

Циклические структуры данных Часть 1

Цикл по словарю

В Python 3 для перебора словаря используется метод `items()`:

```
world = { "afghanistan":30.55,
          "albania":2.77,
          "algeria":39.21 }

for key, value in world.items() :
    print(key + " -- " + str(value))
```

Помните словарь `europe`, который содержал названия некоторых европейских стран в качестве ключей и их столицы как соответствующие значения? Давайте напишем цикл для его итерации!

Инструкции

Напишите цикл `for`, который проходит через каждую пару ключ:значение в `europe`. На каждой итерации должно выводиться столица `X` - это `y`, где `X` - ключ, а `y` - значение пары.

```
# Определение словаря
europe = {'spain':'madrid', 'france':'paris', 'germany':'berlin',
          'norway':'oslo', 'italy':'rome', 'poland':'warsaw',
          'austria':'vienna' }

# Итерация по Europe
for country, capital in europe.items():
    print("Столица " + country + " - это " + capital)
```

```
Столица spain - это madrid
Столица france - это paris
Столица germany - это berlin
Столица norway - это oslo
Столица italy - это rome
Столица poland - это warsaw
Столица austria - это vienna
```

Цикл по массиву NumPy

Если у вас есть одномерный массив `NumPy`, перебор всех элементов может быть таким простым:

```
for x in my_array :  
    ...
```

Если у вас есть двумерный массив NumPy, это более сложно. 2D-массив состоит из нескольких 1D-массивов. Чтобы явно перебрать все отдельные элементы многомерного массива, вам понадобится следующий синтаксис:

```
for x in np.nditer(my_array) :  
    ...
```

Два массива NumPy, которые вы, возможно, узнаете из вводного курса, доступны в вашей сессии Python: `np_height`, массив NumPy, содержащий рост игроков Major League Baseball, и `np_baseball`, 2D-массив NumPy, который содержит и рост (первый столбец), и вес (второй столбец) этих игроков.

Инструкции

- Импортируйте пакет `numpy` под локальным псевдонимом `np`.
- Напишите цикл `for`, который перебирает все элементы в `np_height` и выводит "х дюймов" для каждого элемента, где `x` - значение в массиве.
- Напишите цикл `for`, который посещает каждый элемент массива `np_baseball` и выводит его.

```
import pandas as pd  
import numpy as np  
baseball = pd.read_csv('datasets/MLB(baseball).csv')  
  
np_height = np.array(baseball['Height'])  
np_baseball = np.array(baseball[['Height', 'Weight']])  
  
# Цикл по np_height  
for height in np_height:  
    print(str(height) + " дюймов")  
  
# Цикл по np_baseball  
for x in np.nditer(np_baseball):  
    print(x)
```

```
74 дюймов  
74 дюймов  
72 дюймов  
72 дюймов  
73 дюймов  
69 дюймов  
69 дюймов  
71 дюймов  
76 дюймов  
71 дюймов
```

73 дюймов
73 дюймов
74 дюймов
74 дюймов
69 дюймов
70 дюймов
73 дюймов
75 дюймов
78 дюймов
79 дюймов
76 дюймов
74 дюймов
76 дюймов
72 дюймов
71 дюймов
75 дюймов
77 дюймов
74 дюймов
73 дюймов
74 дюймов
78 дюймов
73 дюймов
75 дюймов
73 дюймов
75 дюймов
75 дюймов
74 дюймов
69 дюймов
71 дюймов
74 дюймов
73 дюймов
73 дюймов
76 дюймов
74 дюймов
74 дюймов
70 дюймов
72 дюймов
77 дюймов
74 дюймов
70 дюймов
73 дюймов
75 дюймов
76 дюймов
76 дюймов
78 дюймов
74 дюймов
74 дюймов
76 дюймов
77 дюймов

81 дюймов
78 дюймов
75 дюймов
77 дюймов
75 дюймов
76 дюймов
74 дюймов
72 дюймов
72 дюймов
75 дюймов
73 дюймов
73 дюймов
73 дюймов
70 дюймов
70 дюймов
70 дюймов
76 дюймов
68 дюймов
71 дюймов
72 дюймов
75 дюймов
75 дюймов
75 дюймов
75 дюймов
68 дюймов
74 дюймов
78 дюймов
71 дюймов
73 дюймов
76 дюймов
74 дюймов
74 дюймов
79 дюймов
75 дюймов
73 дюймов
76 дюймов
74 дюймов
74 дюймов
73 дюймов
72 дюймов
74 дюймов
73 дюймов
74 дюймов
72 дюймов
73 дюймов
69 дюймов
72 дюймов
73 дюймов
75 дюймов

75 дюймов
73 дюймов
72 дюймов
72 дюймов
76 дюймов
74 дюймов
72 дюймов
77 дюймов
74 дюймов
77 дюймов
75 дюймов
76 дюймов
80 дюймов
74 дюймов
74 дюймов
75 дюймов
78 дюймов
73 дюймов
73 дюймов
74 дюймов
75 дюймов
76 дюймов
71 дюймов
73 дюймов
74 дюймов
76 дюймов
76 дюймов
74 дюймов
73 дюймов
74 дюймов
70 дюймов
72 дюймов
73 дюймов
73 дюймов
73 дюймов
73 дюймов
71 дюймов
74 дюймов
74 дюймов
72 дюймов
74 дюймов
71 дюймов
74 дюймов
73 дюймов
75 дюймов
75 дюймов
79 дюймов
73 дюймов
75 дюймов

76 дюймов
74 дюймов
76 дюймов
78 дюймов
74 дюймов
76 дюймов
72 дюймов
74 дюймов
76 дюймов
74 дюймов
75 дюймов
78 дюймов
75 дюймов
72 дюймов
74 дюймов
72 дюймов
74 дюймов
70 дюймов
71 дюймов
70 дюймов
75 дюймов
71 дюймов
71 дюймов
73 дюймов
72 дюймов
71 дюймов
73 дюймов
72 дюймов
75 дюймов
74 дюймов
74 дюймов
75 дюймов
73 дюймов
77 дюймов
73 дюймов
76 дюймов
75 дюймов
74 дюймов
76 дюймов
75 дюймов
73 дюймов
71 дюймов
76 дюймов
75 дюймов
72 дюймов
71 дюймов
77 дюймов
73 дюймов
74 дюймов

71 дюймов
72 дюймов
74 дюймов
75 дюймов
73 дюймов
72 дюймов
75 дюймов
75 дюймов
74 дюймов
72 дюймов
74 дюймов
71 дюймов
70 дюймов
74 дюймов
77 дюймов
77 дюймов
75 дюймов
75 дюймов
78 дюймов
75 дюймов
76 дюймов
73 дюймов
75 дюймов
75 дюймов
79 дюймов
77 дюймов
76 дюймов
71 дюймов
75 дюймов
74 дюймов
69 дюймов
71 дюймов
76 дюймов
72 дюймов
72 дюймов
70 дюймов
72 дюймов
73 дюймов
71 дюймов
72 дюймов
71 дюймов
73 дюймов
72 дюймов
73 дюймов
74 дюймов
74 дюймов
72 дюймов
75 дюймов
74 дюймов

74 дюймов
77 дюймов
75 дюймов
73 дюймов
72 дюймов
71 дюймов
74 дюймов
77 дюймов
75 дюймов
75 дюймов
75 дюймов
78 дюймов
78 дюймов
74 дюймов
76 дюймов
78 дюймов
76 дюймов
70 дюймов
72 дюймов
80 дюймов
74 дюймов
74 дюймов
71 дюймов
70 дюймов
72 дюймов
71 дюймов
74 дюймов
71 дюймов
72 дюймов
71 дюймов
74 дюймов
69 дюймов
76 дюймов
75 дюймов
75 дюймов
76 дюймов
73 дюймов
76 дюймов
73 дюймов
77 дюймов
73 дюймов
72 дюймов
72 дюймов
77 дюймов
77 дюймов
71 дюймов
74 дюймов
74 дюймов
73 дюймов

78 дюймов
75 дюймов
73 дюймов
70 дюймов
74 дюймов
72 дюймов
73 дюймов
73 дюймов
75 дюймов
75 дюймов
74 дюймов
76 дюймов
73 дюймов
74 дюймов
75 дюймов
75 дюймов
72 дюймов
73 дюймов
73 дюймов
72 дюймов
74 дюймов
78 дюймов
76 дюймов
73 дюймов
74 дюймов
75 дюймов
70 дюймов
75 дюймов
71 дюймов
72 дюймов
78 дюймов
75 дюймов
73 дюймов
73 дюймов
71 дюймов
75 дюймов
77 дюймов
72 дюймов
69 дюймов
73 дюймов
74 дюймов
72 дюймов
70 дюймов
75 дюймов
70 дюймов
72 дюймов
72 дюймов
74 дюймов
73 дюймов

74 дюймов
76 дюймов
75 дюймов
80 дюймов
72 дюймов
75 дюймов
73 дюймов
74 дюймов
74 дюймов
73 дюймов
75 дюймов
75 дюймов
71 дюймов
73 дюймов
75 дюймов
74 дюймов
74 дюймов
72 дюймов
74 дюймов
74 дюймов
74 дюймов
73 дюймов
76 дюймов
75 дюймов
72 дюймов
73 дюймов
73 дюймов
73 дюймов
72 дюймов
72 дюймов
72 дюймов
72 дюймов
71 дюймов
75 дюймов
75 дюймов
74 дюймов
73 дюймов
75 дюймов
79 дюймов
74 дюймов
76 дюймов
73 дюймов
74 дюймов
74 дюймов
72 дюймов
74 дюймов
74 дюймов
75 дюймов
78 дюймов

74 дюймов
74 дюймов
74 дюймов
77 дюймов
70 дюймов
73 дюймов
74 дюймов
73 дюймов
71 дюймов
75 дюймов
71 дюймов
72 дюймов
77 дюймов
74 дюймов
70 дюймов
77 дюймов
73 дюймов
72 дюймов
76 дюймов
71 дюймов
76 дюймов
78 дюймов
75 дюймов
73 дюймов
78 дюймов
74 дюймов
79 дюймов
75 дюймов
76 дюймов
72 дюймов
75 дюймов
75 дюймов
70 дюймов
72 дюймов
70 дюймов
74 дюймов
71 дюймов
76 дюймов
73 дюймов
76 дюймов
71 дюймов
69 дюймов
72 дюймов
72 дюймов
69 дюймов
73 дюймов
69 дюймов
73 дюймов
74 дюймов

74 дюймов
72 дюймов
71 дюймов
72 дюймов
72 дюймов
76 дюймов
76 дюймов
76 дюймов
74 дюймов
76 дюймов
75 дюймов
71 дюймов
72 дюймов
71 дюймов
73 дюймов
75 дюймов
76 дюймов
75 дюймов
71 дюймов
75 дюймов
74 дюймов
72 дюймов
73 дюймов
73 дюймов
73 дюймов
73 дюймов
76 дюймов
72 дюймов
76 дюймов
73 дюймов
73 дюймов
73 дюймов
75 дюймов
75 дюймов
77 дюймов
73 дюймов
72 дюймов
75 дюймов
70 дюймов
74 дюймов
72 дюймов
80 дюймов
71 дюймов
71 дюймов
74 дюймов
74 дюймов
73 дюймов
75 дюймов
76 дюймов
73 дюймов

77 дюймов
72 дюймов
73 дюймов
77 дюймов
76 дюймов
71 дюймов
75 дюймов
73 дюймов
74 дюймов
77 дюймов
71 дюймов
72 дюймов
73 дюймов
69 дюймов
73 дюймов
70 дюймов
74 дюймов
76 дюймов
73 дюймов
73 дюймов
75 дюймов
73 дюймов
79 дюймов
74 дюймов
73 дюймов
74 дюймов
77 дюймов
75 дюймов
74 дюймов
73 дюймов
77 дюймов
73 дюймов
77 дюймов
74 дюймов
74 дюймов
73 дюймов
77 дюймов
74 дюймов
77 дюймов
75 дюймов
77 дюймов
75 дюймов
71 дюймов
74 дюймов
70 дюймов
79 дюймов
72 дюймов
72 дюймов
70 дюймов

74 дюймов
74 дюймов
72 дюймов
73 дюймов
72 дюймов
74 дюймов
74 дюймов
76 дюймов
82 дюймов
74 дюймов
74 дюймов
70 дюймов
73 дюймов
73 дюймов
74 дюймов
77 дюймов
72 дюймов
76 дюймов
73 дюймов
73 дюймов
72 дюймов
74 дюймов
74 дюймов
71 дюймов
72 дюймов
75 дюймов
74 дюймов
74 дюймов
77 дюймов
70 дюймов
71 дюймов
73 дюймов
76 дюймов
71 дюймов
75 дюймов
74 дюймов
72 дюймов
76 дюймов
79 дюймов
76 дюймов
73 дюймов
76 дюймов
78 дюймов
75 дюймов
76 дюймов
72 дюймов
72 дюймов
73 дюймов
73 дюймов

75 дюймов
71 дюймов
76 дюймов
70 дюймов
75 дюймов
74 дюймов
75 дюймов
73 дюймов
71 дюймов
71 дюймов
72 дюймов
73 дюймов
73 дюймов
72 дюймов
69 дюймов
73 дюймов
78 дюймов
71 дюймов
73 дюймов
75 дюймов
76 дюймов
70 дюймов
74 дюймов
77 дюймов
75 дюймов
79 дюймов
72 дюймов
77 дюймов
73 дюймов
75 дюймов
75 дюймов
75 дюймов
73 дюймов
73 дюймов
76 дюймов
77 дюймов
75 дюймов
70 дюймов
71 дюймов
71 дюймов
75 дюймов
74 дюймов
69 дюймов
70 дюймов
75 дюймов
72 дюймов
75 дюймов
73 дюймов
72 дюймов

72 дюймов
72 дюймов
76 дюймов
75 дюймов
74 дюймов
69 дюймов
73 дюймов
72 дюймов
72 дюймов
75 дюймов
77 дюймов
76 дюймов
80 дюймов
77 дюймов
76 дюймов
79 дюймов
71 дюймов
75 дюймов
73 дюймов
76 дюймов
77 дюймов
73 дюймов
76 дюймов
70 дюймов
75 дюймов
73 дюймов
75 дюймов
70 дюймов
69 дюймов
71 дюймов
72 дюймов
72 дюймов
73 дюймов
70 дюймов
70 дюймов
73 дюймов
76 дюймов
75 дюймов
72 дюймов
73 дюймов
79 дюймов
71 дюймов
72 дюймов
74 дюймов
74 дюймов
74 дюймов
72 дюймов
76 дюймов
76 дюймов

72 дюймов
72 дюймов
71 дюймов
72 дюймов
72 дюймов
70 дюймов
77 дюймов
74 дюймов
72 дюймов
76 дюймов
71 дюймов
76 дюймов
71 дюймов
73 дюймов
70 дюймов
73 дюймов
73 дюймов
72 дюймов
71 дюймов
71 дюймов
71 дюймов
72 дюймов
72 дюймов
74 дюймов
74 дюймов
74 дюймов
71 дюймов
72 дюймов
75 дюймов
72 дюймов
71 дюймов
72 дюймов
72 дюймов
72 дюймов
72 дюймов
74 дюймов
74 дюймов
77 дюймов
75 дюймов
73 дюймов
75 дюймов
73 дюймов
76 дюймов
72 дюймов
77 дюймов
75 дюймов
72 дюймов
71 дюймов
71 дюймов

75 дюймов
72 дюймов
73 дюймов
73 дюймов
71 дюймов
70 дюймов
75 дюймов
71 дюймов
76 дюймов
73 дюймов
68 дюймов
71 дюймов
72 дюймов
74 дюймов
77 дюймов
72 дюймов
76 дюймов
78 дюймов
81 дюймов
72 дюймов
73 дюймов
76 дюймов
72 дюймов
72 дюймов
74 дюймов
76 дюймов
73 дюймов
76 дюймов
75 дюймов
70 дюймов
71 дюймов
74 дюймов
72 дюймов
73 дюймов
76 дюймов
76 дюймов
73 дюймов
71 дюймов
68 дюймов
71 дюймов
71 дюймов
74 дюймов
77 дюймов
69 дюймов
72 дюймов
76 дюймов
75 дюймов
76 дюймов
75 дюймов

76 дюймов
72 дюймов
74 дюймов
76 дюймов
74 дюймов
72 дюймов
75 дюймов
78 дюймов
77 дюймов
70 дюймов
72 дюймов
79 дюймов
74 дюймов
71 дюймов
68 дюймов
77 дюймов
75 дюймов
71 дюймов
72 дюймов
70 дюймов
72 дюймов
72 дюймов
73 дюймов
72 дюймов
74 дюймов
72 дюймов
72 дюймов
75 дюймов
72 дюймов
73 дюймов
74 дюймов
72 дюймов
78 дюймов
75 дюймов
72 дюймов
74 дюймов
75 дюймов
75 дюймов
76 дюймов
74 дюймов
74 дюймов
73 дюймов
74 дюймов
71 дюймов
74 дюймов
75 дюймов
76 дюймов
74 дюймов
76 дюймов

76 дюймов
73 дюймов
75 дюймов
75 дюймов
74 дюймов
68 дюймов
72 дюймов
75 дюймов
71 дюймов
70 дюймов
72 дюймов
73 дюймов
72 дюймов
75 дюймов
74 дюймов
70 дюймов
76 дюймов
71 дюймов
82 дюймов
72 дюймов
73 дюймов
74 дюймов
71 дюймов
75 дюймов
77 дюймов
72 дюймов
74 дюймов
72 дюймов
73 дюймов
78 дюймов
77 дюймов
73 дюймов
73 дюймов
73 дюймов
73 дюймов
73 дюймов
76 дюймов
75 дюймов
70 дюймов
73 дюймов
72 дюймов
73 дюймов
75 дюймов
74 дюймов
73 дюймов
73 дюймов
76 дюймов
73 дюймов
75 дюймов

70 дюймов
77 дюймов
72 дюймов
77 дюймов
74 дюймов
75 дюймов
75 дюймов
75 дюймов
75 дюймов
72 дюймов
74 дюймов
71 дюймов
76 дюймов
71 дюймов
75 дюймов
76 дюймов
83 дюймов
75 дюймов
74 дюймов
76 дюймов
72 дюймов
72 дюймов
75 дюймов
75 дюймов
72 дюймов
77 дюймов
73 дюймов
72 дюймов
70 дюймов
74 дюймов
72 дюймов
74 дюймов
72 дюймов
71 дюймов
70 дюймов
71 дюймов
76 дюймов
74 дюймов
76 дюймов
74 дюймов
74 дюймов
74 дюймов
74 дюймов
75 дюймов
75 дюймов
71 дюймов
71 дюймов
74 дюймов
77 дюймов
71 дюймов

74 дюймов
75 дюймов
77 дюймов
76 дюймов
74 дюймов
76 дюймов
72 дюймов
71 дюймов
72 дюймов
75 дюймов
73 дюймов
68 дюймов
72 дюймов
69 дюймов
73 дюймов
73 дюймов
75 дюймов
70 дюймов
70 дюймов
74 дюймов
75 дюймов
74 дюймов
74 дюймов
73 дюймов
74 дюймов
75 дюймов
77 дюймов
73 дюймов
74 дюймов
76 дюймов
74 дюймов
75 дюймов
73 дюймов
76 дюймов
78 дюймов
75 дюймов
73 дюймов
77 дюймов
74 дюймов
72 дюймов
74 дюймов
72 дюймов
71 дюймов
73 дюймов
75 дюймов
73 дюймов
67 дюймов
67 дюймов
76 дюймов

74 дюймов
73 дюймов
70 дюймов
75 дюймов
70 дюймов
72 дюймов
77 дюймов
79 дюймов
78 дюймов
74 дюймов
75 дюймов
75 дюймов
78 дюймов
76 дюймов
75 дюймов
69 дюймов
75 дюймов
72 дюймов
75 дюймов
73 дюймов
74 дюймов
75 дюймов
75 дюймов
73 дюймов
74
74
72
72
73
69
69
71
76
71
73
73
74
74
69
70
73
75
78
79
76
74
76
72
71
75

77
74
73
74
78
73
75
73
75
75
74
69
71
74
73
73
76
74
74
70
72
77
74
70
73
75
76
76
78
74
74
76
77
81
78
75
77
75
76
74
72
72
75
73
73
73
70
70
70

76
68
71
72
75
75
75
75
68
74
78
71
73
76
74
74
79
75
73
76
74
74
73
72
74
73
74
72
73
69
72
73
75
75
73
72
72
76
74
72
77
74
77
75
76
80
74
74
75

78
73
73
74
75
76
71
73
74
76
76
74
73
74
70
72
73
73
73
73
71
74
74
72
74
71
74
73
75
75
79
73
75
76
74
76
78
74
76
72
74
76
74
75
78
75
72
74
72

74
70
71
70
75
71
71
73
72
71
73
72
75
74
74
75
73
77
73
76
75
74
76
75
73
71
76
75
72
71
77
73
74
71
72
74
75
73
72
75
75
74
72
74
71
70
74
77
77

75
75
78
75
76
73
75
75
79
77
76
71
75
74
69
71
76
72
72
72
70
72
73
71
72
71
73
72
73
74
74
72
75
74
74
77
75
73
72
71
74
77
75
75
75
78
78
74
76
78

76
70
72
80
74
74
71
70
72
71
74
71
72
71
74
69
76
75
75
76
73
76
73
77
73
72
72
77
77
71
74
74
73
78
75
73
70
74
72
73
73
75
75
74
76
73
74
75
75

72
73
73
72
74
78
76
73
74
75
70
75
71
72
78
75
73
73
71
75
77
72
69
73
74
72
70
75
70
72
72
74
73
74
76
75
80
72
75
73
74
74
73
75
75
71
73
75
74

74
72
74
74
74
73
76
75
72
73
73
73
72
72
72
72
71
75
75
74
73
75
79
74
76
73
74
74
72
74
74
75
78
74
74
74
77
70
73
74
73
71
75
71
72
77
74
70
77

73
72
76
71
76
78
75
73
78
74
79
75
76
72
75
75
70
72
70
74
71
76
73
76
71
69
72
72
69
73
69
73
74
74
72
71
72
72
76
76
76
74
76
75
71
72
71
73
75

76
75
71
75
74
72
73
73
73
73
76
72
76
73
73
73
75
75
75
77
73
72
75
70
74
72
80
71
71
74
74
73
75
76
73
77
72
73
77
76
71
75
73
74
77
71
72
73
69
73

70
74
76
73
73
75
73
79
74
73
74
77
75
74
73
77
73
77
74
74
73
77
74
77
75
77
75
71
74
70
79
72
72
70
74
74
72
73
72
74
74
76
82
74
74
70
73
73
74
77

72
76
73
73
72
74
74
71
72
75
74
74
77
70
71
73
76
71
75
74
72
76
79
76
73
76
78
75
76
72
72
73
73
75
71
76
70
75
74
75
73
71
71
72
73
73
72
69
73

78
71
73
75
76
70
74
77
75
79
72
77
73
75
75
75
73
73
76
77
75
70
71
71
75
74
69
70
75
72
75
73
72
72
72
72
76
75
74
69
73
72
72
75
77
76
80
77
76
79

71
75
73
76
77
73
76
70
75
73
75
70
69
71
72
72
73
70
70
73
76
75
72
73
79
71
72
74
74
74
72
76
76
72
72
71
72
72
70
77
74
72
76
71
76
71
73
70
73

73
72
71
71
71
72
72
74
74
74
71
72
75
72
71
72
72
72
72
74
74
77
75
73
75
73
76
72
77
75
72
71
71
75
72
73
73
71
70
75
71
76
73
68
71
72
74
77
72

76
78
81
72
73
76
72
72
74
76
73
76
75
70
71
74
72
73
76
76
73
71
68
71
71
74
77
69
72
76
75
76
75
76
72
74
76
74
72
75
78
77
70
72
79
74
71
68
77

75
71
72
70
72
72
73
72
74
72
72
75
72
73
74
72
78
75
72
74
75
75
76
74
74
73
74
71
74
75
76
74
76
76
73
75
75
74
68
72
75
71
70
72
73
72
75
74
70

76
71
82
72
73
74
71
75
77
72
74
72
73
78
77
73
73
73
73
73
76
75
70
73
72
73
75
74
73
73
76
73
75
70
77
72
77
74
75
75
75
75
72
74
71
76
71
75
76

83
75
74
76
72
72
75
75
72
77
73
72
70
74
72
74
72
71
70
71
76
74
76
74
74
74
75
75
71
71
74
77
71
74
75
77
76
74
76
72
71
72
75
73
68
72
69
73
73

75
70
70
74
75
74
74
73
74
75
77
73
74
76
74
75
73
76
78
75
73
77
74
72
74
72
71
73
75
73
67
67
76
74
73
70
75
70
72
77
79
78
74
75
75
78
76
75
69

75
72
75
73
74
75
75
73
180
215
210
210
188
176
209
200
231
180
188
180
185
160
180
185
189
185
219
230
205
230
195
180
192
225
203
195
182
188
200
180
200
200
245
240
215
185
175
199
200

215
200
205
206
186
188
220
210
195
200
200
212
224
210
205
220
195
200
260
228
270
200
210
190
220
180
205
210
220
211
200
180
190
170
230
155
185
185
200
225
225
220
160
205
235
250
210
190
160
200

205
222
195
205
220
220
170
185
195
220
230
180
220
180
180
170
210
215
200
213
180
192
235
185
235
210
222
210
230
220
180
190
200
210
194
180
190
240
200
198
200
195
210
220
190
210
225
180
185

170
185
185
180
178
175
200
204
211
190
210
190
190
185
290
175
185
200
220
170
220
190
220
205
200
250
225
215
210
215
195
200
194
220
180
180
170
195
180
170
206
205
200
225
201
225
233
180
225

180
220
180
237
215
190
235
190
180
165
195
200
190
190
185
185
205
190
205
206
220
208
170
195
210
190
211
230
170
185
185
241
225
210
175
230
200
215
198
226
278
215
230
240
184
219
170
218
190

225
220
176
190
197
204
167
180
195
220
215
185
190
205
205
200
210
215
200
205
211
190
208
200
210
232
230
210
220
210
202
212
225
170
190
200
237
220
170
193
190
150
220
200
190
185
185
200
172

220
225
190
195
219
190
197
200
195
210
177
220
235
180
195
195
190
230
190
200
190
190
200
200
184
200
180
219
187
200
220
205
190
170
160
215
175
205
200
214
200
190
180
205
220
190
215
235
191

200
181
200
210
240
185
165
190
185
175
155
210
170
175
220
210
205
200
205
195
240
150
200
215
202
200
190
205
190
160
215
185
200
190
210
185
220
190
202
205
220
175
160
190
200
229
206
220
180

195
175
188
230
190
200
190
219
235
180
180
180
200
234
185
220
223
200
210
200
210
190
177
227
180
195
199
175
185
240
210
180
194
225
180
205
193
230
230
220
200
249
190
208
245
250
160
192
220

170
197
155
190
200
220
210
228
190
160
184
180
180
200
176
160
222
211
195
200
175
206
240
185
260
185
221
205
200
170
201
205
185
205
245
220
210
220
185
175
170
180
200
210
175
220
206
180
210

195
200
200
164
180
220
195
205
170
240
210
195
200
205
192
190
170
240
200
205
175
250
220
224
210
195
180
245
175
180
215
175
180
195
230
230
205
215
195
180
205
180
190
180
190
190
220
210
255

190
230
200
205
210
225
215
220
205
200
220
197
225
187
245
185
185
175
200
180
188
225
200
210
245
213
231
165
228
210
250
191
190
200
215
254
232
180
215
220
180
200
170
195
210
200
220
165
180

200
200
170
224
220
180
198
240
239
185
210
220
200
195
220
230
170
220
230
165
205
192
210
205
200
210
185
195
202
205
195
180
200
185
240
185
220
205
205
180
201
190
208
240
180
230
195
215
190
195

215
215
220
220
230
195
190
195
209
204
170
185
205
175
210
190
180
180
160
235
200
210
180
190
197
203
205
170
200
250
200
220
200
190
170
190
220
215
206
215
185
235
188
230
195
168
190
160
200

200
189
180
190
200
220
187
240
190
180
185
210
220
219
190
193
175
180
215
210
200
190
185
220
170
195
205
195
210
190
190
180
220
190
186
185
190
180
190
170
210
240
220
180
210
210
195
160
180

205
200
185
245
190
210
200
200
222
215
240
170
220
156
190
202
221
200
190
210
190
200
165
190
185
230
208
209
175
180
200
205
200
250
210
230
244
202
240
200
215
177
210
170
215
217
198
200
220

170
200
230
231
183
192
167
190
180
180
215
160
205
223
175
170
190
240
175
230
223
196
167
195
190
250
190
190
190
170
160
150
225
220
209
210
176
260
195
190
184
180
195
195
219
225
212
202
185

200
209
200
195
228
210
190
212
190
218
220
190
235
210
200
188
210
235
188
215
216
220
180
185
200
210
220
185
231
210
195
200
205
200
190
250
185
180
170
180
208
235
215
244
220
185
230
190
200

180
190
196
180
230
224
160
178
205
185
210
180
190
200
257
190
220
165
205
200
208
185
215
170
235
210
170
180
170
190
150
230
203
260
246
186
210
198
210
215
180
200
245
200
192
192
200
192
205

190
186
170
197
219
200
220
207
225
207
212
225
170
190
210
230
210
200
238
234
222
200
190
170
220
223
210
215
196
175
175
189
205
210
180
180
197
220
228
190
204
165
216
220
208
210
215
195
200

```
215
229
240
207
205
208
185
190
170
208
225
190
225
185
180
165
240
220
212
163
215
175
205
210
205
208
215
180
200
230
211
230
190
220
180
205
190
180
205
190
195
```

Цикл по DataFrame (1)

Итерация по объекту Pandas DataFrame обычно выполняется с помощью метода `iterrows()`. Используя его в цикле `for`, каждое наблюдение перебирается, и на каждой итерации метка строки и собственно содержимое этой строки становятся доступными:


```
for lab, row in brics.iterrows() :  
    ...
```

В этом и последующих упражнениях вы будете работать с DataFrame под названием `cars`. Он содержит информацию о количестве автомобилей на человека и о том, едут ли люди направо или налево, для семи стран мира.

Инструкции

Напишите цикл `for`, который перебирает строки `cars` и на каждой итерации выполняет два вызова `print()`: один для вывода метки строки и второй для вывода всех содержимого строки.

```
cars = pd.read_csv('datasets/cars.csv', index_col = 0)  
  
# Итерация по строкам автомобилей  
for lab, row in cars.iterrows():  
    print(lab)  
    print(row)
```

```
US  
cars_per_cap      809  
country           United States  
drives_right      True  
Name: US, dtype: object  
AUS  
cars_per_cap      731  
country           Australia  
drives_right      False  
Name: AUS, dtype: object  
JAP  
cars_per_cap      588  
country           Japan  
drives_right      False  
Name: JAP, dtype: object  
IN  
cars_per_cap      18  
country           India  
drives_right      False  
Name: IN, dtype: object  
RU  
cars_per_cap      200  
country           Russia  
drives_right      True  
Name: RU, dtype: object  
MOR  
cars_per_cap      70  
country           Morocco
```

```
drives_right      True
Name: MOR, dtype: object
EG
cars_per_cap      45
country           Egypt
drives_right      True
Name: EG, dtype: object
```

Цикл по DataFrame (2)

Данные строки, которые генерируются при каждом выполнении `iterrows()`, представляют собой объект Pandas Series. Этот формат не очень удобен для вывода. К счастью, вы можете легко выбирать переменные из объекта Pandas Series, используя квадратные скобки:

```
for lab, row in brics.iterrows() :
    print(row['country'])
```

Инструкции

- Используя итераторы `lab` и `row`, измените код в цикле так, чтобы при первой итерации выводилось `US: 809`, при второй `AUS: 731` и так далее.
- Вывод должен иметь форму `"country: cars_per_cap"`. Убедитесь, что выводится именно эта строка (с правильным пробелом).
- Вы можете использовать функцию `str()`, чтобы преобразовать целочисленные данные в строку и вывести их вместе с меткой страны.

Адаптировать цикл

```
for lab, row in cars.iterrows() :
    # Напечатать (lab)
    print(lab + ": " + str(row['cars_per_cap']))
```

```
US: 809
AUS: 731
JAP: 588
IN: 18
RU: 200
MOR: 70
EG: 45
```

Добавление столбца (1)

Добавить `length` (длину) названий стран из DataFrame `brics` в новый столбец:

```
for lab, row in brics.iterrows() :
    brics.loc[lab, "name_length"] = len(row["country"])
```

Вы можете сделать нечто подобное с DataFrame `cars`.

Инструкции

- Используйте цикл `for`, чтобы добавить новый столбец с именем `COUNTRY`, который содержит верхний регистр названий стран из столбца `"country"`. Для этого можно использовать метод `upper()` для строк.
- Чтобы убедиться, что ваш код работает, выведите `cars`.
- Не делайте отступ для этого кода, чтобы он не был частью цикла `for`.

```
# Код цикла, добавляющего столбец COUNTRY
for lab, row in cars.iterrows():
    cars.loc[lab, "COUNTRY"] = (row["country"]).upper()

# Вывод cars
print(cars)
```

	<code>cars_per_cap</code>	<code>country</code>	<code>drives_right</code>	<code>COUNTRY</code>
US	809	United States	True	UNITED STATES
AUS	731	Australia	False	AUSTRALIA
JAP	588	Japan	False	JAPAN
IN	18	India	False	INDIA
RU	200	Russia	True	RUSSIA
MOR	70	Morocco	True	MOROCCO
EG	45	Egypt	True	EGYPT

Добавление столбца (2)

Использование `iterrows()` для итерации по каждому наблюдению объекта Pandas DataFrame легко понять, но не очень эффективно. На каждой итерации создается новый объект Pandas Series.

Если вы хотите добавить столбец в DataFrame, вызывая функцию для другого столбца, метод `iterrows()` в сочетании с циклом `for` не является предпочтительным способом. Вместо этого вам следует использовать `apply()`.

Сравните версию с `iterrows()` с версией с `apply()`, чтобы получить тот же результат в DataFrame `brics`:

```
for lab, row in brics.iterrows() :
    brics.loc[lab, "name_length"] = len(row["country"])

brics["name_length"] = brics["country"].apply(len)
```

Мы можем сделать нечто подобное для вызова метода `upper()` для каждого имени в столбце `country`. Однако `upper()` - это метод, поэтому нам понадобится немного другой подход:

Инструкции

- Замените цикл `for` однострочным кодом, использующим `.apply(str.upper)`. Вызов должен дать тот же результат: в `cars` должен быть добавлен столбец `COUNTRY`, содержащий верхний регистр названий стран.
- Как обычно, выведите `cars`, чтобы увидеть результат вашего труда.

```
# Использование .apply(str.upper)
# for lab, row in cars.iterrows() :
#     cars.loc[lab, "COUNTRY"] = row["country"].upper()

cars["COUNTRY"] = cars["country"].apply(lambda x: x.upper())
print(cars)
```

	cars_per_cap	country	drives_right	COUNTRY
US	809	United States	True	UNITED STATES
AUS	731	Australia	False	AUSTRALIA
JAP	588	Japan	False	JAPAN
IN	18	India	False	INDIA
RU	200	Russia	True	RUSSIA
MOR	70	Morocco	True	MOROCCO
EG	45	Egypt	True	EGYPT