Mark Lassoff

Programowanie dla początkujących

Poznaj świat programowania!

learntoprogram.tv

Helion

Tytuł oryginału: Programming for Absolute Beginners

Tłumaczenie: Piotr Rajca

Projekt okładki: Studio Gravite / Olsztyn; Obarek, Pokoński, Pazdrijowski, Zaprucki
Materiały graficzne na okładce zostały wykorzystane za zgodą Shutterstock Images LLC.

ISBN: 978-83-283-1908-0

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *http://helion.pl* (księgarnia internetowa, katalog książek)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
http://helion.pl/user/opinie/propoc
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

SPIS TREŚCI

Rozdział 5.

Operacje na danych

W poprzednim rozdziale poznałeś struktury sterujące, instrukcje warunkowe oraz pętle — konstrukcje, które mają kluczowe znaczenie we wszystkich językach programowania, pozwalają one bowiem programom podejmować decyzje na podstawie kryteriów określanych przez programistów. Z kolei w tym rozdziale skoncentrujemy się na danych i sposobach ich przechowywania. Przedstawię w nim trzy struktury danych: listy, krotki oraz słowniki. Dotychczas nie poświęcaliśmy zbyt wiele uwagi zagadnieniom związanym z danymi, ale najłatwiej można sobie je wyobrazić jako zorganizowane informacje.

Różne typy informacji są organizowane na różne sposoby. Na przykład, jeśli dane dotyczą planowanych zakupów, to optymalnym sposobem ich organizacji będzie lista. Jeżeli jednak konieczne będzie zapisanie ocen uczniów z różnych przedmiotów w pewnym okresie czasu, to lista już w ogóle się do tego nie nada. Istnieje wiele różnych struktur danych, których używają programiści, a każda z nich jest zoptymalizowana pod kątem innego typu danych.

Zacznijmy od prostej listy zakupów.

Listy

Zapewne każdy kiedyś korzystał z listy zakupów, lecz nie wszyscy mieli okazję zapisywać ją w kodzie programu w języku Python. Poniższy kod pokazuje, jak można to zrobić:

```
groceries = [ "jablka", "ziemniaki", "chleb", "mleko", "maslo",
➥"pomarancze", "salata", "cukier", "platki" ]
```

W odróżnieniu od zmiennych, które przechowują jedną daną, powyższa lista umożliwiła zapisanie wielu elementów. Bez trudu można sobie wyobrazić mnóstwo sytuacji, w których zastosowanie listy byłoby bardzo wygodne. W powyższym przykładzie wszystkie *elementy* listy są łańcuchami znaków. Równie dobrze można jednak tworzyć listy zawierające liczby zmiennoprzecinkowe, liczby całkowite lub liczby obu tych typów. Poniżej przedstawiłem listę liczb zmiennoprzecinkowych:

Elementy

```
gpas = [ 3.25, 2.26, 1.99, 3.55, 4.0, 3.21, 2.56, 3.06 , 2.72 ]
```

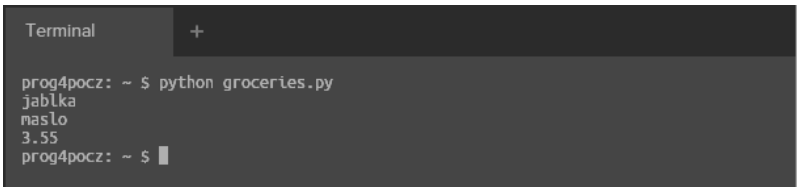
A teraz zacznie się robić ciekawiej. Ponieważ w języku Python listy są traktowane jak obiekty, udostępniają kilka wbudowanych poleceń umożliwiających operowanie na nich i na ich zawartości. To ma naprawdę duże znaczenie, gdyż dostępne funkcje pozwalają na wykonywanie takich operacji jak zliczanie liczby elementów bądź ich sortowanie. Oczywiście takie operacje można także wykonywać, używając samodzielnie napisanego kodu, lecz wymagałoby to naprawdę wiele pracy.

UWAGA: Więcej informacji o **obiektach** znajdziesz w następnym rozdziale, dlatego jeśli to pojęcie nie jest Ci jeszcze znane, nie masz się czym przejmować.

Zanim przejdziemy do kolejnych, bardziej złożonych zagadnień, warto jeszcze poświęcić nieco uwagi sposobom pobierania danych z list. W tym celu wpisz w edytorze program, którego kod zawiera dwie przedstawione wcześniej listy i kilka dodatkowych instrukcji:

```
groceries = [ "jablka", "ziemniaki", "chleb", "mleko", "maslo",
↪"pomarancze", "salata", "cukier", "platki" ]
gpas = [ 3.25, 2.26, 1.99, 3.55, 4.0, 3.21, 2.56, 3.06 , 2.72 ]
print groceries[0]
print groceries[4]
print gpas[3]
```

Kiedy już upewnisz się, że poprawnie wpisałeś kod programu, zapisz go i wykonaj za pomocą polecenia `python`. Jego wyniki będą takie same jak na rysunku 5.1.



Rysunek 5.1. Program wyświetlający dane pobierane z listy wykonany w serwisie Koding.com

W powyższym przykładzie szczególną uwagę zwróć na instrukcje `print`. Jak widać, każda z nich odwołuje się do jednej ze zdefiniowanych wcześniej list i do konkretnego indeksu w ramach danej listy zapisanego w nawiasach kwadratowych. W języku Python, podobnie jak w większości innych języków programowania, listy są indeksowane, co oznacza, że każdemu ich elementowi jest przyporządkowywana liczba, której można używać, by odwołać się do danego elementu. Pierwszy element listy zawsze ma indeks o wartości 0.

Indeks

W przypadku przedstawionej w przykładzie listy zakupów indeksy jej elementów będą miały następującą postać:

Indeks	Element listy
0	jablka
1	ziemniaki
2	chleb
3	mleko
4	maslo
5	pomarancze
6	salata
7	cukier
8	platki

Odwołanie służące do pobierania danych z list ma następującą, ogólną postać, przedstawioną tutaj w połączeniu z poleceniem `print`:

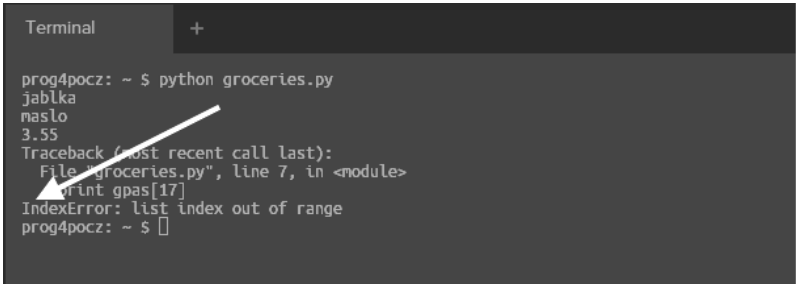
```
print
nazwaListy[indeks]
```

Jeśli masz jakieś doświadczenia związane z korzystaniem z innych języków programowania, to zauważyłeś zapewne, że listy są bardzo podobne do tablic. Jeśli to zauważyłeś, to miałeś rację!

W kontekście listy trzeba pamiętać o jeszcze jednym zagadnieniu. Otóż długość list zależy wyłącznie od ich zawartości. Jakakolwiek próba odwołania się do nieistniejącego indeksu spowoduje zakończenie programu i wyświetlenie komunikatu o błędzie. W ramach testu dodaj do poprzedniego programu poniższy wiersz kodu:

```
print gpas[17]
```

Po wprowadzeniu tej zmiany próba wykonania programu spowoduje wyświetlenie wyników przedstawionych na rysunku 5.2.



Rysunek 5.2. Próba odwołania się do indeksu spoza zakresu listy spowoduje wyświetlenie błędu

EDYCJA ELEMENTÓW LIST

Indeksów można także używać do zmieniania wartości elementów list. Do tego celu używana jest instrukcja przypisania, taka sama jak w przypadku przypisywania wartości do zwyczajnych zmiennych. Poniższa instrukcja zmieni wartość pierwszego elementu listy `gpas`:

```
gpas[0] = 4.0
```

Zwróć uwagę, że ta instrukcja nie zmienia indeksów pozostałych elementów listy.

STOSOWANIE FUNKCJI LIST

Jak już wspomniałem, listy są obiektami, a to daje możliwość stosowania wielu funkcji skojarzonych z tymi obiektami. Warto zajrzeć do oficjalnej dokumentacji języka, by poznać dostępne funkcje operujące na listach.

> Dokumentacja języka Python obejmująca funkcje wykonujące operacje na listach jest dostępna na stronie *https://docs.python.org/2/tutorial/datastructures.html*.

append()

Aby przetestować jedną z tych funkcji, a konkretnie funkcję `append()`, dodaj do poprzedniego programu dwa poniższe wiersze kodu:

```
groceries.append("kurczak")
print groceries
```

Jak widać, funkcja `append()` została poprzedzona kropką i dodana do nazwy listy. W powyższym przypadku wywołanie funkcji `append()` doda łańcuch znaków `"kurczak"` do pierwszego dostępnego elementu listy. Oczywistym jest, że zastosowanie tej funkcji spowodowało powiększenie długości listy.

Po dodaniu powyższego kodu do wcześniejszego programu i jego wykonaniu program wyświetli wyniki przedstawione na rysunku 5.3.

Rysunek 5.3. Warto zauważyć, że w zawartości listy wyświetlonej za pomocą polecenia print znalazł się także łańcuch znaków "kurczak"

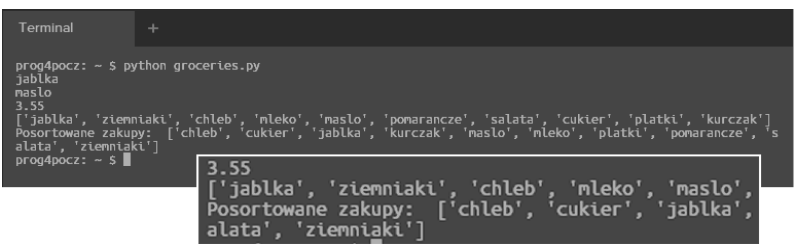
Warto przetestować jeszcze jedną z funkcji operujących na listach. Funkcja `sort()` robi dokładnie to, co sugeruje jej nazwa — sortuje elementy listy.

Można jej używać analogicznie jak funkcji `append()`. Poniżej przedstawiłem kolejne dwa wiersze kodu, które należy dodać do przykładowego programu.

sort()

```
groceries.sort()
print "Posortowane zakupy: ", groceries
```

Uruchom teraz ten programu za pomocą polecenia `python`, a wygeneruje on wyniki przedstawione na rysunku 5.4.



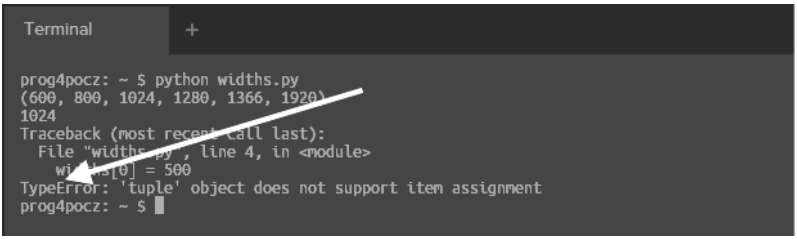
Rysunek 5.4. Po wywołaniu funkcji sort() lista groceries zostanie przeindeksowana. W elemencie o indeksie 0 zostanie zapisany łańcuch "chleb", w elemencie o indeksie 1 — łańcuch "cukier", i tak dalej

KROTKI

Krotki są bardzo podobne do list, choć różnią się od nich pod jednym bardzo ważnym względem: nie są dynamiczne. Po zdefiniowaniu krotki jej elementy nie mogą być zmieniane. Zacznijmy od napisania nowego programu, demonstrującego tworzenie i stosowanie krotek:

```
widths = (600, 800, 1024, 1280, 1366, 1920)
print widths
print widths[2]
widths[0] = 500
```

Po uruchomieniu program ten wygeneruje wyniki przedstawione na rysunku 5.5.



Rysunek 5.5. Warto zwrócić uwagę, że można się odwoływać zarówno do całej krotki, jak i do jej konkretnego elementu, jednak próba przypisania wartości wybranemu elementowi krotki spowodowała zgłoszenie błędu. Krotki nie są dynamiczne, dlatego po zainicjowaniu ich wartości nie można zmieniać

Krotki służą do przechowywania danych, które nie zmieniają się w trakcie działania programu. Na przykład w programie przedstawionym powyżej jest bardzo mało prawdopodobne, by zdefiniowane w krotce szerokości ekranów uległy zmianie.

SŁOWNIKI

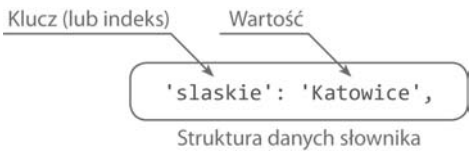
Słowniki są kolejną strukturą danych dostępną w języku Python. Idealnie się one nadają do przechowywania par danych, określanych także czasami jako **pary klucz – wartość**. W przypadku danych tego typu, zamiast indeksów liczbowych, każda informacja jest indeksowana przy użyciu klucza, którego postać określa programista. Przykładami danych, które z powodzeniem można zapisywać za pomocą słowników, są nazwy województw i ich stolic albo nazwy miast i ich kody pocztowe.

Przeanalizujmy następujący program demonstrujący zastosowanie słowników:

```
states = { 'dolnoslaskie' : 'Wroclaw',
           'lubelskie': 'Lublin',
           'malopolskie': 'Krakow',
           'pomorskie': 'Gdansk',
           'slaskie': 'Katowice',
           'wielkopolskie': 'Poznan' }

print states['slaskie']
```

Po uruchomieniu powyższego programu w wierszu poleceń wyświetli on łańcuch znaków `slaskie`. Najważniejsze jednak w tym przykładzie jest to, że kluczem skojarzonym z wartością `"Katowice"` jest `"slaskie"`. Zwrócić uwagę na polecenie `print` oraz sposób, w jaki odwołuje się ono do wartości zapisanej w słowniku (przedstawiony na rysunku 5.6).



Rysunek 5.6. Sposób odwoływania się do wartości przechowywanych w słowniku

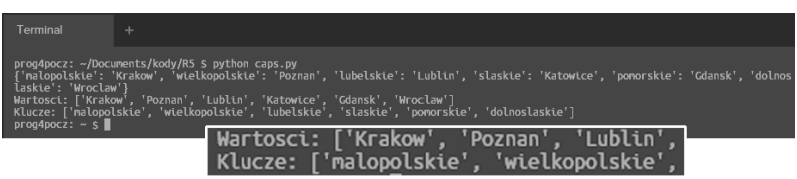
W języku Python słowniki, podobnie jak listy, są obiektami, z którymi są skojarzone funkcje. Przyjrzymy się zatem nieco dokładniej jednej z takich funkcji.

Zmodyfikuj w poniższy sposób kod poprzedniego programu:

```
states = { 'dolnoslaskie' : 'Wroclaw',
           'lubelskie': 'Lublin',
           'malopolskie': 'Krakow',
           'pomorskie': 'Gdansk',
           'slaskie': 'Katowice',
           'wielkopolskie': 'Poznan' }

print states
print "Wartosci:", states.values()
print "Klucze:", states.keys()
```

Kiedy wprowadzisz te zmiany i uruchomisz program z poziomu wiersza poleceń, wyświetli on wyniki przedstawione na rysunku 5.7.



Rysunek 5.7. Wyniki przedstawiające zastosowanie dwóch funkcji słowników: values() i keys()

Jak widać, funkcja `values()` zwraca wszystkie wartości zapisane w słowniku, natomiast funkcja `keys()` — wszystkie klucze.

Ten krótki rozdział może stanowić jedynie ogólny wstęp do zagadnień korzystania z danych w pisanych programach. Niemal każdy program, w mniejszym lub większym stopniu, operuje na danych. Dlatego efektywne posługiwanie się danymi jest bardzo ważnym elementem zestawu narzędzi każdego programisty.

SKOROWIDZ

N

O

P

S

T

V

W

Z

Jeżeli chcesz się nauczyć programować, musisz poznać podstawowe zasady i typowe konstrukcje — pętle, instrukcje warunkowe i zmienne, które są wspólne dla wszystkich powszechnie używanych języków programowania. Jeżeli zamierzasz ugruntować fundamenty, by uczyć się konkretnych języków, sięgnij po tę książkę.

Dzięki niej napiszesz i uruchomisz swój pierwszy program. Następnie nauczysz się pobierać dane od użytkownika, zapisywać wyniki w plikach oraz zrozumiesz, jak działają zmienne. Kolejny krok to nauka korzystania z pętli i instrukcji warunkowych — w ten sposób będziesz w stanie zaimplementować wiele nawet bardzo zaawansowanych programów. Mark Lassoff wszystkie koncepcje prezentuje na przykładzie języka Python, którego składnia jest zbliżona do innych języków, takich jak JavaScript, Java czy C. Sięgnij po tę książkę i wkrocz w świat programowania!

Dzięki tej książce:

- Poznasz podstawy programowania
- Nauczysz się korzystać z podstawowych konstrukcji — pętli, instrukcji warunkowych
- Wykorzystasz zmienne
- Pobierzesz dane od użytkownika oraz zapiszesz wyniki do pliku
- Zgłębisz podstawy języka Python

Wkrocz w świat programowania!

Mark Lassoff — urodzony programista. Swoją przygodę z programowaniem rozpoczął na komputerze Commodore 64, tworząc gry w języku BASIC. Ma bogate doświadczenie w pisaniu oprogramowania, stron oraz aplikacji internetowych. Jego pasją jest uczenie programowania innych — dziesięć lat działalności w tym obszarze sprawia, że dziś jest uznawany za jednego z najlepszych nauczycieli.

Helion

37818 numer katalogowy

księgarnia internetowa

http://helion.pl

zamówienia telefoniczne

0 801 339900

0 601 339900

Sprawdź najnowsze promocje:
http://helion.pl/promocje
Książki najchętniej czytane:
http://helion.pl/bestsellery
Zamów informacje o nowościach:
http://helion.pl/nowosci

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
http://helion.pl

ISBN 978-83-283-1908-0

9 788328 319080

Informatyka w najlepszym wydaniu

cena: 19,90 zł