

Operacje wejścia/wyjścia

Pliki

Pliki - Otwarcie/Zamknięcie pliku

Otwarcie pliku realizowane jest za pomocą funkcji `open()` przyjmującej dwa argumenty - nazwę pliku i tryb otwarcia, funkcja ta zwraca obiekt, który pozwala na wykonanie podstawowych operacji na pliku.

Zamknięcie pliku

jest realizowane poprzez wywołanie funkcji `close()` na zwróconym obiekcie.

```
fp = open("tmp.txt", "w")

# pozostałe instrukcje programu

# wymagane zamknięcie pliku
fp.close()
```



Operacje wejścia/wyjścia

Pliki - Otwarcie/Zamknięcie pliku

Tryb otwarcia pliku określa sposób, w jaki plik będzie używany. Możliwymi trybami są:

- **r** - plik będzie tylko odczytywany,
- **r+** - dane z/do pliku będą odczytywane i zapisywane,
- **w** - dane do pliku będą zapisywane (jeśli plik istnieje jego zawartość zostanie usunięta),
- **a** - dane do pliku będą dopisywane (jeśli plik istnieje jego zawartość nie zostanie usunięta).

W przypadku dopisania do trybu litery **b**, plik zostanie otwarty w trybie binarnym.



Operacje wejścia/wyjścia

Pliki - Otwarcie/Zamknięcie pliku

Jakie problemy istnieją
w tym rozwiązaniu?

```
fp = open("tmp.txt", "w")  
  
# pozostałe instrukcje programu  
  
# wymagane zamknięcie pliku  
fp.close()
```



Operacje wejścia/wyjścia

Pliki - Otwarcie/Zamknięcie pliku

Jakie problemy istnieją
w tym rozwiązaniu?

Problem:

Co się stanie jeśli zapomnimy
wywołać funkcji `close()`?

```
fp = open("tmp.txt", "w")  
  
# pozostałe instrukcje programu  
  
# wymagane zamknięcie pliku  
fp.close()
```



Operacje wejścia/wyjścia

Pliki - Konstrukcja with

Specjalna konstrukcja **with** automatycznie zamyka plik po zakończeniu wykonywania bloku kodu.

```
with open("tmp.txt", "w") as fp:  
    # kod operujący na pliku  
    pass
```

```
# w tym miejscu plik jest zamknięty
```



Operacje wejścia/wyjścia

Pliki - Odczyt/Zapis danych

Wczytanie całego pliku

do zmiennej można uzyskać poprzez wywołanie funkcji **read()** na obiekcie reprezentującym plik.

Nazwa text będzie wskazywała na napis w postaci ciągu znaków (string).

```
with open("tmp.txt") as fp:  
    text = fp.read()  
    print("Zawartość pliku:", text)
```



Operacje wejścia/wyjścia

Pliki - Odczyt/Zapis danych

Odczytanie pojedynczej linii

z pliku możliwe
jest za pomocą funkcji
readline().

W przypadku napotkania
na koniec pliku funkcja
zwraca pusty napis.

```
# odczytanie zawartości pliku linia po linii
with open("tmp.txt") as fp:
    line = fp.readline()
    while line != '':
        print("Linia z pliku: ", line)
        line = fp.readline()
```



Operacje wejścia/wyjścia

Pliki - Odczyt/Zapis danych

Odczytanie kolejnych linii

z pliku może być
zrealizowane za pomocą pętli
for.

```
# odczytanie zawartości pliku linia po linii
with open("tmp.txt") as fp:
    for line in fp:
        print("Linia z pliku: ", line)
```



Operacje wejścia/wyjścia

Pliki - Odczyt/Zapis danych

Za pomocą konstruktora **list()** oraz funkcji **readlines()** można **wczytać wszystkie linie z pliku do listy**.

```
with open("tmp.txt") as fp:  
    lines = list(fp)  
    print(lines)
```

```
with open("tmp.txt") as fp:  
    lines = fp.readlines()  
    print(lines)
```



Operacje wejścia/wyjścia

Pliki - Odczyt/Zapis danych

Dane odczytane z pliku otwartego w trybie binarnym zwracane są w postaci ciągu bajtów, by zamienić je na ciąg znaków (string) muszą zostać zdekodowane.

`bytes.decode(encoding="utf-8", errors="strict")`

`bytearray.decode(encoding="utf-8", errors="strict")`

*Return a string decoded from the given bytes. Default encoding is 'utf-8'. errors may be given to set a different error handling scheme. The default for errors is 'strict', meaning that encoding errors raise a `UnicodeError`. Other possible values are 'ignore', 'replace' and any other name registered via `codecs.register_error()`, see section *Error Handlers*. For a list of possible encodings, see section *Standard Encodings*.*

(źródło: <https://docs.python.org/3/library/stdtypes.html#bytes>)



Operacje wejścia/wyjścia

Pliki - Odczyt/Zapis danych

Zapisanie ciągu znaków do pliku odbywa się za pomocą funkcji **write()** wywoływanej na obiekcie reprezentującym plik.

```
with open("tmp.txt", "w") as fp:  
    fp.write("plik testowy\nwieloliniowy")
```



Operacje wejścia/wyjścia

Pliki - Odczyt/Zapis danych

Dane zapisywane do pliku otwartego w trybie binarnym muszą być w postaci ciągu bajtów, by zamienić ciąg znaków (string) na ciąg bajtów należy napis zakodować.

`str.encode(encoding="utf-8", errors="strict")`

Return an encoded version of the string as a bytes object. Default encoding is 'utf-8'. errors may be given to set a different error handling scheme. The default for errors is 'strict', meaning that encoding errors raise a UnicodeError. Other possible values are 'ignore', 'replace', 'xmlcharrefreplace', 'backslashreplace' and any other name registered via codecs.register_error(), see section Error Handlers. For a list of possible encodings, see section Standard Encodings.

(źródło: <https://docs.python.org/3/library/stdtypes.html#bytes>)



Operacje wejścia/wyjścia

Pytania

1. Za co odpowiada instrukcja `with` w przypadku operacji na plikach?
2. W jaki sposób wczytać wszystkie linie z pliku do listy?



Operacje wejścia/wyjścia

Literatura

1. Input and Output - Reading and Writing Files,
<https://docs.python.org/3/tutorial/inputoutput.html#reading-and-writing-files>



Operacje wejścia/wyjścia

