# Словари и pandas

## Мотивация для словарей

Чтобы понять, почему словари полезны, взгляните на два списка, определенные в скрипте. страны содержит названия некоторых европейских «стран». capitals перечисляет соответствующие названия их столицы.

### Инструкции

* Используйте метод \*\*\*\*index()\*\*\*\* для стран, чтобы найти индекс Германии. Сохраните этот индекс как ind\_ger.
* Используйте ind\_ger для доступа к «столице» Германии из списка столиц. Распечатай.

# Определение стран и их столиц  
countries = ['spain', 'france', 'germany', 'norway']  
capitals = ['madrid', 'paris', 'berlin', 'oslo']  
  
# Получение индекса 'germany': ind\_ger  
  
# Использование ind\_ger для вывода столицы Германии

berlin

## Доступ к словарю

Если ключи словаря выбраны разумно, доступ к значениям в словаре будет простым и интуитивно понятным. Например, чтобы получить столицу Франции из Европы, вы можете использовать:

Европа['Франция']

Здесь france — это ключ, а paris — возвращаемое значение.

### Инструкции

Со строками в countries и capitals создайте словарь под названием europe с четырьмя парами ключ:значение. Остерегайтесь капитализации! Убедитесь, что вы везде используете строчные буквы. Распечатайте слово «Европа» и проверьте, соответствует ли результат вашим ожиданиям.

* Проверьте, какие ключи есть в europe, вызвав метод keys(). Выведите результат.
* Выведите значение, которое принадлежит ключу 'italy'.

# Словарь словарей  
europe = { 'spain': { 'capital':'madrid', 'population':46.77 },  
 'france': { 'capital':'paris', 'population':66.03 },  
 'germany': { 'capital':'berlin', 'population':80.62 },  
 'norway': { 'capital':'oslo', 'population':5.084 } }  
  
# Вывод столицы Франции  
  
# Создание подсловаря data  
data = {'capital':'rome','population':59.83}  
  
# Добавление данных в europe под ключ 'italy'  
  
# Вывод europe

{'spain': {'capital': 'madrid', 'population': 46.77}, 'france': {'capital': 'paris', 'population': 66.03}, 'germany': {'capital': 'berlin', 'population': 80.62}, 'norway': {'capital': 'oslo', 'population': 5.084}, 'italy': {'capital': 'rome', 'population': 59.83}}

## Управление словарями (1)

Если вы знаете, как получить доступ к словарю, вы также можете присвоить ему новое значение. Чтобы добавить новую пару ключ-значение в europe, вы можете использовать что-то вроде этого:

europe['iceland'] = 'reykjavik'

### Инструкции

* Добавьте ключ 'italy' со значением 'rome' в словарь europe.
* Чтобы убедиться, что 'italy' теперь является ключом в europe, выведите 'italy' из europe.
* Добавьте еще одну пару ключ-значение в europe: ключ 'poland', значение 'warsaw'.
* Выведите europe.

# Определение словаря  
europe = {'spain':'madrid', 'france':'paris', 'germany':'berlin', 'norway':'oslo' }  
  
# Добавление Италии в Европу  
  
  
# Вывод Италии в Европе  
  
  
# Добавление Польши в Европу  
  
  
# Вывод Европы

True  
{'spain': 'madrid', 'france': 'paris', 'germany': 'berlin', 'norway': 'oslo', 'italy': 'rome', 'poland': 'warsaw'}

# Заранее определенные списки  
names = ['Соединенные Штаты', 'Австралия', 'Япония', 'Индия', 'Россия', 'Марокко', 'Египет']  
dr = [True, False, False, False, True, True, True]  
cpc = [809, 731, 588, 18, 200, 70, 45]  
  
# Импорт pandas как pd  
import pandas as pd  
  
# Создание словаря my\_dict с тремя парами ключ:значение: my\_dict  
my\_dict = {'country': names, 'drives\_right': dr, 'cars\_per\_cap': cpc}  
  
# Создание DataFrame cars из my\_dict: cars  
cars = pd.DataFrame(my\_dict)  
  
# Вывод cars  
print(cars)

country drives\_right cars\_per\_cap  
0 Соединенные Штаты True 809  
1 Австралия False 731  
2 Япония False 588  
3 Индия False 18  
4 Россия True 200  
5 Марокко True 70  
6 Египет True 45

## Изменение словаря (2)

* Кто-то решил пошутить и изменил ваш точно сгенерированный словарь. Адаптированная версия словаря europe доступна в скрипте.

Можете его очистить? Не меняйте определение europe, а просто добавьте команды Python в скрипт для обновления и удаления пар ключ:значение.

### Инструкции

* Столица Германии не 'Bonn'; это 'Berlin'. Обновите его значение.
* Австралия не входит в Европу, Австрия! Удалите ключ 'australia' из europe.
* Напечатайте europe, чтобы увидеть, сработала ли ваша очистка.

# Определение словаря  
europe = {'spain':'madrid', 'france':'paris', 'germany':'bonn',  
 'norway':'oslo', 'italy':'rome', 'poland':'warsaw',  
 'australia':'vienna' }  
  
# Обновление столицы Германии  
  
# Удаление Австралии  
  
# Вывод словаря europe

{'spain': 'madrid', 'france': 'paris', 'germany': 'berlin', 'norway': 'oslo', 'italy': 'rome', 'poland': 'warsaw'}

## Словари в словарях

Помните списки? Они могли содержать всё, даже другие списки. Для словарей то же самое. Словари могут содержать пары ключ:значение, где значения снова являются словарями.

Для примера взгляните на скрипт, в котором создана другая версия europe - словаря, с которым вы работали. Ключи все еще - это названия стран, но значения представляют собой словари, которые содержат больше информации, чем просто столицы.

Совершенно возможно цеплять квадратные скобки для выбора элементов. Чтобы получить население Испании из europe, например, вам нужно:

europe['spain']['population']

### Инструкции

* Используйте цепочку квадратных скобок, чтобы выбрать и вывести столицу Франции.
* Создайте словарь с именем data, с ключами 'capital' и 'population'. Установите их соответственно 'rome' и 59.83.
* Добавьте новую пару ключ-значение в europe; ключ - 'italy', а значение - data, словарь, который вы только что создали.

# Словарь из словарей  
europe = { 'spain': { 'capital':'madrid', 'population':46.77 },  
 'france': { 'capital':'paris', 'population':66.03 },  
 'germany': { 'capital':'berlin', 'population':80.62 },  
 'norway': { 'capital':'oslo', 'population':5.084 } }  
  
# Вывести столицу Франции  
  
  
# Создать подсловарь data  
  
# Добавить data в словарь europe под ключом 'italy'  
  
# Вывести europe

{'spain': {'capital': 'madrid', 'population': 46.77}, 'france': {'capital': 'paris', 'population': 66.03}, 'germany': {'capital': 'berlin', 'population': 80.62}, 'norway': {'capital': 'oslo', 'population': 5.084}, 'italy': {'capital': 'rome', 'population': 59.83}}

# Pandas

## Словарь в DataFrame (1)

Pandas - это библиотека с открытым исходным кодом, предоставляющая высокопроизводительные и простые в использовании структуры данных и инструменты для анализа данных на Python. Звучит многообещающе!

DataFrame - одна из самых важных структур данных в Pandas. Это, по сути, способ хранить табличные данные, где вы можете помечать строки и столбцы. Один из способов создания DataFrame - из словаря.

В упражнениях, которые следуют, вы будете работать с данными о транспортных средствах из разных стран. Каждое наблюдение соответствует стране, а столбцы содержат информацию о количестве транспортных средств на душу населения, о том, ездят ли люди слева или справа и так далее.

Три списка определены в скрипте:

* names, содержащий имена стран, для которых доступны данные.
* dr, список с логическими значениями, определяющими, едут ли люди слева или справа в соответствующей стране.
* cpc, количество автотранспортных средств на 1000 человек в соответствующей стране.

Каждый ключ словаря - это метка столбца, а каждое значение - список, содержащий элементы столбца.

### Инструкции

Импортировать pandas как pd

Использовать предопределенные списки для создания словаря под названием my\_dict. Должно быть три пары ключ-значение:

* ключ 'country' и значение names.
* ключ drives\_right и значение dr.
* ключ cars\_per\_cap и значение cpc.
* my\_dict = {'country': names, 'drives\_right': dr, 'cars\_per\_cap': cpc}

Использовать pd.DataFrame(), чтобы преобразовать ваш словарь в DataFrame под названием cars.

cars = pd.DataFrame(my\_dict)

Напечатать cars и посмотреть, какой он красивый.

print(cars)

# Предопределенные списки  
names = ['Соединенные Штаты', 'Австралия', 'Япония', 'Индия', 'Россия', 'Марокко', 'Египет']  
dr = [True, False, False, False, True, True, True]  
cpc = [809, 731, 588, 18, 200, 70, 45]  
  
# Импорт pandas как pd  
import pandas as pd  
  
# Создать словарь my\_dict с тремя парами ключ:значение: my\_dict  
  
# Создать DataFrame cars из my\_dict: cars  
  
# Напечатать cars

country drives\_right cars\_per\_cap  
0 Соединенные Штаты True 809  
1 Австралия False 731  
2 Япония False 588  
3 Индия False 18  
4 Россия True 200  
5 Марокко True 70  
6 Египет True 45

## Словарь в DataFrame (2)

В скрипте уже представлен код на Python, который решает предыдущее упражнение. Заметили ли вы, что метки строк (т.е. метки для различных наблюдений) автоматически установлены как целые числа от 0 до 6?

Для решения этой проблемы был создан список row\_labels. Вы можете использовать его для указания меток строк в DataFrame cars. Вы это делаете, устанавливая атрибут index для cars, к которому можно обратиться как cars.index.

### Инструкции

* Укажите метки строк, установив cars.index равным row\_labels.
* Снова выведите cars на печать и проверьте, правильно ли установлены метки строк.

# Создать список меток строк  
row\_labels = ['US', 'AUS', 'JAP', 'IN', 'RU', 'MOR', 'EG']  
  
# Установить метки строк для cars  
  
# Напечатать cars

country drives\_right cars\_per\_cap  
US Соединенные Штаты True 809  
AUS Австралия False 731  
JAP Япония False 588  
IN Индия False 18  
RU Россия True 200  
MOR Марокко True 70  
EG Египет True 45

## CSV в DataFrame

Перенос данных в словарь, а затем построение DataFrame работает, но это не очень эффективно. Что если у вас миллионы наблюдений? В таких случаях данные обычно доступны в файлах с регулярной структурой. Один из таких типов файлов - это CSV-файл, что означает "значения, разделенные запятыми".

Чтобы импортировать CSV-данные в Python как DataFrame Pandas, можно использовать read\_csv().

Давайте изучим эту функцию с теми же данными об автомобилях из предыдущих упражнений. На этот раз, однако, данные доступны в CSV-файле под названием cars.csv. Он доступен в вашей текущей рабочей директории, поэтому путь к файлу просто cars.csv.

### Инструкции

* Для импорта файлов CSV вам все еще нужен пакет pandas: импортируйте его как pd.
* Используйте pd.read\_csv(), чтобы импортировать данные из cars.csv в виде DataFrame. Сохраните этот DataFrame как cars.
* Выведите cars на экран. Все выглядит нормально?

# Импортировать pandas как pd  
  
# Импортировать данные из cars.csv: cars  
cars = pd.DataFrame(pd.read\_csv('datasets/cars.csv'))  
# Вывести cars

Unnamed: 0 cars\_per\_cap country drives\_right  
0 US 809 United States True  
1 AUS 731 Australia False  
2 JAP 588 Japan False  
3 IN 18 India False  
4 RU 200 Russia True  
5 MOR 70 Morocco True  
6 EG 45 Egypt True

## CSV в DataFrame (2)

Ваш вызов read\_csv() для импорта данных CSV не вызвал ошибку, но вывод не совсем соответствует нашим ожиданиям. Метки строк были импортированы как еще один столбец без имени.

Помните о index\_col, аргументе read\_csv(), который вы можете использовать для указания столбца в файле CSV, который следует использовать в качестве меток строк? Именно это вам и нужно здесь!

Код на Python, который решает предыдущее упражнение, уже включен; можете ли вы внести соответствующие изменения, чтобы исправить импорт данных?

### Инструкции

* Запустите код с помощью кнопки "Запустить код" и убедитесь, что первый столбец должен фактически использоваться в качестве меток строк.
* Укажите аргумент index\_col внутри pd.read\_csv(): установите его на 0, чтобы первый столбец был использован в качестве меток строк.
* Улучшился ли вывод cars сейчас?

# Исправление импорта с помощью включения index\_col  
  
# Вывод cars

cars\_per\_cap country drives\_right  
US 809 United States True  
AUS 731 Australia False  
JAP 588 Japan False  
IN 18 India False  
RU 200 Russia True  
MOR 70 Morocco True  
EG 45 Egypt True

## Квадратные скобки (1)

В видео вы видели, что вы можете индексировать и выбирать данные в Pandas DataFrame разными способами. Самый простой, но не самый мощный способ - использовать квадратные скобки.

В примере кода те же данные об автомобилях импортируются из файлов CSV в виде Pandas DataFrame. Чтобы выбрать только столбец cars\_per\_cap из cars, вы можете использовать:

cars['cars\_per\_cap']  
cars[['cars\_per\_cap']]

Одиночная скобка создает объект Pandas Series, двойная скобка создает объект Pandas DataFrame.

### Инструкции

* Используйте одинарные квадратные скобки для вывода столбца country из cars как объекта Pandas Series.
* Используйте двойные квадратные скобки для вывода столбца country из cars как объекта Pandas DataFrame.
* Используйте двойные квадратные скобки для вывода DataFrame с колонками country и drives\_right из cars, в данном порядке.

# Вывод столбца 'country' в виде объекта Pandas Series  
  
# Вывод столбца 'country' в виде объекта Pandas DataFrame  
  
# Вывод DataFrame с столбцами 'country' и 'drives\_right'

US United States  
AUS Australia  
JAP Japan  
IN India  
RU Russia  
MOR Morocco  
EG Egypt  
Name: country, dtype: object  
 country  
US United States  
AUS Australia  
JAP Japan  
IN India  
RU Russia  
MOR Morocco  
EG Egypt  
 country drives\_right  
US United States True  
AUS Australia False  
JAP Japan False  
IN India False  
RU Russia True  
MOR Morocco True  
EG Egypt True

## Квадратные скобки (2)

Квадратные скобки могут делать не только выбор столбцов, но и получать строки или наблюдения из DataFrame. Следующий вызов выбирает первые пять строк из DataFrame cars:

cars[0:5]

Результатом будет еще один DataFrame, содержащий только указанные вами строки.

Обратите внимание: вы можете выбирать строки с помощью квадратных скобок, только если указываете срез, например, 0:4. Кроме того, вы используете целочисленные индексы строк, а не метки строк!

### Инструкции

* Выберите первые 3 наблюдения из cars и выведите их.
* Выберите четвертое, пятое и шестое наблюдения, соответствующие индексам строк 3, 4 и 5, и выведите их.

# Импорт данных по автомобилям  
  
# Вывод первых 3 наблюдений  
  
  
# Вывод четвертого, пятого и шестого наблюдений

cars\_per\_cap country drives\_right  
US 809 United States True  
AUS 731 Australia False  
JAP 588 Japan False  
 cars\_per\_cap country drives\_right  
IN 18 India False  
RU 200 Russia True  
MOR 70 Morocco True

## loc и iloc (1)

С помощью loc и iloc можно выполнять практически любую операцию выбора данных в DataFrame, о которой вы можете подумать. loc основан на метках, что означает, что вы должны указать строки и столбцы на основе их меток строк и столбцов. iloc основан на целочисленных индексах, поэтому вам нужно указывать строки и столбцы по их целочисленным индексам, как вы делали в предыдущем упражнении.

Попробуйте выполнить следующие команды в IPython Shell, чтобы экспериментировать с loc и iloc для выбора наблюдений. Каждая пара команд здесь дает одинаковый результат.

cars.loc['RU']  
cars.iloc[4]  
  
cars.loc[['RU']]  
cars.iloc[[4]]  
  
cars.loc[['RU', 'AUS']]  
cars.iloc[[4, 1]]

Как и раньше, включен код, который импортирует данные об автомобилях как DataFrame в Pandas.

### Инструкции

* Используйте loc или iloc, чтобы выбрать наблюдение, соответствующее Японии, в виде Series. Метка этой строки - JAP, индекс - 2. Обязательно выведите полученный Series.
* Используйте loc или iloc, чтобы выбрать наблюдения для Австралии и Египта в виде DataFrame. Вы можете узнать метки/индексы этих строк, осмотрев cars в IPython Shell. Обязательно выведите полученный DataFrame.

# Вывод наблюдения для Японии  
  
# Вывод наблюдений для Австралии и Египта

cars\_per\_cap 588  
country Japan  
drives\_right False  
Name: JAP, dtype: object  
 cars\_per\_cap country drives\_right  
AUS 731 Australia False  
EG 45 Egypt True

## loc и iloc (2)

loc и iloc также позволяют выбирать как строки, так и столбцы из DataFrame. Чтобы провести эксперимент, попробуйте следующие команды в IPython Shell. Опять же, парные команды дают одинаковый результат.

cars.loc['IN', 'cars\_per\_cap']  
cars.iloc[3, 0]  
  
cars.loc[['IN', 'RU'], 'cars\_per\_cap']  
cars.iloc[[3, 4], 0]  
  
cars.loc[['IN', 'RU'], ['cars\_per\_cap', 'country']]  
cars.iloc[[3, 4], [0, 1]]

### Инструкции

* Выведите значение drives\_right для строки, соответствующей Марокко (метка строки - MOR).
* Выведите под-DataFrame, содержащий наблюдения для России и Марокко, а также столбцы country и drives\_right.

# Вывести значение drives\_right для Марокко  
  
# Вывести под-DataFrame

True  
 country drives\_right  
RU Russia True  
MOR Morocco True

## loc и iloc (3)

Возможно также выбирать только столбцы с помощью loc и iloc. В обоих случаях достаточно указать срез от начала до конца перед запятой:

cars.loc[:, 'country']  
cars.iloc[:, 1]  
  
cars.loc[:, ['country','drives\_right']]  
cars.iloc[:, [1, 2]]

### Инструкции

* Выведите столбец drives\_right в виде Series, используя loc или iloc.
* Выведите столбец drives\_right в виде DataFrame, используя loc или iloc.
* Выведите как столбец cars\_per\_cap, так и drives\_right в виде DataFrame, используяlocилиiloc`.

# Вывести столбец drives\_right как Series  
  
# Вывести столбец drives\_right как DataFrame  
  
# Вывести cars\_per\_cap и drives\_right как DataFrame

US True  
AUS False  
JAP False  
IN False  
RU True  
MOR True  
EG True  
Name: drives\_right, dtype: bool  
 drives\_right  
US True  
AUS False  
JAP False  
IN False  
RU True  
MOR True  
EG True  
 cars\_per\_cap drives\_right  
US 809 True  
AUS 731 False  
JAP 588 False  
IN 18 False  
RU 200 True  
MOR 70 True  
EG 45 True