

# Przetwarzanie plików

Podstawy programowania w języku Python



# Ścieżka dostępu do pliku

- określa lokalizację pliku w systemie plików
- względna - określają lokalizację pliku w odniesieniu do bieżącego katalogu
- absolutna - określają pełną lokalizację pliku od katalogu głównego (/)
- system Windows używa innej konwencji nazewnictwa niż ta stosowana w systemach Unix/Linux

`./folder/plik.txt`

`..\plik.txt`

`C:\folder\plik.txt`

# Ścieżki dostępu w Pythonie

Ścieżkę dostępu do pliku znajdującego się w systemie Windows można w Pythonie zapisać:

- stosując escape'owanie

```
"C:\\folder\\plik.txt"
```

- stosując znak ukośnika (slash'a)

```
"C:/folder/plik.txt"
```

# Strumienie i uchwyt plików

- **strumień** - abstrakcja, która reprezentuje sekwencję danych, z których można odczytywać lub do których można zapisywać, może być powiązany z plikiem fizycznym na dysku twardym, ale także z innymi źródłami danych, takimi jak pamięć podręczna lub urządzenia sieciowe
- **uchwyt pliku** - konkretny obiekt reprezentujący otwarty plik w systemie operacyjnym, zawiera on informacje o pliku, takie jak jego nazwa, lokalizacja, rozmiar i uprawnienia dostępu

# Tryby otwarcia strumienia

- tryb odczytu - pozwala tylko na operacje odczytu, próba zapisu w strumieniu spowoduje wyjątek
- tryb zapisu - umożliwia tylko operacje zapisu, próba odczytania strumienia spowoduje wyjątek
- tryb aktualizacji - pozwala na zarówno zapis, jak i odczyt

# Tryby otwarcia strumienia

- **r** - otwiera plik do odczytu (domyślny)
- **w** - otwiera plik do zapisu (nadpisuje istniejącą zawartość)
- **a** - otwiera plik do dopisywania (dodaje nową zawartość na końcu)
- **r+** - otwiera plik do odczytu i zapisu
- **w+** - otwiera plik do odczytu i zapisu (nadpisuje istniejącą zawartość)
- **a+** - otwiera plik do odczytu i dopisywania

# Rodzaje strumieni

- **wejściowe** - do odczytywania danych, np. **sys.stdin** (standardowe wejście) i pliki otwarte w trybie **r**
- **wyjściowe** - służą do zapisu danych, np. **sys.stdout** (standardowe wyjście) i pliki otwarte w trybie **w** lub **a**
- **binarne** - służą do odczytywania i zapisywania danych binarnych, np. pliki otwarte w trybie **rb** lub **wb**
- **tekstowe** - służą do odczytywania i zapisywania danych tekstowych, np. pliki otwarte w trybie **r** lub **w** bez flagi **b**

# Metody manipulujące na danych

- **read()** - odczytuje cały plik i zwraca jego zawartość jako łańcuch znaków
- **readline()** - odczytuje jedną linię z pliku i zwraca ją jako łańcuch znaków
- **readlines()** - odczytuje wszystkie linie z pliku i zwraca je jako listę łańcuchów znaków
- **write(dane)** - zapisuje dane w pliku, dane mogą być łańcuchem znaków, listą, słownikiem lub innym typem danych



# Poprawna obsługa strumieni

Aby poprawnie i bezpiecznie korzystać ze strumieni należy zawsze po ich użyciu zapewnić ich zamknięcie np. jednym ze sposobów:

- automatycznie – korzystając z konstrukcji **with**

```
1  with open("file.txt", "r") as file:  
2      # odczyt danych z pliku
```

- ręcznie – wywołując na uchwycie pliku metodę **close()**

```
1  file = open("file.txt", "r")  
2  # odczyt danych z pliku  
3  file.close()
```

# Obsługa błędów

- stosowanie konstrukcji **try-catch**
- przykłady możliwych błędów:
  - **FileNotFoundError** - plik nie istnieje
  - **PermissionError** - brak uprawnień do odczytu lub zapisu pliku
  - **IsADirectoryError** - podana ścieżka jest katalogiem, a nie plikiem
  - **IOError** - ogólny błąd wejścia/wyjścia
- stosowanie funkcji **strerror()** z modułu **os** pozwalającej „przetłumaczyć” kod błędu

```
1  try:
2      with open("plik.txt", "r") as f:
3          # Kod operujący na pliku
4  except FileNotFoundError:
5      print("Plik nie istnieje")
6  except PermissionError:
7      print("Brak uprawnień")
8  except Exception as e:
9      print(f"Wystąpił błąd: {e}")
```

# Pytanie

Która funkcja otwiera strumień tekstowy w trybie odczytu?

- a) `open(filename, "w")`
- b) `open(filename, "r")`
- c) `open(filename, "a")`
- d) `open(filename, "x")`

**Odpowiedź: b)**

# Pytanie

Zakładając, że plik **file.dat** nie istnieje, wskaż co wyświetli się na ekranie po uruchomieniu skryptu:

- a) Wystąpił nieznany błąd
- b) OK
- c) Nie mogę otworzyć pliku
- d) wystąpi wyjątek FileNotFoundError

```
1  try:
2      open("file.dat", "wb")
3      print("OK")
4  except IOError:
5      print("Nie mogę otworzyć pliku")
6  except:
7      print("Wystąpił nieznany błąd")
```

## Odpowiedź: b)

W przypadku próby otwarcia strumienia binarnego w trybie zapisu (wb), nieistniejący plik zostanie utworzony natomiast istniejący zostanie wyzerowany.

# Pytanie

Zakładając, że plik **file.txt** zawiera kilka wierszy tekstu, wskaż co wyświetli się na ekranie po wykonaniu poniższego kodu:

```
1  for value in open("file.txt"):
2      print(value)
```

- a) pierwsza linia tekstu z pliku
- b) zawartość pliku
- c) wszystkie wiersze z pliku rozdzielone pustymi wierszami
- d) wystąpi **SyntaxError**

**Odpowiedź:** c)

Pętla będzie wyświetlała kolejne wiersze, które w pliku zakończone są znakami nowej linii. Po wyświetleniu każdej wiersza z pliku, znak nowej linii będzie także generowała funkcja `print()`.