Sets

```
fruits_set = {"apple", "banana", "cherry"}
print(fruits_set) # {'banana', 'apple', 'cherry'}
print(len(fruits_set)) # 3
print(type(fruits_set)) # <class 'set'>

numbers_list = [1, 3, 2, 1, 57, 5, 9, 7, 3, 2]
print(set(numbers_list))  # {1, 2, 3, 5, 7, 9, 57}
print("banana" in fruits_set) #  בודקת אם מילת BANANA ב fruits_setקיימת, ומחזירה TRUE/False
fruits_set.add("orange") # {'orange', 'banana', 'apple', 'cherry'}

tropical_fruits_set = {"pineapple", "mango", "papaya"}
fruits_set.update(tropical_fruits_set) # ממזגת שתי רשימות

tropical_fruits_set = {"pineapple", "mango", "papaya", "mango"}
print(tropical_fruits_set) # מוחקת הכפיליות ברשימה

tropical_fruits_set.remove("mango") # מוחקת כל מילות ה mangoמהרשימה
fruits_set.discard("orange") # remove "orange" from set


numbers_set = {1, 422, 22, 77, 9}
print(sorted(numbers_set)) # מיון [1, 9, 22, 77, 422]
print(sorted(numbers_set, reverse=True)) # מיון בסדר הפוך [422, 77, 22, 9, 1]

# Set functions
# print({1, 2, 3}.intersection({2, 3, 4, 5}))  # {2, 3}
# print({1, 2, 3}.union({2, 3, 4, 5}))  # {1, 2, 3, 4, 5}
# print({1, 2, 3}.symmetric difference({2, 3, 4, 5}))  # {1, 4, 5}
```

Array

```
grades = [67, 78, 98, 23, 45, 56, 45]

grades.append(99)  # [67, 78, 98, 23, 45, 56, 45, 99]
print(grades.index(45))  # return index a first number 45 in array
print(grades.count(45))  # counter how many number 45 in array grades
grades.remove(45)  # Delete the first number 45 in array
removed_grade = grades.pop(1) # 78 - index (1)
print(grades.pop())  # remove a last number

fruits = ['apple', 'banana', 'cherry']
fruits.insert(1, "orange") # insert orange in index 1
fruits.reverse() # reverse a array

cars = ['Ford', 'BMW', 'Volvo']
cars.sort() # sorted a array
cars.sort(reverse=True) # sorted a array reverse

fruits = ['apple', 'banana', 'cherry']
cars = ['Ford', 'BMW', 'Volvo']
fruits.extend(cars) # מיזוג בין שתי מערכים - ['apple', 'banana', 'cherry', 'Ford', 'BMW', 'Volvo']
```

String

```
#String functions
txt = "I love apples, apple are my favorite fruit"
x = txt.count("apple")
print(x) => 2

#Checking if the given string starts/ends with
txt = "Hello, welcome to my world."
x = txt.endswith(".")
print(x) => True


txt = "Hello, welcome to my world."
x = txt.startswith("Hello")
print(x) => True


#Checking if the given string contains specific characters only
txt = "Company12"
x = txt.isalnum()
print(x) => True


txt = "CompanyX"
x = txt.isalpha()
print(x) => True


txt = "50800"
x = txt.isdigit()
print(x) => True
```

```
txt = "hello world!"
x = txt.islower()
print(x) => True


txt = "THIS IS NOW!"
x = txt.isupper()
print(x) => True


# Join and split the given string by the specific separator
my_tuple = ("John", "Peter", "Vicky")
x = "#".join(my_tuple)
print(x) => John#Peter#Vicky


txt = "welcome to the jungle"
x = txt.split()
print(x) => ['welcome', 'to', 'the', 'jungle']


# Lower case letters become upper case letters and vice versa
txt = "Hello my FRIENDS"
x = txt.lower()
print(x) => hello my friends


txt = "Hello my friends"
x = txt.upper()
print(x) => HELLO MY FRIENDS
```

חריגות

1(

```
try:
    age = int(input("Enter your age"))
    print("Your age is:", age)
except ValueError:
    print("Your age must contain digits only")

try:
    grade = int(input("Enter your grade:"))

    if grade < 0 or grade > 100:
        raise ValueError

    print(f"Your grade is:{grade}")
except ValueError:
    print("Wrong grade input. Must be a positive number between 0 and 100 only")
```

)

2(

```
try:
    my_list = ["abc", "def", "dfg"]
    index = int(input("Enter your index"))
    print(my_list[index])
except ValueError:
    print("Your index must be numeric only")
except IndexError:
    print("Your index is out of range")
```

)

3(

```
try:
    divide_by = int(input("Enter an integer number to divide (not zero)):"))
    print(120 / divide_by)
except ValueError:
    print("You must enter a numeric value (not zero)")
except ZeroDivisionError:
    print("You can't divide by zero")
```

)

4(

```
students_info_dictionary = {"327583828": "Kobi Levi", "358796939": "Avi Cohen"}

# Method 1
try:
    student_id = input("Enter student's id number:")

    print(f"Student with id {student_id} is: {students_info_dictionary[student_id]}")
except KeyError:
    print("Student id doesn't exist")

# Method 2
student_id = input("Enter student's id number:")

print(
    f"Student with id {student_id} is: {students_info_dictionary[student_id]}"
    if student_id in students_info_dictionary
    else "Student id doesn't exist")
```

<Function>

```
def get_user_age() -> int:
    """
    Function gets user's age
    :raises ValueError This exception is thrown when the user's age is not numeric
    :return: Getting user's age
    """
    try:
        return int(input("Enter your age:"))
    except ValueError:
        raise ValueError("Your age can contain digits")


try:
    print("Your age is:", get_user_age())
except ValueError as v:
    print(v)
```

Txt

"r" - קריאה - ערך ברירת מחדל. פותח קובץ לקריאה, שגיאה אם הקובץ לא קיים

"א" - הוסף - פותח קובץ להוספה, יוצר את הקובץ אם אינו קיים

"w" - כתיבה - פותח קובץ לכתיבה, יוצר את הקובץ אם אינו קיים

"x" - צור - יוצר את הקובץ שצוין, מחזיר שגיאה אם הקובץ קיים

f = open("demofile.txt") – **open file**

f = open("demofile.txt" , "w") – **open file**

print(f.read()) – **לקריאת הקובץ**

print(f.read(**5**)) – **read line 5**

f = open("demofile2.txt", "a") – **open file**
f.write("Now the file has more content!")

f.close() – **close the file**

<u>#open and read the file after the appending:</u>

f = open("demofile2.txt", "r")

print(f.read()) => **Now the file has more content!**

#DELETE

```
import os
os.remove("demofile.txt") – remove file
```

בדיקה אם הקובץ קיים

```
import os
if os.path.exists("demofile.txt"):
  os.remove("demofile.txt")
else:
  print("The file does not exist")
```

```
from os.path import exists, isfile
print(exists("test.txt") and isfile("test.txt"))

try:
    with open('test.txt', 'r') as f:  # a w
        print(f.read())
        for line in f:
            print(line.rstrip())

        f.seek(0)
        print(f.readlines())

        print([line.rstrip() for line in f.readlines()])

        for line in reversed(f.readlines()):
            print(line.rstrip())

        for line in f.readlines()[1:3]:
            print(line.rstrip())

except FileNotFoundError:
    print("The specified file doesn't exist")
except PermissionError:
    print("You don't have permission to read from the file")

# WRITING TO A FILE
try:
    with open('test.txt', 'w') as f:  # a
        f.write("My amazing text")
except PermissionError:
    print("You don't have permission to read from the file")
```