Hibakezelés

2025

1 Hibakezelés

Pythonban a `try` blokk lehetővé teszi egy kódblokk tesztelését hibák keresésére. Ha a blokkon belül keletkezik egy hiba, akkor kapunk egy kivételt (except), amely a hiba típusára utal.

Gyakori kivételtípusok: - TypeError: Érték- vagy típusellenőrzési hiba. - ValueError: Érvénytelen érték vagy argumentum. - ZeroDivisionError: Nullával való osztás hiba. - FileNotFoundError: Fájl nem található hiba. - IndexError: Érvénytelen indexelés hiba egy sorozatban (pl. lista, tuple). - KeyError: Érvénytelen kulcs hiba egy szótárban. - Exception: Az összes kivétel ősosztálya. Elkap bármely típusú kivételt.

A hibát az `except` kulcshoz tartozó kivételtípusok alapján el lehet kapni és le lehet kezelni. A kivételkezelés sorrendje számít. Ha több `except` blokk van jelen, akkor a Python az első olyan blokkot fogja választani, amelyik a kiváltott kivétel típusával megegyezik. Az általános célú `Exception` minden hibát elkap, ezért ezt kell a legutoljára hagyni. Az üresen hagyott `except` kulcsszó az általános `Exception` típus rövidített alakja.

Lehetőség van itt is egy olyan `else` ág használatára, amely csak akkor fut le, ha nem keletkezett előtte hiba.
Egy `finally` blokk a végén mindig lefuttatható, függetlenül attól, hogy történt-e előtte hiba.

[ ]:
```
try:
    oszto = int(input("Adja meg az osztót: "))
    eredmeny = 10 / oszto
    print("Az eredmény:", eredmeny)

except ZeroDivisionError:
    print("Hiba: Nullával való osztás nem megengedett!")

except ValueError:
    print("Hiba: Érvénytelen bemenet! Egész számot adjon meg.")

except Exception as error:
    print("Hiba: ", error)

else:
    print("A try blokk sikeresen lefutott.")

finally:
```

```
    print("A finally blokk mindig lefut, függetlenül attól, volt-e kivétel vagy␣
 ↪sem.")
```

1.1 Kivétel dobása

Saját ellenőrzések során talált hibák esetén, mi magunk is dobhatunk egy kivételt a `raise` kulcsszó segítségével. A kulcsszó után adjuk meg a kivétel típusát és a hibüzenetet.

Típushiba jelzése:

[ ]:
```
x = "hello"

if not type(x) is int:
    raise TypeError("Hiba: Érvénytelen bemenet! Egész számot adjon meg.")
```

Tartomány hiba jelzése:

[ ]:
```
x = -1

if x < 0:
    raise ValueError("Hiba: Az érték kisebb, mint nulla.")
```

1.2 Kivétel tovább dobása

Egy függvényben elkapott kivétel tovább adható egy üres `raise` kulcsszóval. A hívó rész újra elkaphatja és lekezelheti a tovább küldött hibát.

[ ]:
```
def osztas(a, b):
    try:
        eredmeny = a / b
        return eredmeny

    except ZeroDivisionError as error:
        print(f"Hiba: {error}")
        raise  # Az eredeti kivétel tovább dobása


# Kivétel kezelése az osztas() függvény hívásakor
try:
    result = osztas(10, 0)
    print("Az eredmény:", result)

except ZeroDivisionError:
    print("A kivétel el lett kapva a főprogramban is.")
```