



# Instrukcja input i funkcje matematyczne

Wprowadzenie do języka Python (III)





Niezwykle ważne jest, aby móc uzyskać dane wejściowe od użytkowników, a Python to bardzo ułatwia.

Funkcja `input()` wyświetla komunikat, a następnie przypisuje użytkownikom dane wejściowe aż do naciśnięcia klawisza return

(wydaj znak nowej linii) do zmiennej (zmiennych). Wartości przekazywane do polecenia `'input'` są zawsze łańcuchami tekstowymi, więc przygotuj się do rzutowania, jeśli zajdzie taka potrzeba.

## KOD

```
name = input('Jak masz na imię: ')
```

```
print('Cześć', name)
```





Możesz także wczytać 2 lub więcej wartości za pomocą jednej funkcji `input()`. Tutaj poprosimy o dwie dane, a potem zsumujemy 2 liczby. Odejmę również, pomnożę, podzielę i użyję modułu na wartościach.

Moduł zwraca pozostałą część dzielenia.

Zostanie również stworzone rzutowanie ciągów znaków na liczby całkowite. Funkcja `split()` dzieli dane wprowadzone przez użytkownika na podstawie białych znaków (m.in. spacje) między wartościami.





## KOD

```
# Przypisz 2 wartości, dzieląc dane wejściowe użytkownika uwzględniając spacje
num_1, num_2 = input('Wprowadź 2 liczby:').split()
# Konwertuj łańcuchy na liczby zwykłe (liczby całkowite)
liczba_1 = int(num_1)
liczba_2 = int(num_2)
# Dodaj wprowadzone wartości i zapisz w sumie
suma_1 = liczba_1 + liczba_2
# Odejmij wartości i zachowaj różnicę
różnica = liczba_1 - liczba_2
# Pomnóż wartości i przechowuj w iloczynie
iloczyn = liczba_1 * liczba_2
# Podziel wartości i przechowuj w ilorazie
iloraz = liczba_1 / liczba_2
```





```
# Użyj modułu na wartościach, aby znaleźć resztę
reszta = liczba_1 % liczba_2

# format() ładuje wartości zmiennych w kolejności do {} symboli
zastępczych

print("{} + {} = {}".format(liczba_1, liczba_2, suma_1))
print("{} - {} = {}".format(liczba_1, liczba_2, różnica))
print("{} * {} = {}".format(num_1, num_2, iloczyn))
print("{} / {} = {}".format(liczba_1, liczba_2, iloraz))
print("{} % {} = {}".format(liczba_1, liczba_2, reszta))
```

Funkcja `format()` dopasowuje wartości znalezione w nawiasach po formacie słowa kluczowego `z {}` (nawiasami klamrowymi), które znajdują się w ciągu instrukcji `print`.





## Twój pierwszy problem z Pythonem

Należy napisać program, który prosi użytkownika o wprowadzenie liczby mil. Zamienisz mile na kilometry (kilometry = mile \* 1,60934)

Następnie wydrukuj to na przykład 5 mil to 8,0467 kilometrów

### KOD

```
# Wpisz swój kod tutaj
```





## Rozwiązanie

### KOD

```
# Poproś użytkownika o wprowadzenie mil i przypisanie ich do
zmiennnej mile

mile = input('Wprowadź mile: ')

# Konwertuj z łańcucha na liczbę całkowitą

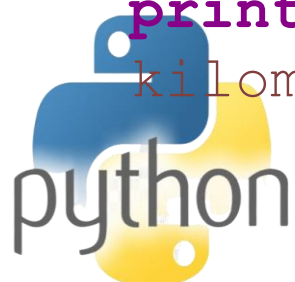
mile = int(mile)

# Wykonaj obliczenia, mnożąc 1,60934 razy mile

kilometry = mile * 1,60934

# Wydrukuj wyniki za pomocą format()

print("{} mile równają się {} kilometrom".format(mile,
kilometry))
```





## Moduł matematyczny

Moduł to plik, który zawiera garść gotowego kodu. Technicznie stworzysz moduł już teraz. Omówię je bardziej szczegółowo później.

Python udostępnia wiele funkcji dzięki swojemu modułowi matematycznemu, takie jak :

### KOD

```
# Zaimportuj moduł matematyczny
```

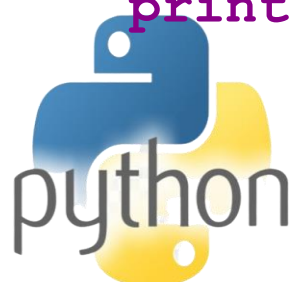
```
import math
```

```
# Ponieważ użyłeś importu, uzyskujesz dostęp do metod, odwołując się do modułu
```

```
print("ceil(4.4) = ", math.ceil(4.4))
```

```
print("floor(4.4) = ", math.floor(4.4))
```

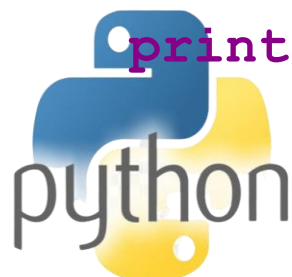
```
print("fabs(-4.4) = ", math.fabs(-4.4))
```







```
# Silnia = 1 * 2 * 3 * 4
print("silnia(4) = ", math.factorial(4))
# Zwróć resztę z dzielenia
print("fmod(5,4) = ", math.fmod(5, 4))
# Skonwertuj liczbę zmiennoprzecinkową i zwróć int
print("trunc(4.2) = ", math.trunc(4.2))
# Zwróć x^y
print("pow(2,2) = ", math.pow(2, 2))
# Zwróć pierwiastek kwadratowy
print("sqrt(4) = ", math.sqrt(4))
# Wartości specjalne
print("mate.e = ", math.e)
print("math.pi = ", math.pi)
```





```
# Zwróć e^x
print("exp(4) = ", math.exp(4))
# Zwróć logarytm naturalny e * e * e ~ 20, więc log(20) mówi
# że e^3 ~ 20
print("log(20) = ", math.log(20))
# Możesz zdefiniować podstawę i 10^3 = 1000
print("log(1000,10) = ", math.log(1000, 10))
# Możesz również użyć podstawy 10 w ten sposób
print("log10(1000) = ", math.log10(1000))
```





```
# Mamy następujące funkcje trygonometryczne
# sin, cos, tan, asin, acos, atan, atan2, asinh, acosh,
# atanh, sinh, cosh, tanh
# Można funkcję sin wykorzystać w ten sposób:
print("sin(0)", math.sin(0))

# Konwertuj radiany na stopnie i na odwrot
print("stopnie(1.5708) = ", math.degrees(1.5708))
print("radians(90) = ", math.radians(90))
```

