2. SQL問い合わせ（SQLite3, Python を使用）

データベース演習

URL: <http://www.kkaneko.jp/cc/dbenshu/index.html>

**概要**

前半は，テーブル定義でのデータ型と主キーの指定を行う．後半は，SQL問い合わせの結果を，カーソルで扱う．

Pythonと SQL を使う。

■ **SQL 問い合わせ (SQL Query)**

SQL 問い合わせは，リレーショナルデータベースに格納された1つまたは複数のテーブルを使う．例えば，次のように書くと，*m* 個のテーブル *T*1, *T*2, ..., *T*m の直積集合から条件 <*expression*> を満足する行のみを選び，そうして出来たテーブルの属性 *A*1, *A*2, ..., *A*n を出力するという意味になる．

SELDCT *A*1, *A*2, ..., *A*n

FROM *T*1, *T*2, ,,,, *T*m

WHERE <*expression*>

**■ 条件 (Condition)**

テーブル名とドット「**.**」と列名の並びを列の修飾名と呼ぶ．例えば，テーブル*R* の列 *A* の修飾名は *R.A* である．文字定数は，**'X'** のように，シングルクォーテーションマーク「**'**」で囲む．比較演算子は **=, >, <, >=, <=, <>** の6種類がある．探索条件は「**R.A > 20**」のように，列の修飾名と比較演算子と定数の並びである．扱うテーブルが1つのときは，列の修飾名の代わりに列名を使うことができる（「**A > 20**」のように）．条件は**and** や **or** で連結することができる．

**準備**

1. **Python のインストール**

　WindowsでのPython のインストールは次のページで説明している．

[**https://www.kkaneko.jp/tools/win/python.html**](https://www.kkaneko.jp/tools/win/python.html)

2. **pandas のインストール**

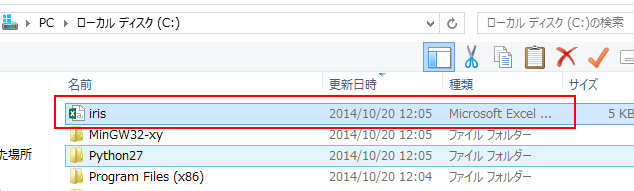
　Windows のコマンドプロンプトを**管理者として開き**，次のコマンドを実行．

**pip install -U pandas**

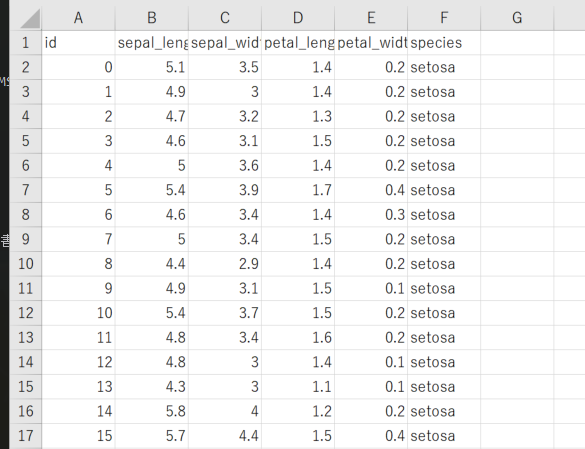
**CSVファイル iris.csv のダウンロード**

1. Web ブラウザで <https://www.kkaneko.jp/sample/iris.csv>を開き、ダウンロードする

② ダウンロードされたファイル **iris.csv** を **C:\iris.csv にコピー**



③ iris.csv は次のようなファイルである．



**テーブル定義**

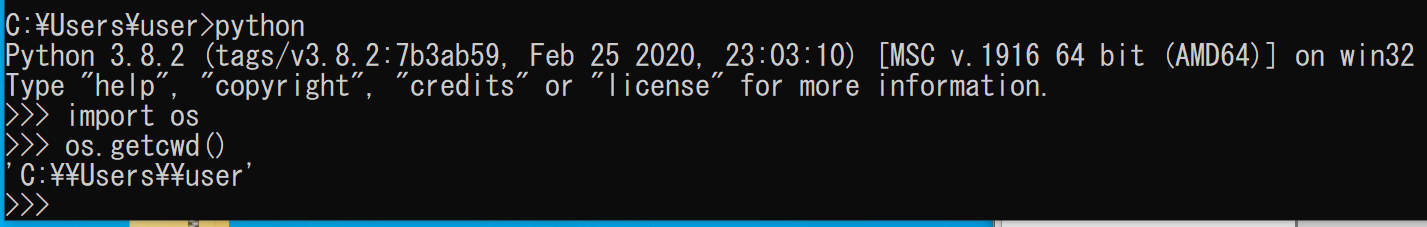
①　Python を起動する

**python**

②　**os.getcwd()** により**カレントディレクトリを確認．**

**import os**

**os.getcwd()**



③ **テーブル定義**を行う，**次の Python プログラムを実行**

**データ型の指定は，各列で，integer, real などのデータ型を指定している．**

**id は主キーであるので「primary key」を指定している．**

**※** 複数行にわたるユニコード文字列を使いたいところでは、「u""" ・・・ """」のように書く．

import pandas as pd

import sqlite3

c = sqlite3.connect('**hoge.sqlite**')

sql = u"""

**create table iris (**

**id integer primary key,**

**sepal\_length real,**

**sepal\_width real,**

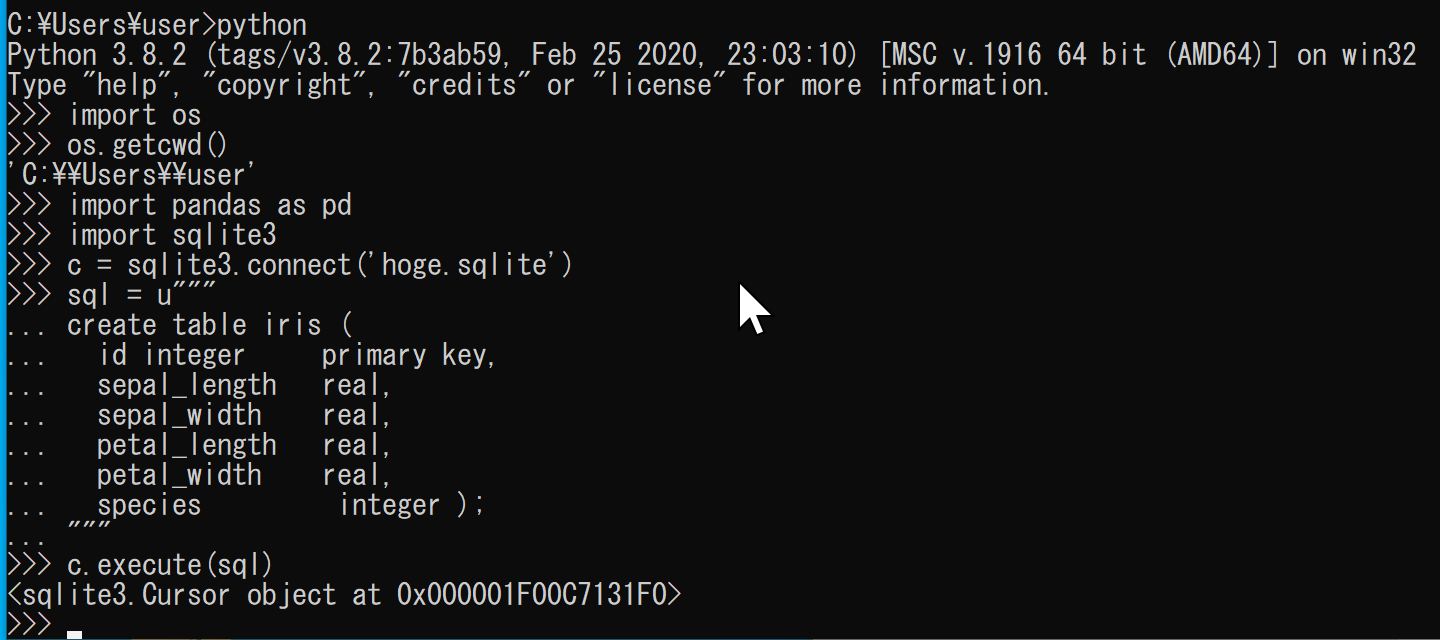
**petal\_length real,**

**petal\_width real,**

**species integer );**

"""

c.execute(sql)

****

④ **テーブル生成**（空のテーブルにレコードを挿入）を行う，**次の Python プログラムを実行**

　プログラム中の「?」は**SQLプレースホルダー**である．

x = pd.read\_csv( '**c:\\iris.csv**')

for index, r in x.iterrows():

sql = u"**insert into iris values (?, ?, ?, ?, ?, ?)**"

c.execute(sql, (**r[0], r[1], r[2], r[3], r[4], r[5]**))

c.commit()

テキスト

自動的に生成された説明

(1) オブジェクト **x** に CSVファイルを読み込む

x = pd.read\_csv( '**c:\\iris.csv**')

* Windows でのファイル名「**C:\iris.csv**」は、Python のプログラム中では「**'C:\\iris.csv'**」のように書く
* 読み込みたいCSVファイルの先頭行に、「id, sepal\_length, sepal\_width, petal\_length, petal\_width, species」のようなヘッダーが**ない**場合には「header=None」を付ける

（今回は，ヘッダーがあるので，「header=None」を付けない）

(2) オブジェクト **x** に格納されたデータを iris テーブルに挿入する

for index, r in x.iterrows():

sql = u"**insert into iris values (?, ?, ?, ?, ?, ?)**"

c.execute(sql, (**r[0], r[1], r[2], r[3], r[4], r[5]**))

* 「**for r in x:**」は x の各行について繰り返すという Python プログラム
* 「**insert into iris values**」は、テーブルに1行挿入するというSQLプログラム
* 「?」は，SQLプレースホルダー
* 「**r[0], r[1], r[2], r[3], r[4], r[5]**」は、もとのCSVデータファイルの0列目、１列目、2列目、3列目、4列目、5列目を使うという意味

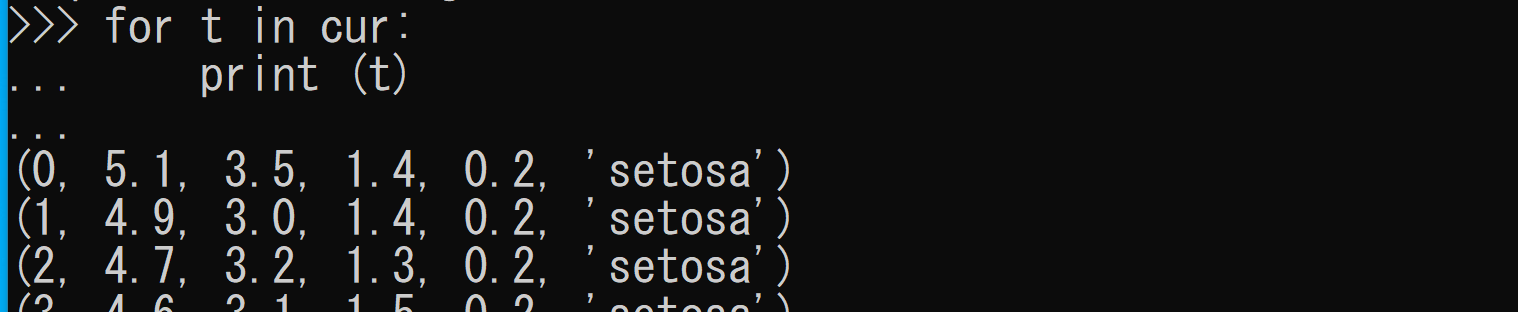
⑤ テーブルをすべて読みだす. カーソルを使う．

**cur = c.cursor()**

**cur.execute(u"select \* from iris")**

**for t in cur:**

**print (t)**

****

* 「**select \* from iris**」は，SQL 問い合わせ
* 「**cur**」は，カーソルである．問い合わせ結果を得るのに使う．

⑥ テーブルから，条件に合致するレコードを得る．

**cur = c.cursor()**

**cur.execute(u"select \* from iris where id = 2")**

**for t in cur:**

**print (t)**

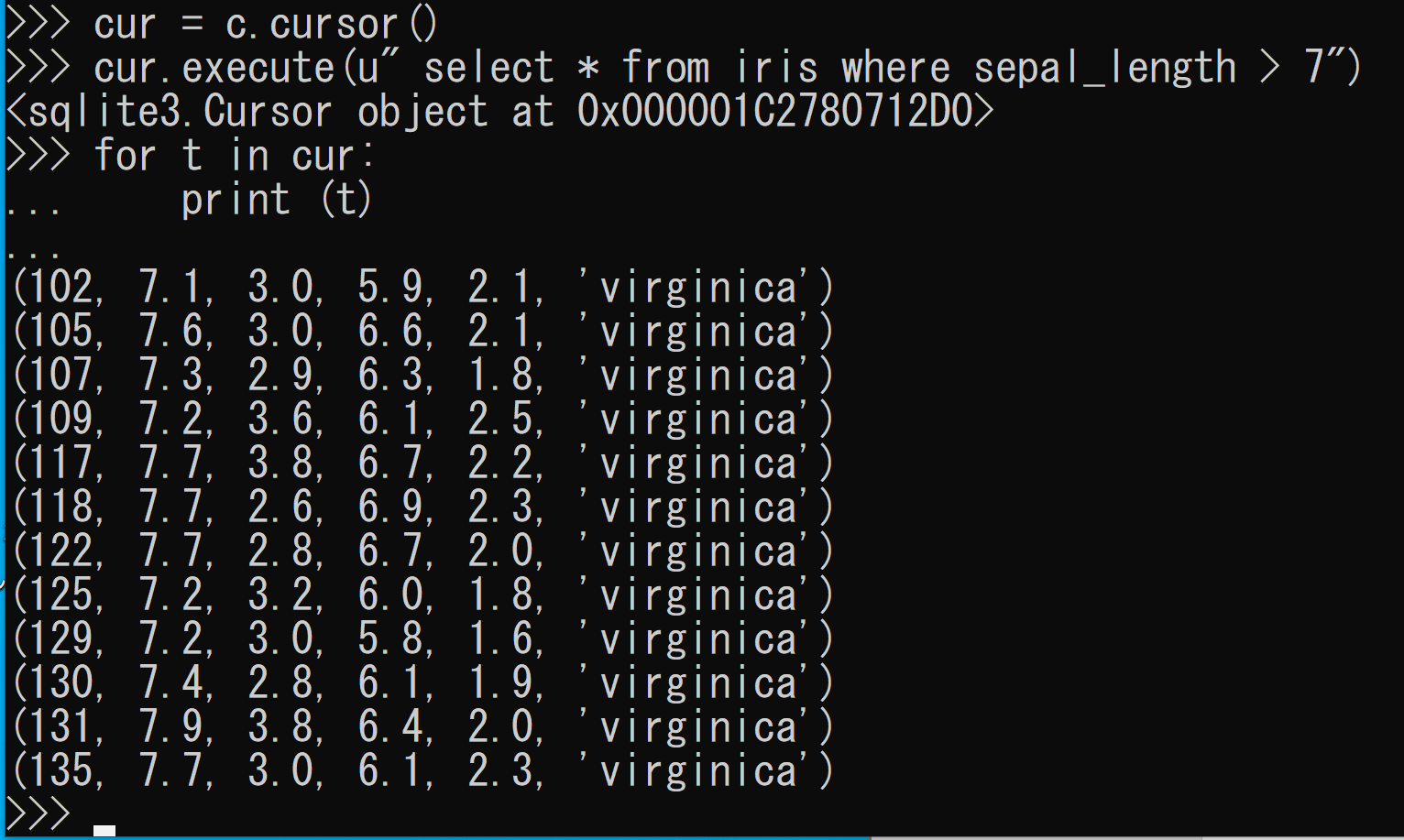
****

**cur = c.cursor()**

**cur.execute(u" select \* from iris where sepal\_length > 7")**

**for t in cur:**

**print (t)**

****

⑦ 終了処理として，データベース接続を切断する．

**c.close()**

