edə bildik

```
[ ]: while True:
    try:
        num=int(input('bir rəqəm yazın:'))
        num1=int(input('bir rəqəm yazın:'))
    except ValueError:
        print('sadəcə rəqəm daxil edin')
    else:
        try:
            print(num/num1)
        except ZeroDivisionError:
```

```
print('0-a bölmək olmaz')
```

```
bir rəqəm yazın:12
bir rəqəm yazın:5
2.4
bir rəqəm yazın:3
bir rəqəm yazın:0
0-a bölmək olmaz
bir rəqəm yazın:11
bir rəqəm yazın:
sadəcə rəqəm daxil edin
bir rəqəm yazın:4
bir rəqəm yazın:34
0.11764705882352941
bir rəqəm yazın:q
sadəcə rəqəm daxil edin
```

Yuxarıdakı nümunədə else operatorundan istifadə edərək xəta baş verəcək halda ayrı ayrı blokları çalışdıraraq nəticələri əldə edə bildik.

[]: