

Temat1: Operacje na plikach w Pythonie

Programowanie to przede wszystkim bezpieczeństwo naszych aplikacji. W Pythonie umożliwiają nam to operacje na plikach. Aby poznać sposób pracy na plikach proszę zapoznać z materiałem poniżej

Wyobraź sobie, że napisałeś rewelacyjny program do obsługi klientów hotelu. Brzmi świetnie, ale dane wszystkich klientów byłyby przechowywane w pamięci operacyjnej tak długo, jak długo uruchomiony byłby Twój program. A co w przypadku awarii komputera lub systemu operacyjnego? Wszystko musiałbyś wpisywać od nowa. I tutaj z pomocą przychodzą nam operacje na plikach. Potrzebne dane możesz zapisywać do pliku znajdującego się na dysku komputera i odczytywać je w razie potrzeby. Prawda, że to wspaniałe? Zobaczmy, jak to zrobić w języku Python. W tym celu spójrz na listing 11.1.

LISTING 11.1. Operowanie na pliku tekstowym

```
1 plik = open("plik.txt", "w")
2 plik.write("Pierwsze zdanie\n")
3 plik.close()
4
5 plik = open("plik.txt", "a")
6 plik.write("Drugie zdanie\n")
7 plik.close()
8
9 plik = open("plik.txt", "r")
10 zawartosc = plik.read()
11 plik.close()
12
13 print(zawartosc)
```

Na początku musimy stworzyć zmienną, która będzie tzw. **uchwytem do pliku**. Nie przejmuj się zbytnio terminologią. Najważniejsze, abyś wiedział, że ta zmienna reprezentuje plik w kodzie programu. W naszym przypadku nazywa się ona `plik`

itworzymy ją w linii 1. Następnie za pomocą funkcji `open` przypisujemy do niej plik zapisany na dysku. Wewnątrz metody `open` znajdują się dwa parametry. Pierwszy to nazwa ścieżki do pliku (czyli np. `C:\Python34\plik.txt`). Jeżeli plik, do którego chcemy zapisać nasze dane, jest położony w tym folderze, w którym znajduje się nasz kod źródłowy, wystarczy podać jego nazwę, tak jak zrobiliśmy to w naszym programie. Drugi parametr oznacza tryb, w jakim chcemy otworzyć plik:

`r` — otwarcie pliku tylko do odczytu.

`r+` — otwarcie pliku do odczytu i zapisu i ustawienie na jego początek.

`w` — otwarcie pliku do zapisu. Jeśli plik istnieje, zostanie nadpisany, jeśli nie istnieje, zostanie utworzony.

`w+` — otwarcie pliku do zapisu i odczytu. W przypadku zapisu nadpisuje istniejący plik lub tworzy go.

`a` — otwarcie pliku w celu dodania danych.

`a+` — otwarcie pliku w celu dodania danych lub ich odczytu i ustawienie na jego koniec.

W linii 2 używamy metody `write()` do zapisania danych do pliku. Zauważ, że na końcu znajduje się znak `'\n'`. Jest to znak końca linii i spowoduje on przejście naszego programu do nowej linii. Spróbuj wywołać nasz program bez niego i zaobserwuj różnicę. W linii 3 znajduje się NIEZWYKLE ważna funkcja odpowiedzialna za zamykanie pliku. Zapamiętaj, że ZAWSZE po otwarciu pliku (czyli w języku Python wywołaniu funkcji `open`) należy go **zamknąć** (czyli wywołać funkcję `close()`). W przeciwnym razie może to spowodować szereg nieprzewidzianych problemów. W linii 5 ponownie otwieramy plik, jednak tym razem w celu dodania kolejnych informacji (stąd parametr `'a'` wewnątrz metody `open`). Następnie w linii 6 zapisujemy dane do pliku, a w linii 7 zamykamy plik. Na koniec w linii 9 otwieramy plik jedynie do odczytu i w linii 10 zapisujemy jego zawartość do zmiennej `zawartosc`.

Oczywiście wszystkie powyższe zabiegi można by przeprowadzić przy pojedynczym otwarciu i zamknięciu pliku, jednak program został napisany w taki sposób, aby lepiej zobrazować operacje na plikach.

Spójrz teraz na listing 11.2.

LISTING 11.2. Inne tryby potrzebne do operowania na plikach tekstowych

```
1 plik = open("plik.txt", "w")
2 plik.write("Ala ma kota.")
3 plik.close()
4
5 plik = open("plik.txt", "r+")
```

```
▲
6  zawartosc = plik.read()
7  print(zawartosc)
8
9  nowaZawartosc = zawartosc.replace("kota", "psa")
10 print(nowaZawartosc)
11 plik.write(nowaZawartosc)
12 plik.close()
```

Na początku tworzymy prosty plik tekstowy z zawartością "Ała ma kota.". Równie dobrze mógłbyś to zrobić ręcznie. Następnie w linii 5 otwieramy plik ponownie z parametrem "r+". Zawartość pliku wczytujemy do zmiennej `zawartosc` w linii 6. W linii 9 tworzymy nową zmienną o nazwie `nowaZawartosc`, a następnie przypisujemy do niej wynik działania metody `replace()`. Powoduje ona zamianę ciągu znaków "kota" na ciąg znaków "psa". Całość zapisujemy do pliku.

We wszystkich powyższych przypadkach operowaliśmy na plikach tekstowych, jednak czasami potrzebujemy zapisać niestandardowe dane, np. niedawno poznany słownik. Wtedy musimy użyć **plików binarnych**, czyli takich, które bezpośrednio rozumie system operacyjny. Jest to nieco bardziej skomplikowane zagadnienie, ale postaraj się je potraktować jako coś ekstra. Nie zrażaj się, jeśli na razie nie zrozumiesz go do końca. Ja też wielu rzeczy na początku nie rozumiałem. To przyszło z czasem i z doświadczeniem.

W języku Python można również wykonywać operacje na plikach binarnych, tzn. nie tekstowych, a takich, jakie są czytelne dla systemu operacyjnego. Jest to jednak dość zawile i uważam, że nie ma najmniejszego sensu, abyś uczył się tego w tym momencie. Ważne, abyś wiedział, że do operowania na plikach binarnych należy dodać do trybu otwierania literkę `b`, np.:

```
plik = open("plik", "wb")
```

Temat2: Przegląd języków programowania

Kolejna ostatnia lekcja w tym roku szkolnym. Podstawy programowania są tylko w klasie pierwszej. To nie znaczy, że programowania nie będzie. Będzie na pewno i to bardziej zaawansowane. Programowanie aplikacji internetowych to kolejny przedmiot, który Was czeka. Nie tylko będziemy tworzyć aplikacje internetowe. Wrócimy do JavaScriptu (bardziej zaawansowanego, biblioteka jquery, będzie też php, współpraca z bazą mysql. Poznamy też C++. Języki programowania to kolejny duży dział informatyki. Powstają coraz to nowsze bardziej "obiektywne" z licznymi dostępnymi bibliotekami. Zachęcam do poznania jednego języka programowania na poziomie wysokim. Wtedy łatwiej nam jest przejść na inny. Różnica dotyczy składni, natomiast wszędzie występują: pętle, instrukcje warunkowe, funkcje, tablice, obiekty, metody. Mało było na zajęciach o algorytmach dlatego zachęcam do odświeżenia tego zagadnienia.

pozdrawiam
Wojciech Motus