

# Python programming language

Week 11. Python Try...Except, Python User Input

**Alisher Ismailov**

Lecturer

Department of Information technology  
Andijan Branch of Tashkent Financial Institute

Email: [alisherismailov534@gmail.com](mailto:alisherismailov534@gmail.com)

# Python dasturlash tili

11-Mavzu. Python Try Except istinolari

**Ma'ruzachi: Alisher Ismoilov**

Axborot texnologiyalari kafedrası

Toshkent moliya instituti Andijon filiali

Elektron pochta: [alisherismailov534@gmail.com](mailto:alisherismailov534@gmail.com)

# 11-Mavzu. Python Try Except istinolari

## Reja:

- 🐍 Istisno (exception) bilan ishlash
- 🐍 Ko‘p istisnolar (except) bilan ishlash
- 🐍 Else kalit so‘zidan foydalanish
- 🐍 Finally kalit so‘zidan foydalanish
- 🐍 Exception (Istisno)ni raise (ko‘tarish)
- 🐍 Foydalanuvchi ma’lumot kiritishi

# Python Try Except istinolari

**Try** va **Except** iboralarini Python dasturlash tilidagi koddagi xatolarni hal qilish uchun ishlatiladi. Try bloki kodlarni xatolar uchun tekshirish uchun ishlatiladi, ya'ni dasturda xato bo'lmaganda try blokidagi kod ishlaydi. **Except** blokidagi kod, dasturda biron bir xatoga duch kelganda bajariladi.

# Python Try Except istinolari





**Try** – kod blokini xatolarni topish imkonini beradi.

**Except** – bloki xato bo'lganda dsturchiga kodni to'xtab qolishini oldini olish imkon beradi.

**Else** – bloki xato bo'lmaganda kodni bajarishga imkon beradi.

**Finally** – bloki sinash natijasidan qat'iy nazar bajariladi.

# Python Try Except istinolari

-  **Try** – kod blokini xatolarni topish imkonini beradi.
-  **Except** – bloki xato bo‘lganda dsturchiga kodni to‘xtab qolishini oldini olish imkon beradi.
-  **Else** – bloki xato bo‘lmaganda kodni bajarishga imkon beradi.
-  **Finally** – bloki sinash natijasidan qat’iy nazar bajariladi.

# Python Try Except istinolari

## Istisno (exception) bilan ishlash

- **Xato sodir bo'lganda, Python kodi odatda ishlashdan to'xtaydi va xato xabarini ko'rsatadi. Ushbu istisnolarni try kalit so'zi yordamida hal qilish mumkin. Quyidagi misolda try bloki kodi bajarilmaydi, chunki x o'zgaruvchisi qiymati mavjud emas:**

# Python Try Except istinolari

```
try:  
    print(x)  
except:  
    print("Xatolik yuz berdi")
```

# Python Try Except istinolari

**try** blokida xato aniqlanganligi sababli, **except** blokini kodi ishga tushdi. **Try** bloki bo'lmasa, dastur ishdan chiqadi va xato xabari ko'rsatiladi. Try va Except kalit so'zlarini o'chirib yuqoridagi kodni tekshirib ko'ramiz:

```
In [2]: try:  
        print(x)  
except:  
        print("Xatolik yuz berdi")
```

```
Xatolik yuz berdi
```

# Python Try Except istinolari

```
In [3]: print(x)
```

```
-----  
NameError                                Traceback  
~\AppData\Local\Temp\ipykernel_2952\289819153.py in  
----> 1 print(x)
```

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

# Python Try Except istinolari

Yuqoridagi muammoni yechish uchun yoki kodni to'g'rilash uchun x o'zgaruvchisini yaratish va unga qiymat berish kerak. Agar o'zgaruvchiga qiymat berishni xoxlamasangiz berilganlar turiga qarab **null** qiymatini bering. Masalan, string o'zgaruvchisini qiymati bo'lmasa o'zgaruvchiga "" qiymatini berish mumkin, ya'ni ushbu o'zgaruvchi qiymati bo'sh degan ma'noni anglatadi. Agar int yoki float berilgan turiga oid o'zgaruvchi bo'lsa unga 0 qiymati berish mumkin.

# Python Try Except istinolari

```
x = ''
```

```
print(x)
```

natija bo'sh bo'ladi lekin kod xatosiz ishga tushadi.

Try va except orqali tekshirib ko'rsak, try bloki kodida xatolik aniqlanmaydi:

```
x = ''
```

```
try:
```

```
    print(x)
```

```
except:
```

```
    print("Xatolik yuz berdi")
```

# Python Try Except istinolari

natija bo‘sh bo‘ladi lekin kod xatosiz ishga tushadi. x o‘zgaruvchisiga boshqa qiymat bersangiz, dastur o‘sha qiymatni natija sifatida ko‘rsatadi. Masalan, x o‘zgaruvchisiga ‘dasturlash’ qiymati beramiz va natijani tekshiramiz.

```
x = ‘dasturlash’
```

```
try:
```

```
    print(x)
```

```
except:
```

```
    print(“Xatolik yuz berdi”)
```

# Python Try Except istinolari

```
x = 'dasturlash'  
  
try:  
    print(x)  
except:  
    print("Xatolik yuz berdi")
```

dasturlash

# Python Try Except istinolari

## Ko'p istisnolar (except) bilan ishlash

- Istisno (except) bloklarini xoxlagancha ishlatish mumkin. Masalan, agar o'zgaruvchiga qiymat berilmagan bo'lsa uni NameError istisnosi orqali aniqlash mumkin. Quyidagidek yoziladi:
- try:
- print(x)
- except NameError:
- print("x o'zgaruvchisi yaratilmagan")
- except:
- print("Boshqa xatolik yuz berdi")

# Python Try Except istinolari

## Else kalit soʻzidan foydalanish

- **Yozilgan kodda xatolik boʻlmasa, bajariladigan kod blokini aniqlash uchun else kalit soʻzidan foydalanish mumkin:**
- **Ushbu misolda try blokida hech qanday xato yoʻq:**

# Python Try Except istinolari

```
In [5]: try:  
        print("Salom")  
    except:  
        print("Xatolik yuz berdi")  
    else:  
        print("Bexato ishlar qda")
```

Salom

Bexato ishlar qda

# Python Try Except istinolari

- Agar **try** blokida xatolik yuzaga kelsa **else** bloki ishga tushmaydi.

```
try:
    print(x)
except NameError:
    print("x o'zgaruvchisi yaratilmagan")
except:
    print("Boshqa xatolik yuz berdi")
else:
    print("Bexato ishlamoqda")
```

```
x o'zgaruvchisi yaratilmagan
```

# Python Try Except istinolari

## Finally kalit soʻzidan foydalanish

- `try` blokida xatolik yuzaga keladimi yoki yoʻqmi `finally` bloki ishga tushadi.

```
In [8]: try:
        print(x)
        except:
        print("Xatolik yuz berdi")
        finally:
        print("'try except' yakunlandi")
```

```
Xatolik yuz berdi
'try except' yakunlandi
```

# Python Try Except istinolari

- Ushbu istisnolarni real hayotdagi muammolarga qo‘llash foydalidir. Quyidagi misolda, yozish mumkin bo‘lmagan faylni ochishga va unga ma’lumot yozishga harakat qiladigan dastur yozamiz va uni istisno orqali xatosini oldini olamiz:

```
In [10]: try:
          f = open("demofile.txt")
          try:
              f.write("Yangi qator")
          except:
              print("Faylga yozishda xatolik yuz berdi")
          finally:
              f.close()
      except:
          print("Faylni ochishda xatolik yuz berdi")
```

```
Faylni ochishda xatolik yuz berdi
```

# Python Try Except istinolari

## Exception (Istisno)ni raise (ko'tarish)

- Python dasturchisi sifatida, agar biror holat yuzaga kelsa, istisno qilishni tanlash mumkin.
- Istisnoni tashlash uchun **raise** kalit so'zidan foydalanish mumkin. Quyidagi misolda xatoni nimaligini dasturchi o'zi tushunadigan qilib belgilashi mumkin.
- Xatoni **raise** qilish, agar **x o'zgaruvchisining qiymati 20** dan past bo'lsa, dastur to'xtaydi:

# Python Try Except istinolari

```
x = 10

if x < 20:
    raise Exception("kiritilgan qiymat 20 dan katta bo`lishi shart")

-----
Exception                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_11240\4011371985.py in <module>
      2
      3 if x < 20:
----> 4     raise Exception("kiritilgan qiymat 20 dan katta bo`lishi shart")

Exception: kiritilgan qiymat 20 dan katta bo`lishi shart
```

# Python Try Except istinolari

- Agar if holati to'g'ri bo'lsa ya'ni **x o'zgaruvchisi qiymati 20** dan baland bo'lsa hech qanday xatosiz ishlaydi.
- **Raise** kalit so'zi istisnoni ko'tarish uchun ishlatiladi. Dasturchi qanday xatoni ko'tarishni va matnini foydalanuvchiga chop etishni belgilashi mumkin.

```
In [14]: x = 40  
  
         if x < 20:  
             raise Exception("kiritilgan qiymat 20 dan katta bo'lishi shart")
```

```
In [ ]:
```

# Python Try Except istinolari

- Masalan, agar `x` butun son bo'lmasa, **TypeError Exception (istisnoni)**ni ko'tarish quyidagidek yozilishi mumkin:

```
x = "salom"

if not type(x) is int:
    raise TypeError("Faqat raqam kiritsh mumkin")
```

```
-----
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_11240\2651724584.py in <module>
      2
      3 if not type(x) is int:
----> 4     raise TypeError("Faqat raqam kiritsh mumkin")

TypeError: Faqat raqam kiritsh mumkin
```

# Python Try Except istinolari

## Foydalanuvchi ma'lumot kiritishi

- Python dasturlash tilida foydalanuvchi klaviaturadan ma'lumot kiritishi mumkin.
- Bu shuni anglatadiki, biz foydalanuvchidan ma'lumot so'rashimiz mumkin. Bu usul Python 3.6 da Python 2.7 dan biroz farq qiladi. Python 3.6 **input()** funksiyasidan foydalaniladi. Python 2.7 **raw\_input()** funksiyasidan foydalaniladi.

# Python Try Except istinolari

- Quyidagi misol foydalanuvchidan qaysi dasturlash tilini bilishini soʻraydi va kiritilgan maʼlumotni ekranda chop etadi:

```
dasturlashtili = input("Qaysi Dasturlash Tilini Bilasiz? ")  
print("Siz "+dasturlashtili+" dsturlash tilini bilasiz")
```

```
Qaysi Dasturlash Tilini Bilasiz? Java  
Siz Java dsturlash tilini bilasiz
```

# Python Try Except istinolari

- Quyidagi misol foydalanuvchidan oxirgi o`qigan kitobini nomini kiritishni so`raydi va kiritilgan ma`lumotni ekranda chop etadi:

```
kitobnomi = input("oxirgi o`qigan kitobingiz nomi? ")  
print("Siz o`qigan kitob"+kitobnomi)
```

```
oxirgi o`qigan kitobingiz nomi? Python asoslari  
Siz o`qigan kitobPython asoslari
```

# Python Try Except istinolari

- Quyidagi misol foydalanuvchidan ismini kiritishni soʻraydi va kiritilgan maʼlumotni ekranda chop etadi:
- `X = input("Ismingizni kiriting")`
- `Print(X)`

# Python Try Except istinolari

- Quyidagi misol foydalanuvchidan tug`ilgan yilini kiritishni so`raydigan va foydalanuvchini yoshini hisoblab ko`rsatadigan kod keltirilgan:

- `X= input("Tug`ilgan yilingizni kiriting")`
- `A = int(X)`
- `B = 2023-A`
  
- `Print(B)`

# Foydalanilgan adabiyotlar

1. Mastering Object-Oriented Python: Build powerful applications with reusable code using OOP design patterns and Python 3.7, 2nd Edition, Steven F. Lott, Packt Publishing (June 14, 2019)
2. Learning Python, 5th Edition Fifth Edition, Mark Lutz , O'Reilly Media, June 12, 2013
3. Python Programming for Beginners: The Ultimate Guide for Beginners to Learn Python Programming: Crash Course on Python Programming for Beginners, AMZ Publishing, Independently published (July 13, 2021)
4. <https://www.python.org/>
5. <https://www.w3schools.com/>
6. <https://www.codecademy.com/catalog/language/python>
7. <https://realpython.com/>
8. <https://www.anaconda.com/>

**E'tiboringiz  
uchun rahmat!**