

# Eksploracja danych internetowych

## Laboratorium 1

Prowadzący: pracownik UR

Wykonał: Piotr Rojek, pr125159

### Zadanie 1

Zainstaluj Python oraz Pandas w ostatniej stabilnej wersji.

```
Wiersz polecenia x + v - □ x
C:\Users\Piotr>python --version
Python 3.13.2

C:\Users\Piotr>python -c "import pandas as pd; print(pd.__version__)"
2.2.3
```

### Zadanie 2

Uruchom interaktywny interpreter poleceń.

```
Wiersz polecenia - python x + v - □ x
Microsoft Windows [Version 10.0.26100.2894]
(c) Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\Piotr>python
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb 4 2025, 15:23:48) [MSC v.1942 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> |
```

### Zadanie 3

Zdefiniuj listę składającą się z 10 przykładowych elementów tego samego typu (np.: string).

```
>>> list1 = ["Ala", "ma", "kota", "a", "kot", "ma", "Ale", "i", "są", "szczęśliwi"]
```

### Zadanie 4

Zdefiniuj listę składającą się z 10 elementów różnych typów.

```
>>> list2 = [1, 3.14, 3 + 7j, "dom", True, [2, "lista"], ("Jakaś", "krotka"), {"Jan\ ": 13, "Adam": 15}, {"jabłko", "gruszka", "śliwka"}, b"kot"]
```

### Zadanie 5

Spróbuj w pętli dokonać konwersji elementów z powyższej listy do typu całkowitego, a wyniki zapisz w oddzielnej liście (użyj funkcji append do dołączania wyników do nowej listy).

```
>>> list3 = []
>>> for element in list2:
...     try:
...         if type(element) is complex:
...             element = element.real
...             list3.append(int(element))
...     except:
...         pass
...
>>> print(list3)
[1, 3, 3, 1]
```

### Zadanie 6

Napisz program, który wczytuje ze standardowego wejścia ciąg znaków i sprawdza czy jest on palindromem.

```
text = input("Podaj ciąg znaków do sprawdzenia palindroma: ")
text = text.lower()
size = int(len(text) / 2)
i = 0
palindrom = True
while i < size:
    if text[i] != text[-i - 1]:
        palindrom = False
        break
    i += 1
print("Czy słowo '" + text + "' to palindrom:", palindrom)
```

Testy:

```
Podaj ciąg znaków do sprawdzenia palindroma: ala
Czy słowo 'ala' to palindrom: True
```

```
Podaj ciąg znaków do sprawdzenia palindroma: alla
Czy słowo 'alla' to palindrom: True
```

```
Podaj ciąg znaków do sprawdzenia palindroma: kotek
Czy słowo 'kotek' to palindrom: False
```

```
Podaj ciąg znaków do sprawdzenia palindroma: Kajak
Czy słowo 'kajak' to palindrom: True
```

## Zadanie 7

Napisz funkcję o nazwie `is_prime`, która przyjmuje jeden parametr (liczbę naturalną) i sprawdza, czy jest ona liczbą pierwszą.

```
def is_prime(number: int): 11 usages
    if number < 2:
        print("Liczba '" + str(number) + "' nie jest liczbą pierwszą")
        return
    if number == 2:
        print("Liczba '2' jest liczbą pierwszą")
        return
    if (number % 2) == 0:
        print("Liczba '" + str(number) + "' nie jest liczbą pierwszą")
        return
    czy_pierwsza = True
    dzielnik = 3
    while dzielnik <= (number / 2) + 1:
        if number % dzielnik == 0:
            czy_pierwsza = False
            break
        dzielnik += 2
    if czy_pierwsza:
        print(f"Liczba '{str(number)}' jest liczbą pierwszą")
    else:
        print(f"Liczba '{str(number)}' nie jest liczbą pierwszą")

is_prime(0)
is_prime(1)
is_prime(2)
is_prime(3)
is_prime(4)
is_prime(5)
is_prime(9)
is_prime(10)
is_prime(11)
is_prime(12)
is_prime(13)
```

Testy:

```
Liczba '0' nie jest liczbą pierwszą
Liczba '1' nie jest liczbą pierwszą
Liczba '2' jest liczbą pierwszą
Liczba '3' jest liczbą pierwszą
Liczba '4' nie jest liczbą pierwszą
Liczba '5' jest liczbą pierwszą
Liczba '9' nie jest liczbą pierwszą
Liczba '10' nie jest liczbą pierwszą
Liczba '11' jest liczbą pierwszą
Liczba '12' nie jest liczbą pierwszą
Liczba '13' jest liczbą pierwszą
```

## Zadanie 8

Stwórz przykładowy obiekt klasy DataFrame i zaprezentuj podstawowe operacje (wybór kolumny, wierszy, przeglądanie, itp.).

```
import pandas as pd

data = {
    "substancja": ["tlen", "magnez", "żelazo", "chlor", "wodór", "promet"],
    "gęstość (kg/m³)": [1.429, 1738, 7874, 3.214, 0.082, 7264],
    "topnienie (°C)": [-218.79, 650, 1538, -101.5, -259.198, 1042],
    "wrzenie (°C)": [-182.962, 1090, 2861, -34.04, -252.762, 3000]
}

df = pd.DataFrame(data)

print("Wypisanie całej ramki:")
print(df)

print("\nWypisanie drugiej kolumny")
print(df["gęstość (kg/m³)"])

print("\nSprawdzenie drugiej kolumny, czy wartości są większe od 10")
print(df["gęstość (kg/m³)"] > 10)

print("\nWypisanie trzeciego wiersza:")
print(df.loc[2])

print("\nWypisanie pierwszych trzech wierszy:")
print(df.head(3))

print("\nWypisanie ostatnich dwóch wierszy:")
print(df.tail(2))
```

Testy:

```
Wypisanie całej ramki:
  substancja  gęstość (kg/m³)  topnienie (°C)  wrzenie (°C)
0        tlen           1.429        -218.790        -182.962
1      magnez          1738.000          650.000         1090.000
2      żelazo          7874.000         1538.000         2861.000
3       chlor           3.214         -101.500          -34.040
4      wodór           0.082         -259.198         -252.762
5      promet          7264.000         1042.000         3000.000

Wypisanie drugiej kolumny
0           1.429
1          1738.000
2          7874.000
3           3.214
4           0.082
5          7264.000
Name: gęstość (kg/m³), dtype: float64
```

Sprawdzenie drugiej kolumny, czy wartości są większe od 10

0 False

1 True

2 True

3 False

4 False

5 True

Name: gęstość (kg/m<sup>3</sup>), dtype: bool

Wypisanie trzeciego wiersza:

substancja żelazo

gęstość (kg/m<sup>3</sup>) 7874.0

topnienie (°C) 1538.0

wrzenie (°C) 2861.0

Name: 2, dtype: object

Wypisanie pierwszych trzech wierszy:

	substancja	gęstość (kg/m <sup>3</sup> )	topnienie (°C)	wrzenie (°C)
0	tlen	1.429	-218.79	-182.962
1	magnez	1738.000	650.00	1090.000
2	żelazo	7874.000	1538.00	2861.000

Wypisanie ostatnich dwóch wierszy:

	substancja	gęstość (kg/m <sup>3</sup> )	topnienie (°C)	wrzenie (°C)
4	wodór	0.082	-259.198	-252.762
5	promet	7264.000	1042.000	3000.000