

Wprowadzanie danych w Pythonie

Funkcja `input()` pozwala wpisać dane podczas działania programu. Dowiesz się, jak zapisać odpowiedź użytkownika w zmiennej i kiedy użyć `int(input())`, aby wykonywać obliczenia.

Jak działa `input`

```
imie = input("Jak masz na imię? ")
print("Cześć,", imie)
```

Program wyświetla pytanie „Jak masz na imię?” i czeka na odpowiedź. Użytkownik wpisuje tekst i naciska Enter. Wpisana odpowiedź trafia do zmiennej `imie`. Następnie `print()` wyświetla powitanie.

Przykładowy przebieg programu

```
Jak masz na imię? Ola
Cześć, Ola
```

Ola to odpowiedź wpisana przez użytkownika. Tekst pytania w nawiasie `input()` jest komunikatem — nie staje się wartością zmiennej.

Zwykły `input` zawsze zwraca tekst

```
liczba = input("Podaj liczbę: ")
print(liczba)
```

Jeśli wpiszesz 7, zmienna `liczba` będzie przechowywać tekst "7". Nazwa zmiennej nie zmienia rodzaju danych. Samo `print()` pokaże 7 bez cudzysłowów, ale w zmiennej nadal będzie tekst.

Przykład z dwiema odpowiedziami

```
imie = input("Podaj imię: ")
miasto = input("Podaj miasto: ")
print(imie, "mieszka w miejscowości", miasto)
```

Po wpisaniu imienia Ola i miasta Kraków program wypisze:

```
Ola mieszka w miejscowości Kraków
```

Jak wczytać liczbę całkowitą

```
wiek = int(input("Ile masz lat? "))  
print("Za rok będziesz mieć", wiek + 1, "lat.")
```

Najpierw `input()` pobiera odpowiedź jako tekst. Następnie `int()` zamienia ten tekst na liczbę całkowitą. Dopiero ta liczba zostaje przypisana do zmiennej `wiek`.

Gdy użytkownik wpisze 12

```
input() pobiera tekst "12"  
int() zamienia "12" na liczbę 12  
wiek + 1 daje wynik 13
```

Przykładowy przebieg programu

```
Ile masz lat? 12  
Za rok będziesz mieć 13 lat.
```

Najważniejsza różnica

Porównaj dwa programy. W obu użytkownik wpisuje najpierw 2, a potem 3.

Bez `int` — łączenie tekstów

```
a = input("Podaj pierwszą liczbę: ")  
b = input("Podaj drugą liczbę: ")  
print(a + b)
```

Wynik: 23. Zmienne przechowują teksty "2" i "3". Operator `+` łączy je w tekst "23".

Z `int` — dodawanie liczb

```
a = int(input("Podaj pierwszą liczbę: "))  
b = int(input("Podaj drugą liczbę: "))  
print(a + b)
```

Wynik: 5. Zmienne przechowują liczby całkowite 2 i 3, więc operator `+` wykonuje dodawanie.

Zapamiętaj: `input()` daje tekst, a `int(input())` pozwala otrzymać liczbę całkowitą. `int()` przyjmuje także zapis liczby ujemnej, na przykład -4.

Przykład obliczania obwodu prostokąta

```
a = int(input("Podaj długość boku a: "))
b = int(input("Podaj długość boku b: "))
obwod = 2 * a + 2 * b
print("Obwód prostokąta wynosi", obwod)
```

Program pobiera długości dwóch boków jako liczby całkowite. Następnie oblicza obwód i wyświetla wynik. W tym przykładzie wpisuj dodatnie liczby całkowite.

Dla boków 4 i 7 obliczenie to $2 * 4 + 2 * 7 = 22$. Program wypisze: Obwód prostokąta wynosi 22.

Dwa częste błędy

Dodawanie liczby do tekstu

```
wiek = input("Ile masz lat? ")
print(wiek + 1)
```

Ten kod spowoduje błąd `TypeError`: `wiek` jest tekstem, a `1` liczbą. Popraw pierwszą linię na `wiek = int(input("Ile masz lat? "))`.

Wpisanie słowa zamiast liczby

Jeśli program używa `int(input())`, a wpiszesz słowo, na przykład `kot`, pojawi się błąd `ValueError`. Wpisz liczbę całkowitą, na przykład `8`. Pusta odpowiedź też nie jest poprawnym zapisem liczby.

Sprawdź swoją wiedzę

W każdym przykładzie użytkownik wpisuje 5. Co wyświetli program?

Przykład A

```
x = input("Podaj wartość: ")
print(x + x)
```

Przykład B

```
x = int(input("Podaj wartość: "))
print(x + x)
```

Odpowiedzi

A: 55 — połączenie tekstów "5" i "5".

B: 10 — dodanie liczb 5 i 5.