

Kivételkezelés: try-except

Miért van rá szükség?

1. Elfogy a memória.
2. Hiányzik a kettőspont a ciklusfej végéről:

```
for szam in [1,2,3]  
    print(szam)
```

Vagy a ciklusmag után hiányzik az üres sor.
3. Behúzás-hiba
4. return utasítás függvényen kívül van
5. Lista utolsó utáni elemét íratnám ki.

Szintaktikai és szemantikai hiba.

```
szam = int(input('Adj meg egy számot!'))  
print(f'A megadott szám négyzete {szam**2}')  
Ide adjunk meg egy betűt!
```

```
try:  
    szam = int(input('Adj meg egy számot!'))  
    print(f'A megadott szám négyzete {szam**2}')  
except:  
    print('Nem számot adtál meg!')  
print('A program vége!')
```

```
except Exception as e:  
    print(e)  
    print('Nem számot adtál meg!')
```

Exception helyett ValueError vagy ZeroDivisionError
Kivételobjektum keletkezik, amit el kell kapni.

```
except ValueError as e:  
    -----  
except ZeroDivisionError as e:  
    -----
```

```
oszto = int(input('Melyik számmal osszam el a 10-et?'))  
print(f'Az eredmény: {10/oszto}')  
A két except ág tesztelése.
```

```
except(ValueError, ZeroDivisionError) as e:  
    print(e)  
    print(type(e))
```

Kivételkezelés: else-finally

```
munkanapok = ['h', 'k', 'sz', 'cs', 'p']
hanyadik = int(input('Hányadik?'))
munkanap = munkanapok[hanyadik-1]
print(f'A {hanyadik}. munkanap a(z) {munkanap}.')
```

7 esetén IndexError, W esetén ValueError

```
try:
    hanyadik = int(input('Hányadik?'))
    munkanap = munkanapok[hanyadik-1]
except ValueError:
    print('Nem szám.')
```

except IndexError:

```
    print('1-5 között legyen!')
```

else: # helyes esetén

```
    print(f'A {hanyadik}. munkanap a(z) {munkanap}.')
```

finally:

```
    print('Köszönöm a választ!')
```

Új

```
def hanyados(osztando, oszto):
    try:
        return osztando/oszto
    except ZeroDivisionError:
        print('Nullával nem lehet osztani!')
```

finally:

```
    print('A függvény végrehajtva.') # return után áll, de végrehajtodik
```



```
print(hanyados(3,2))
print(hanyados(3,0))
# de kijött a return none
eredmeny = hanyados(3,0)
if eredmeny: # azaz ha az eredmény nem none!
    print(f'Az eredmény: {eredmeny}')
```

Saját kivételosztály definiálása

```
rgb = []
srsz = 1
while len(rgb) != 3:
    try:
        rgb_osszetevo = int(input(f'Add meg a(z) {srsz}. RGB-összetevőt!'))
    except ValueError as e:
        print(e)
        print('Adj meg új értéket!')
```

```
else:
    rgb.append(rgb_osszetevo)
    srsz += 1
```

```
print(rgb)
```

```
# futtatás: 255, 333, w – csak utóbbi lesz a hiba
```

```
# Az egész elé:
```

```
def rgb_osszetevo_ellen(ertek):
    if 0 <= ertek <= 255:
        return ertek
    else:
        ValueError(f'{ertek}: érvénytelen RGB-összetevő!')
```

```
# Saját kivételosztály:
```

```
class RGBError(Exception): # az Exception osztály leszármazottja
    """
    Nem megfelelő RGB-összetevő esetén
    """
    except RGBError as e:
        print(e)
        print('A számnak a [0;255] tartományba kell esnie! Adj meg új értéket!')
```

```
# Az egész elé javítva:
```

```
def rgb_osszetevo_ellen(ertek):
    if 0 <= ertek <= 255:
        return ertek
    else:
        raise RGBError(f'{ertek}: érvénytelen RGB-összetevő!')
    # a raise kivételobjektumot hoz létre
```