

Konwersje między typami sekwencyjnymi w Pythonie

Ponieważ typy sekwencje są zmienne lub niezmiennie (mutowalne lub niemutowalne), może zaistnieć potrzeba konwersji między tymi typami. Ponadto przy wczytywaniu i zapisywaniu danych do plików tekstowych może pojawić się konieczność wykonania konwersji, gdyż w plikach przechowywane są dane typu tekstowego. W tabeli podano podstawowe metody konwersji między typami sekwencyjnymi.

Konstrukcja języka	Opis konwersji
<code>liczba = int(napis)</code>	napis na liczbę
<code>napis = str(liczba)</code>	liczba na napis
<code>lista = list(napis)</code>	napis na listę
<code>lista = list(krotka)</code>	krotka na listę
<code>krotka = tuple(napis)</code>	napis na krotkę
<code>krotka = tuple(lista)</code>	lista na krotkę
<code>''.join(['x', 'y', 'z'])</code> <code>' '.join(['x', 'y', 'z'])</code>	lista na napis, wyniki: 'xyz' 'x y z'
<code>s = list(s)</code> . . <code>s = ''.join(s)</code>	konwersja tekstu na listę (a następnie listy na tekst) stosuje się, gdy w programie chcemy wykonywać przypisywanie wartości elementom listy, ale wczytaliśmy napis

Operacje na plikach tekstowych

Podobnie jak w innych językach programowania, język Python umożliwia wczytywanie danych z plików oraz ich zapisywanie do pliku. W poniższych tabelach zawarte są informacje na temat podstawowych metod umożliwiających obsługę plików tekstowych oraz trybu dostępu do tych plików.

Konstrukcja języka	Opis	Przykład
<code>open("nazwa_pliku.txt", "tryb")</code>	Otwieranie pliku w wybranym trybie dostępu (tryby opisane są w następnej tabeli)	<code>odczyt.open("dane.txt", "r")</code> <code>zapis.open("dane.txt", "w")</code>
<code>close()</code>	zamykanie pliku (po zamknięciu niemożliwe jest wykonywanie operacji odczytywania i zapisywania)	<code>odczyt.close()</code> <code>zapis.close()</code>
<code>read(rozmiar)</code>	odczytywanie z pliku podanej przez parametr <code>rozmiar</code> liczby znaków w postaci napisu. Jeśli <code>rozmiar</code> nie jest określony, odczytywane są wszystkie znaki z pliku od bieżącej pozycji do końca pliku	<code>odczyt.read(5)</code> <code>odczyt.read()</code>
<code>readline(rozmiar)</code>	odczytywanie z bieżącego wiersza pliku podanej przez parametr <code>rozmiar</code> liczby znaków w postaci napisu. Jeśli <code>rozmiar</code> nie jest określony, odczytywane są wszystkie znaki z bieżącego wiersza od bieżącej pozycji do końca wiersza	<code>odczyt.readline(5)</code> <code>odczyt.readline()</code>
<code>readlines()</code>	odczytywanie wszystkich wierszy pliku jako listy	<code>odczyt.readlines()</code>
<code>write(dane)</code>	zapisywanie napisu <code>dane</code> do pliku	<code>zapis.write(napis)</code>
<code>writelines(dane)</code>	zapisywanie listy <code>dane</code> do pliku w formie napisów	<code>zapis.writelines(lista)</code>

Tryb	Opis
"r"	jeśli plik istnieje, odczytywane są dane z pliku tekstowego, w przeciwnym wypadku pojawia się komunikat o błędzie
"w"	jeśli plik istnieje, zawarte w nim dane zostają zastąpione nowymi, w przeciwnym wypadku zostaje utworzony nowy plik
"a"	jeśli plik istnieje, nowe dane dopisywane są na końcu pliku tekstowego, w przeciwnym wypadku zostaje utworzony nowy plik
"r+"	odczytywanie danych z pliku tekstowego i zapisywanie do niego. Jeśli plik nie istnieje, pojawia się komunikat o błędzie
"w+"	zapisywanie danych do pliku tekstowego i odczytywanie ich z tego pliku. Jeśli plik istnieje, dane w nim zawarte zostają zastąpione nowymi, w przeciwnym wypadku zostaje utworzony nowy plik
"a+"	dopisywanie danych do pliku tekstowego i odczytywanie danych z tego pliku. Jeśli plik istnieje, nowe dane dopisywane są na jego końcu, w przeciwnym wypadku zostaje utworzony nowy plik

Przykład 1.

W pliku `dane_p1.txt`, w oddzielnych wierszach, znajduje się 70 słów o długości od 3 do 14 znaków. Napisz program, który wczytuje dane z tego pliku, a następnie przepisuje do pliku `wyniki_p1.txt` tylko te słowa, które mają parzystą liczbę znaków.

```
dane = open("dane_p1.txt", "r")
wyniki = open("wyniki_p1.txt", "w")
for k in dane:
    if len(k.rstrip()) % 2 == 0:
        wyniki.write(k)
dane.close()
wyniki.close()
```

Przykład 2.

W pliku `dane_p2.txt`, w oddzielnych wierszach, znajdują się 100 liczb całkowitych. Napisz program, który wczytuje dane z pliku wejściowego i przepisuje do pliku `wyniki_p2.txt` tylko te kwadraty liczb zapisanych w kolejnych wierszach, których wartość jest podzielna przez 3.

```
dane = open("dane_p2.txt", "r")
wyniki = open("wyniki_p2.txt", "w")
for k in dane:
    w = int(k) ** 2
    if w % 3 == 0:
        wyniki.write(str(w)+'\n')
dane.close()
wyniki.close()
```

Ćwiczenie 1.

Napisz program, który wykonuje kolejno następujące operacje:

- tworzy plik o nazwie `dane_c1.txt` i przepisuje do niego w kolejnych wierszach nazwy kolejnych miesięcy roku. Nazwy miesięcy są zdefiniowane w programie w postaci listy,
- wypisuje na ekranie tylko te miesiące zawarte w pliku `dane_c1.txt`, których nazwa zaczyna się na literę "m".
- wypisuje do nowego pliku o nazwie `wyniki_c1.txt` nazwy wszystkich miesięcy znajdujących się w pliku `dane_c1.txt`, których nazwa liczy więcej niż 7 znaków.

Ćwiczenie 2.

W pliku `dane_c2.txt` znajduje się w oddzielnych wierszach 100 par oddzielonych spacją liczb z przedziału `[5, 500]`. Napisz program, który odczytuje dane z pliku wejściowego, a następnie przepisuje do pliku `wyniki_c2.txt` tylko te iloczyny par liczb zapisanych w kolejnych wierszach, które mieszczą się w przedziale `[250, 10000]`.

Zadanie domowe 1.

Napisz program, który sprawdza, czy dwa wyrazy podane przez użytkownika są anagramami, czyli wyrazami, które powstały w wyniku przestawienia tych samych liter. Takimi wyrazami są przykładowo `algorytm` i `logarytm`.

Zadanie domowe 2.

Metoda szyfrowania płotowego polega na tym, że znaki tekstu jawnego zapisuje się w taki sposób, aby tworzyły kształt przypominający płot. Tekst zaszyfrowany (szyfrogram) otrzymuje się odczytując kolejne wiersze tak utworzonej konstrukcji. Klucz tego algorytmu to wysokość płotu, czyli liczba wierszy.

Zaszyfrujmy dla przykładu wyraz `KRYPTOANALIZA` stosując płot o wysokości 3 wierszy. Po wprowadzeniu tekstu jawnego do tablicy uzyskujemy następujący efekt:

K				T				A				A
	R		P		O		N		L		Z	
		Y				A				I		

Odczytując szyfrogram wierszami uzyskujemy wynik: `KTAARPONLZYAI`.

Na podstawie powyższych informacji napisz program, który szyfruje i deszyfruje wyraz podany przez użytkownika metodą płotową.