



Pystart.pl

Wyjątki, obsługa błędów

lekcja czterdziesta druga

PyStart #42 Wyjątki, obsługa błędów

Co może pójść nie tak?



```
1
2 number1 = int(input('Number 1: '))
3 number2 = int(input('Number 2: '))
4
5 result = number2 / number1
6 print(result)
```

2, 3: Input może zwrócić coś czego int nie będzie mógł zrobić liczby.

5: Nie można dzielić przez zero

PyStart #42 Wyjątki, obsługa błędów

Co może pójść nie tak?



```
1
2  number1 = int(input('Number 1: '))
3  number2 = int(input('Number 2: '))
4
5  result = number2 / number1
6  print(result)
```

2, 3: Input może zwrócić coś czego int nie będzie mógł zrobić liczby.

ValueError

5: Nie można dzielić przez zero

ZeroDivisionError

PyStart #42 Wyjątki, obsługa błędów

Jak to zapisać bezpieczniej?



```
1 try:
2     number1 = int(input('Number 1: '))
3     number2 = int(input('Number 2: '))
4     result = number2 / number1
5     print(result)
6 except ValueError:
7     print('Podano nieprawidłową wartość')
8 except ZeroDivisionError:
9     print('Nie dzielimy przez zero')
```

- Try... except
- Ile razy ten except?
- można też
`except (ValueError, ZeroDivisionError)`
- finally

PyStart #42 Wyjątki, obsługa błędów

Zadania dla nabrania wprawy

Przygotuj listę która zawiera 10 elementów. Zapytaj użytkownika o to która z wartości powinna zostać zwrócona. Jeśli użytkownik odwoła się do wartości która nie istnieje (`IndexError`), to wyświetl komunikat, że takiej wartości nie ma.

Przygotuj słownik zawierający 3 wartości. Sprawdź co się stanie jeśli spróbujesz odwołać się od klucza, który nie istnieje w tym słowniku. Spróbuj obsłużyć taką sytuację łapiąc taki wyjątek.

42.1



PyStart #42 Wyjątki, obsługa błędów

Jak wysłać wyjątek?

Jak wyrzucić wyjątek w przypadku gdy podana liczba nie jest podzielna przez 5?

```
1 while True:
2     value = int(input('Podaj liczbę: '))
3     if not value % 5 == 0:
4         raise Exception('Liczba nie jest podzielna przez 5.')
5
6     print(value)
```

Traceback (most recent call last):

File "exceptions-show.py", line 4, in <module>

raise Exception('Liczba nie jest podzielna przez 5.')

Exception: Liczba nie jest podzielna przez 5.

PyStart #42 Wyjątki, obsługa błędów

Ciekawi?



Gdzie należy dodać try except?

To zależy.

PyStart #42 Wyjątki, obsługa błędów

Jak wysłać wyjątek?

Jak wyrzucić wyjątek w przypadku gdy podana liczba nie jest podzielna przez 5?

```
1 while True:
2     try:
3         value = int(input('Podaj liczbę: '))
4         if not value % 5 == 0:
5             raise Exception('Liczba nie jest podzielna przez 5.')
6
7         print(value)
8     except Exception as e:
9         print(e)
```

#1

PyStart #42 Wyjątki, obsługa błędów

Jak wysłać wyjątek?

Jak wyrzucić wyjątek w przypadku gdy podana liczba nie jest podzielna przez 5?

```
1 while True:
2     try:
3         value = int(input('Podaj liczbę: '))
4         if not value % 5 == 0:
5             raise Exception('Liczba nie jest podzielna przez 5.')
6
7         print(value)
8     except Exception as e:
9         print(e)
```

#1

#2

```
1 try:
2     while True:
3         value = int(input('Podaj liczbę: '))
4         if not value % 5 == 0:
5             raise Exception('Liczba nie jest podzielna przez 5.')
6
7         print(value)
8     except Exception as e:
9         print(e)
10
```

PyStart #42 Wyjątki, obsługa błędów

Zadania dla nabrania wprawy

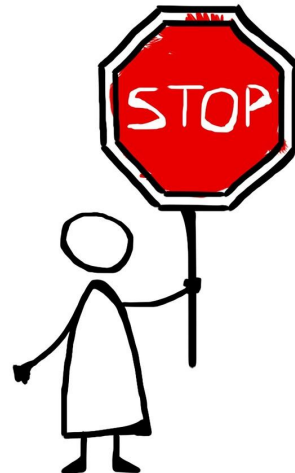
Zapytaj użytkownika o dowolny tekst, który użytkownik chciałby odwrócić
(Wyświetlić od końca do początku).

42.2

Jeśli użytkownik poda pusty string powinien zostać zwrócony wyjątek.



Pytaj użytkownika o nazwy 10 owoców, jeżeli dany owoc znajduje się już na liście to powinien zostać wyrzucony wyjątek i program powinien przestać pytać.



PyStart #42 Wyjątki, obsługa błędów

Ciekawi?



```
1 class ValueToSmall(Exception):  
2     pass  
3  
4 class ValueToBig(Exception):  
5     pass  
6
```

```
8 try:  
9     value = int(input('Podaj liczbę: '))  
10    if value < 10:  
11        raise ValueToSmall()  
12    if value > 20:  
13        raise ValueToBig()  
14  
15    print(value)  
16 except (ValueToSmall, ValueToBig) as error:  
17    print(error)
```

PyStart #42 Wyjątki, obsługa błędów

Podsumowując

- Wyjątki służą do obsługi sytuacji gdy wystąpi błąd.
- Różne typy błędów realizujemy za pomocą dziedziczenia.
- W Pythonie jest wiele “wbudowanych” typów wyjątków, warto zwrócić uwagę na komunikaty.



PyStart #42 Wyjątki, obsługa błędów

Zadania dla nabrania wprawy

Na podstawie API https://restcountries.eu/rest/v2/name/<nazwa_państwa>
np. <https://restcountries.eu/rest/v2/name/Polska> Przygotuj aplikację, która zapyta użytkownika o nazwę Państwa, a następnie wyświetli jego stolicę.

42.3

Pamiętaj by obsłużyć sytuację gdy:

- użytkownik wprowadzi nazwę Państwa, którego nie ma w bazie.
- skrypt nie będzie miał dostępu do internetu

Warto zastanowić się również nad tym co powinno się stać gdy użytkownik zada pytanie o to samo Państwo dwa razy. *Czy powinniśmy pytać o nie drugi raz API? Pomyśl o uzasadnieniu swojej decyzji.*

