

2

ZAMIENIANIE ŻMUDNEJ ARYTMETYKI W ZABAWĘ PRZY UŻYCIU LIST I PĘTLI

*To znaczy, że jutro znowu muszę tam iść?
Aidan Farrell po pierwszym dniu w szkole*



Większości osób słowo matematyka kojarzy się z wykonywaniem operacji arytmetycznych: dodawaniem, odejmowaniem, mnożeniem i dzieleniem. Choć można dość łatwo wykonywać te obliczenia przy użyciu kalkulatorów i komputerów, często wiąże się to z realizowaniem wielu powtarzalnych zadań. Na przykład, aby dodać dwadzieścia różnych liczb za pomocą kalkulatora, trzeba nacisnąć przycisk $+$ dziewiętnaście razy!

W tym rozdziale pokażemy, jak za pomocą Pythona automatyzować pewne żmudne aspekty operacji arytmetycznych. Na początku omówimy operatory matematyczne i różne typy danych, jakie można wykorzystywać w Pythonie. Następnie opiszemy metody przechowywania i obliczania wartości przy użyciu zmiennych. Wytlumaczymy również, jak używać list i pętli do powtarzania kodu. Na zakończenie połączymy te pojęcia programistyczne, aby napisać funkcje, które wyręczą nas w wykonywaniu złożonych obliczeń, automatyzując ten proces. Jak pokażemy, Python może być dużo lepszym kalkulatorem niż jakikolwiek tradycyjny kalkulator dostępny na rynku, i co więcej, jest całkowicie darmowy!

Podstawowe operatory

Wykonywanie operacji arytmetycznych w interaktywnej powłoce Pythona jest proste – wystarczy wpisać wyrażenie i nacisnąć klawisz ENTER. W tabeli 2.1 przedstawione zostały najważniejsze operatory matematyczne.

Tabela 2.1. Najpopularniejsze operatory matematyczne w Pythonie

Operator	Składnia
Dodawanie	+
Odejmowanie	-
Mnożenie	*
Dzielenie	/
Potęgowanie	**

Otwórzmy powłokę Pythona, aby przetestować wybrane operatory arytmetyczne, korzystając z przykładów z listingu 2.1.

```
>>> 23 + 56 # Dodawanie
79
>>> 45 * 89 # Operatorem mnożenia jest gwiazdka
4005
>>> 46 / 13 # Operatorem dzielenia jest ukośnik
3.5384615384615383
>>> 2 ** 4 # 2 do 4-tej potęgi
16
```

Listing 2.1. Testowanie wybranych operatorów matematycznych

Wyniki obliczeń powinny zostać wyświetlone w postaci danych wyjściowych. Można użyć spacji do zwiększenia czytelności kodu ($6 + 5$) lub też nie ($6+5$), ponieważ dla Pythona nie stanowi to różnicy przy wykonywaniu działań.

Warto mieć świadomość, że dzielenie w wersji Python 2 jest nieco kłopotliwe. Na przykład, gdy Python 2 zobaczy $46/13$, przyjmie założenie, że interesują nas jedynie liczby całkowite i w związku z tym zwróci liczbę całkowitą (3) zamiast liczby dziesiętnej, jak na listingu 2.1. Ponieważ pobraliśmy wersję Python 3, nie powinniśmy mieć tego problemu. Jednak stosowany w kolejnych rozdziałach pakiet graficzny wykorzystuje wersję Python 2, dlatego wykonując dzielenie, będziemy musieli sygnalizować, że interesują nas wartości dziesiętne.

Działania na zmiennych

Możemy również stosować operatory na zmiennych. W rozdziale 1 pokazaliśmy, jak używać zmiennych podczas definiowania funkcji. Podobnie jak w algebrze, w programowaniu zmienne umożliwiają dzielenie długich, skomplikowanych obliczeń na szereg etapów za sprawą przechowywania wyników, które można w przyszłości wykorzystać ponownie. Na listingu 2.2 pokazano, w jaki sposób można użyć zmiennych do przechowania liczb i wykonywania na nich działań, niezależnie od wartości tych zmiennych.