

# PODSTAWY PROGRAMOWANIA W PYTHON

Dzień 7



# AGENDA

## DAY 4

- praca z plikami
- pliki csv (wartości oddzielana przecinkami)
- with
- tuple (krotka)
- dict (słownik)

# 1. Praca z plikami

# pliki tekstowe

otwieramy plik

```
plik = open("ścieżka_do_pliku", tryb)
```

tryby:

**r** - tylko do odczytu

**w** - zapisywanie pliku (stary plik o tej samej nazwie będzie usunięty)

**r+** - do odczytu i zapisu

**a** - dopisywanie do pliku (dane są dopisane do końca istniejącego pliku)

# pliki tekstowe

**plik.read()** – odczytanie całego pliku, zwracany jest string zawierający cały tekst pliku (włącznie ze znakami \n) – opc. argument – **int** określająca ilość bajtów do wczytania

**plik.readline()** – odczytanie jednej linii z pliku, zwracany jest string z liniijką testu, włącznie ze znakiem \n

**plik.readlines()** – odczytuje cały tekst – zwraca listę stringów - linijek

```
for line in plik:  
    print(line, end='')
```

## with

pliki należy zamykać po użyciu:

```
plik = open(„plik.txt”)  
    # kod  
plik.close()
```

otwarcie pliku za pomocą with pozwala na automatyczne zamykanie pliku przez Pythona

```
with open(„plik.txt”) as plik:  
    print(plik.readline())
```

# pliki tekstowe

`plik.write(string)` – zapisuje string do pliku w obecnej pozycji kursora, zwraca liczbę zapisanych znaków – należy pamiętać o znaku `\n`

`plik.writelines(iterable)` – zapisuje elementy z kolekcji jako poszczególne linie w pliku

Plik musi być otworzony w trybie do zapisu aby móc go zmieniać!

# CSV

Pliki CSV – comma separated values – dane oddzielane przecinkami

Imie,Nazwisko,Adres,Telefon

Joanna,Kowalska,Gdansk Przytulna,64 654-65-45

Adam,Nowak,Gdynia Swietojanska,0700325487

Do obsługi plików CSV można użyć biblioteki csv



# Inne struktury danych

tuple, dict

# **tuple – krotka**

## **lista niemutowalna**

```
imiona = ('Ola', 'Ala', 'Ewa')
```

```
nazwiska = 'Kowalska', 'Nowak', 'Tomczyk',
```

Tuple jest typem nie zmiennym, więc nie można dodawać, usuwać i zmieniać elementów. Można za to jako element użyć listy, wtedy elementy wew. listy można zmieniać

# dict - słownik

```
osoba = {'imie': 'Anna', 'nazwisko': 'Kowalska'}
```

```
osoba['imie']
```

Klucze mogą być tylko typu niemutowalnego

Klucze muszą być unikalne



# Thanks!!