

ניהול שגיאות וחריגות

try, except, raise, finally

למה צריך לנהל שגיאות?

- שליטה בזרימת ההרצה:
- אנחנו לא רוצים שההרצה תעצור למרות שהאינטרפרטר מחייב לעצור
- אנחנו רוצים לעצור את ההרצה למרות שהאינטרפרטר לא מחייב
- הבנה של מה שקורה, סוגי שגיאות
- אנחנו יכולים לטפל בשגיאות בתוך הקוד שלנו

שגיאות סינטקס וחריגות

```
x = is
```

```
SyntaxError: invalid syntax
```

```
x,y = 0,1
```

```
y/x
```

```
Traceback (most recent call last):
```

```
File "<stdin>", line 1, in <module>
```

```
ZeroDivisionError: division by zero
```

<https://docs.python.org/3/library/exceptions.html#builtin-exceptions>

try except

```
[print(3/i) for i in range(5,-5,-1)]
```

```
for i in range(5,-5,-1):  
    try:  
        print(3/i)  
    except:  
        pass
```

```
for i in range(5,-5,-1):  
    try:  
        print(3/i)  
    except:  
        print("x --> inf")
```

עם סוג השגיאה try expect

```
while True:
```

```
    try:
```

```
        x = int(input("Please enter a number: "))
```

```
        break
```

```
    except ValueError:
```

```
        print("Oops! That was no valid number. Try again...")
```

• אפשר להעמיס כמה סוגי שגיאות

```
except (RuntimeError, TypeError, NameError):
```

```
    pass
```

try except דוגמה לזרימת ניהול שגיאות

```
import sys
try:
    f = open('myfile.txt')
    s = f.readline()
    i = int(s.strip())
except OSError as err:
    print(f"OS error: {err}")
except ValueError:
    print("Could not convert data to an integer.")
except:
    print("Unexpected error:", sys.exc_info()[0])
    raise
```

raise רוצים ליצור שגיאה

```
try:  
    a = int(input("Enter a positive integer: "))  
    if a <= 0:  
        raise ValueError("That is not a positive number!")  
except ValueError as val_err:  
    print(val_err)
```

יצירת סוג שגיאה חדש

- יצרת מודול ואתה רוצה שיהיו לו שגיאות משלו. למשל?

- אתה יכול לרשת את Exception

```
class MyError(Exception):  
    pass
```


יצירת שגיאה assert

assert condition, error

בכל מקרה finally

```
try:
    raise KeyboardInterrupt
finally:
    print('Goodbye, world!')
```

- שאלה הנשאלת: בשביל מה זה טוב? הרי יש לנו כבר except ואפשר לכתוב אחריו מה שרוצים
דוגמה

```
def fin(x):
    result = "a"
    try:
        result += x
    except:
        return "error"
    finally:
        print("the best result we have is " + result)
    return "successful: x is str"
```