

Tomasz Królikowski

Nr albumu: 153790

Zadanie znajduję się w repozytorium GIT pod adresem:

[https://github.com/krolikowski80/studia\\_WSB/tree/main/Python/intro/zad1](https://github.com/krolikowski80/studia_WSB/tree/main/Python/intro/zad1)

### Zadanie 3.3 - Korzystanie z dokumentacji języka Python

W ramach tego zadania zapoznałem się z dokumentacją Pythona i wyszukałem informacje na temat funkcji wbudowanej, modułu oraz wyjątku.

#### 1. Funkcja wbudowana: `zip()`

Funkcja `zip()` jest bardzo przydatna, gdy chcemy połączyć elementy dwóch (lub więcej) iterowalnych obiektów w krotki. Jeśli jedna z list ma więcej elementów niż druga, `zip()` działa do najkrótszej z nich. Dzięki temu można łatwo tworzyć pary danych, które można później wykorzystywać np. w pętlach.

Przykład użycia funkcji `zip()`:

```
list(zip([1, 2, 3], ['a', 'b', 'c']))  
# Wynik: [(1, 'a'), (2, 'b'), (3, 'c')]
```

**Dokumentacja:** <https://docs.python.org/3/library/functions.html#zip>

#### 2. Moduł: `math`

Moduł `math` zawiera wiele przydatnych funkcji matematycznych. Możemy w nim znaleźć operacje na liczbach, takie jak pierwiastkowanie, potęgowanie, obliczanie logarytmów czy funkcje trygonometryczne. Jest to bardzo użyteczny moduł przy pracy z obliczeniami numerycznymi i analizą danych.

Przykład użycia funkcji `sqrt()` do obliczenia pierwiastka kwadratowego:

```
import math  
math.sqrt(16) # Wynik: 4.0
```

**Dokumentacja:** <https://docs.python.org/3/library/math.html>

#### 3. Wyjątek: `ValueError`

Błąd `ValueError` występuje w sytuacji, gdy funkcja otrzymuje poprawny typ danych, ale w niewłaściwym formacie. Na przykład, gdy próbujemy przekonwertować napis, który nie jest liczbą, na typ `int`, Python zgłasza wyjątek `ValueError`. Jest to częsty błąd, który należy obsługiwać, aby zapobiegać awariom programu.

Przykład błędu **ValueError**:

```
int("abc") # ValueError: invalid literal for int() with base 10: 'abc'
```

**Dokumentacja:** <https://docs.python.org/3/library/exceptions.html#ValueError>

### Zadanie 3.4 - Program w Pythonie

W ramach tego zadania napisałem program, który wykorzystuje:

- Funkcję wbudowaną **zip()** do łączenia dwóch list,
- Moduł **math** do obliczenia pierwiastka kwadratowego,
- Obsługę wyjątku **ValueError** za pomocą **try-except**.

#### Kod programu

```
# Tworzę dwie listy z liczbami i literami
import math
list_a = [1, 2, 3] # Lista liczb
list_b = ['a', 'b', 'c'] # Lista liter
combined = list(zip(list_a, list_b)) # Łączenie list w pary
print("Połączone listy:", combined)

# Importuję moduł math do wykonywania operacji matematycznych
# Obliczam pierwiastek kwadratowy z liczby 16
print("Pierwiastek kwadratowy z 16:", math.sqrt(16))

# Obsługa błędu za pomocą bloku try-except
try:
    # Próbuję przekonwertować napis na liczbę całkowitą
    num = int("abc")
except ValueError as e:
    # Obsługa wyjątku - wyświetlenie komunikatu o błędzie
    print("Błąd ValueError złapany:", e)
```

#### Linki do dokumentacji

<https://docs.python.org/3/library/functions.html#zip> - Funkcja do łączenia iterowalnych obiektów.

<https://docs.python.org/3/library/math.html> - Moduł do obliczeń matematycznych.

<https://docs.python.org/3/library/exceptions.html#ValueError> - Obsługa błędów związanych z niepoprawnymi wartościami.