

Instrukcje warunkowe

if ... elif ... else

Instrukcje warunkowe pozwalają kierować wykonywanymi instrukcjami pod pewnym warunkiem. Jeżeli kod wpisany po słowie kluczowym `if` wykona się jako wartość prawdziwa (niezerowa, nie-`None`, itp.), wykonany zostanie kod pod tym słowem kluczowym.

```
1 if ... :  
2     print('this is true')
```

Do instrukcji warunkowej można dodać słowo kluczowe `else`. Wtedy, jeżeli wartość przy słowie kluczowym `if` ewaluuje się jako fałsz, wykonany zostanie kod pod słowem kluczowym `else`.

```
1 if ... :  
2     print('this is true')  
3 else:  
4     print('this is false')
```

```
1 if name != 'José Jiménez':  
2     print('this is false')  
3 else:  
4     print('this is true')
```

Możliwe jest także sprawdzenie kilku warunków przed przejściem do `else`. Do sprawdzenia drugiego i kolejnych warunków służy słowo kluczowe `elif` (w wielu innych językach rozwijane jako `else if`).

```
1 if name == 'José Jiménez':  
2     print('My name José Jiménez')  
3 elif name == 'Max Peck':  
4     print('Your name is Max Peck')  
5 else:  
6     print('Your name is neither José Jiménez nor Max Peck')
```

```
1 if not 0 <= k <= n:  
2     raise ValueError('Sample larger than population')
```

not, in, is

Słowo kluczowe `in` pozwala na sprawdzenie czy dana wartość zawiera się w zbiorze (iteratorze).

```

1 if name in ['José Jiménez', 'Max Peck']:
2     print('Your name is José Jiménez or Max Peck')
3 else:
4     print('Your name is neither José Jiménez nor Max Peck')

```

`not` pozwala zanegować warunek.

```

1 if not name == 'José Jiménez':
2     print('Not José')
3 else:
4     print('My name José Jiménez')

```

```

1 if not name:
2     print('Name is not set')
3 else:
4     print('You have set your name')

```

`is` porównuje czy dwa obiekty są tożsame.

```

1 if name is None:
2     print('Name is not set')
3 else:
4     print('You have set your name')

```

Bardzo kuszący jest następujący przykład:

```

1 if name is 'Max Peck':
2     print('You are Max!')
3 else:
4     print('You are not Max!')

```

Nie jest on jednak do końca poprawny. Słowo kluczowe `is` porównuje czy dwa obiekty są tym samym obiektem, nie czy mają taką samą wartość. Poniższy przykład ilustruje, że pomimo że dwa obiekty przechowują takiego samego stringa to nie są sobie tożsame, mimo że są sobie równe.

```

1 a = 'hello'
2 b = 'hello'
3
4 print(f'a is {a}, b is {b}')
5 print(f'a == b returns: {a==b}')
6 print(f'a is b returns: {a is b}')
7
8 a = 'hello'
9 b = ''.join('hello')
10
11 print(f'a is {a}, b is {b}')
12 print(f'a == b returns: {a==b}')
13 print(f'a is b returns: {a is b}')

```

switch statement?

- Why `switch` is bad practise?

" You can do this easily enough with a sequence of `if... elif... elif... else`. There have been some proposals for `switch` statement syntax, but there is no consensus (yet) on whether and how to do range tests. See PEP 275 for complete details and the current status. "

```
1 if name == 'José Jiménez':
2     print('My name José Jiménez')
3 elif name == 'Ivan Ivanovic':
4     print('Your name is Ivan Ivanovic')
5 elif name == 'Max Peck':
6     print('Your name is Max Peck')
7 else:
8     print('Your name is other')
```

```
1 def switch(x):
2     return {
3         'a': 1,
4         'b': 2,
5     }[x]
6
7 switch['a']
```

```
1 choices = {'a': 1, 'b': 2}
2 result = choices.get(key, 'default')
```

Zadania kontrolne

Dzienniczek ucznia

Napisz program, który wczytuje od użytkownika kolejne oceny i:

- sprawdza czy wprowadzona ocena jest na liście dopuszczalnych na wydziale ocen
- jeżeli ocena jest na liście dopuszczalnych na wydziale ocen, dodaje ją do dzienniczka
- jeżeli wpisano cyfrę nie znajdującą się na liście dopuszczalnych ocen, wyświetl informację i zakończ wpisywanie
- wyświetla wyliczoną dla dzienniczka ocen średnią arytmetyczną
- jeżeli wciśnięto sam Enter, oznacza to koniec wpisywania do dzienniczka
- wykorzystaj moduł `statistics` do wyliczania średniej

Warunek

: - Zastosuj akademicką skalę ocen `[2, 3, 3.5, 4, 4.5, 5]`

Podpowiedź

: - dla ułatwienia wszystkie oceny mogą być typu `float` - `len()` `sum()` - `in` - `statistics.mean()`

