**MySQL 연동**

python을 이용해서 MySQL 연동을 해보자.

먼저 pip나 PyCharm 인터프리터 설정으로 pymysql 모듈을 설치해준다.

|  |  |  |
| --- | --- | --- |
| 1 | pip install PyMySQL | [cs](http://colorscripter.com/info#e) |

엑셀파일도 다루어야하기 때문에 openpyxl 모듈이 없다면 마찬가지로 pip를 이용해서 설치해준다.

아래 모듈들을 import 해준다.

|  |  |  |
| --- | --- | --- |
| 1  2  3 | import pymysql  from openpyxl import Workbook  from openpyxl import load\_workbook | [cs](http://colorscripter.com/info#e) |

간단한 테스팅을 위해 local에 설치된 MySQL DB에 테이블을 만들어준다.

|  |  |  |
| --- | --- | --- |
| 1 | create table test(num int(11), name varchar(10)); | [cs](http://colorscripter.com/info#e) |

필수적인 부분은 아니지만, 가독성을 위해 CRUD함수들에 사용할 클래스를 하나 만들어 놓자.

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5 | #DB 테이블 칼럼대로 만든 객체  class Test:      def \_\_init\_\_(self, num, name):          self.num = num          self.name = name | [cs](http://colorscripter.com/info#e) |

이제 각 CRUD를 함수로 만들어 사용하자.

모든 함수들은 connection Leak를 막아주기 위해 try .. finally 구문을 사용하고, with문으로  cursor리소스를 자동으로 해제되도록 코드를 작성한다.

**DB Select All**

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | #전체 Select  def select\_all():      conn = pymysql.connect(host='localhost', user='root', password='비밀번호', db='python', charset='utf8')      try:          with conn.cursor() as curs:              sql = "select \* from test"              curs.execute(sql)              rs = curs.fetchall()              for row in rs:                  print(row)      finally:          conn.close() | [cs](http://colorscripter.com/info#e) |

**DB Insert**

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | #DB Insert  def insert\_test(test\_obj):      conn = pymysql.connect(host='localhost', user='root', password='비밀번호', db='python', charset='utf8')      try:          with conn.cursor() as curs:              sql = 'insert into test values(%s, %s)'              curs.execute(sql, (test\_obj.num, test\_obj.name))          conn.commit()      finally:          conn.close() | [cs](http://colorscripter.com/info#e) |

**DB Delete**

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | #num칼럼으로 DB Delete  def delete\_test(num):      conn = pymysql.connect(host='localhost', user='root', password='비밀번호', db='python', charset='utf8')      try:          with conn.cursor() as curs:              sql = 'delete from test where num=%s'              curs.execute(sql, num)          conn.commit()      finally:          conn.close() | [cs](http://colorscripter.com/info#e) |

**DB Delete All**

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | #DB Delete All  def delete\_all():      conn = pymysql.connect(host='localhost', user='root', password='비밀번호', db='python', charset='utf8')      try:          with conn.cursor() as curs:              sql = 'delete from test'              curs.execute(sql)          conn.commit()      finally:          conn.close() | [cs](http://colorscripter.com/info#e) |

**DB Update**

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | #DB Update  def update\_test(test\_obj):      conn = pymysql.connect(host='localhost', user='root', password='비밀번호', db='python', charset='utf8')      try:          with conn.cursor() as curs:              sql = 'update test set name=%s where num=%s'              curs.execute(sql, (test\_obj.name, test\_obj.num))          conn.commit()      finally:          conn.close() | [cs](http://colorscripter.com/info#e) |

**MySQL 데이터 엑셀파일 쓰기**

종종 DB의 내용을 엑셀파일로 받어서 확인을 해야하는 경우가 있다.

csv파일로 만들어서 바로 올리고 내려받고 할 수 있는데... 필자가 여러번 해본 경험으로 한글 때문에 문자가 깨지고, 간혹 데이터 안에 ","가 들어 있으면 다른 특수문자를 사용하여 구분을 해야 정확하게 정제된 데이터가 떨어진다.

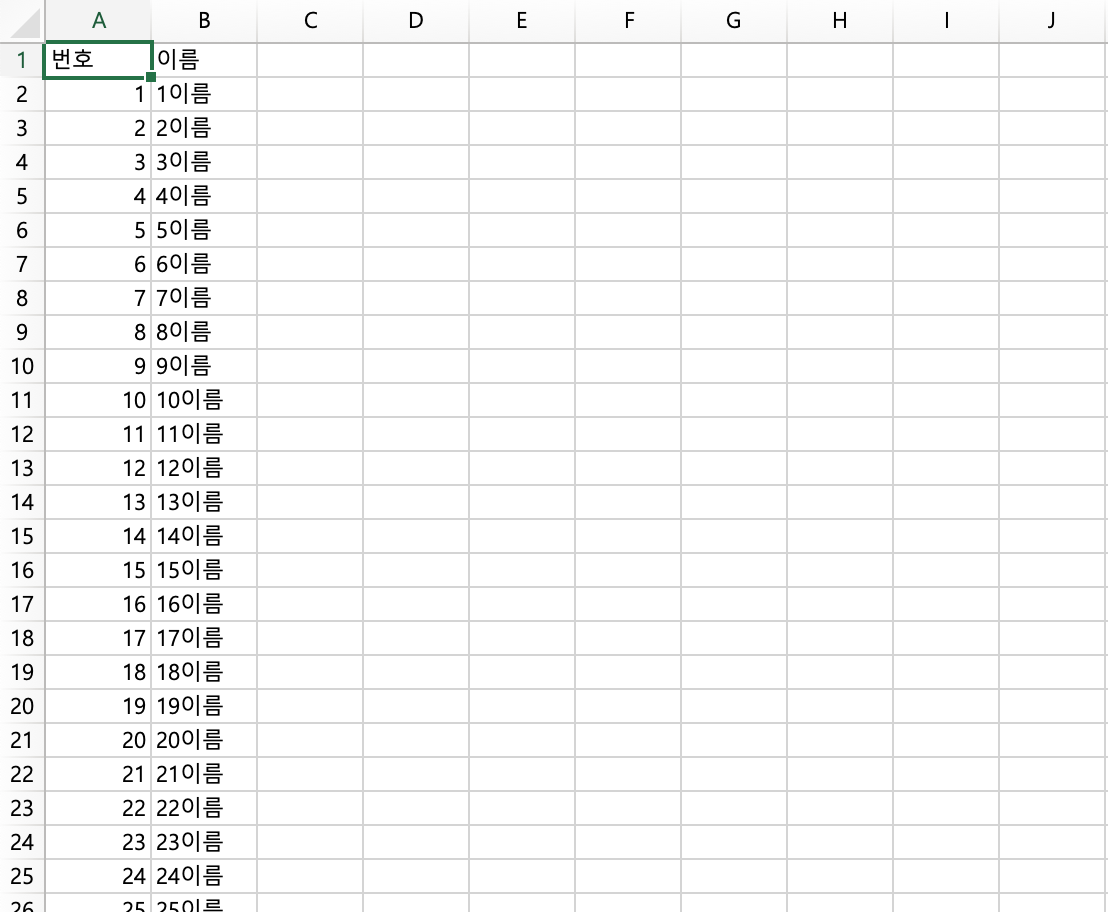
어차피 csv파일도 엑셀파일로 변환해야할거 속도적인 측면에서 많이 차이가 안난다면 Python을 이용해서 데이터를 이동시키는게 편하다.

insert\_test함수를 사용해서 데이터를 1000개정도 만들어 놓고 엑셀파일로 떨어진것을 확인해보자.

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33 | #전체 Select 하여 엑셀파일 쓰기  def select\_all\_to\_excel():      conn = pymysql.connect(host='localhost', user='root', password='비밀번호', db='python', charset='utf8')      try:          with conn.cursor() as curs:              sql = "select \* from test"              curs.execute(sql)              rs = curs.fetchall()                wb = Workbook()              ws = wb.active                #첫행 입력              ws.append(('번호','이름'))                #DB 모든 데이터 엑셀로              for row in rs:                  ws.append(row)                wb.save('/Users/Jamong/Desktop/숫자.xlsx')      finally:          conn.close()          wb.close()      if \_\_name\_\_ == "\_\_main\_\_":      #데이터 1000개정도 넣기      for i in range(1,1000):          test = Test(i, str(i) + '이름')          insert\_test(test)        #DB -> 엑셀파일      select\_all\_to\_excel() | [cs](http://colorscripter.com/info#e) |

-결과

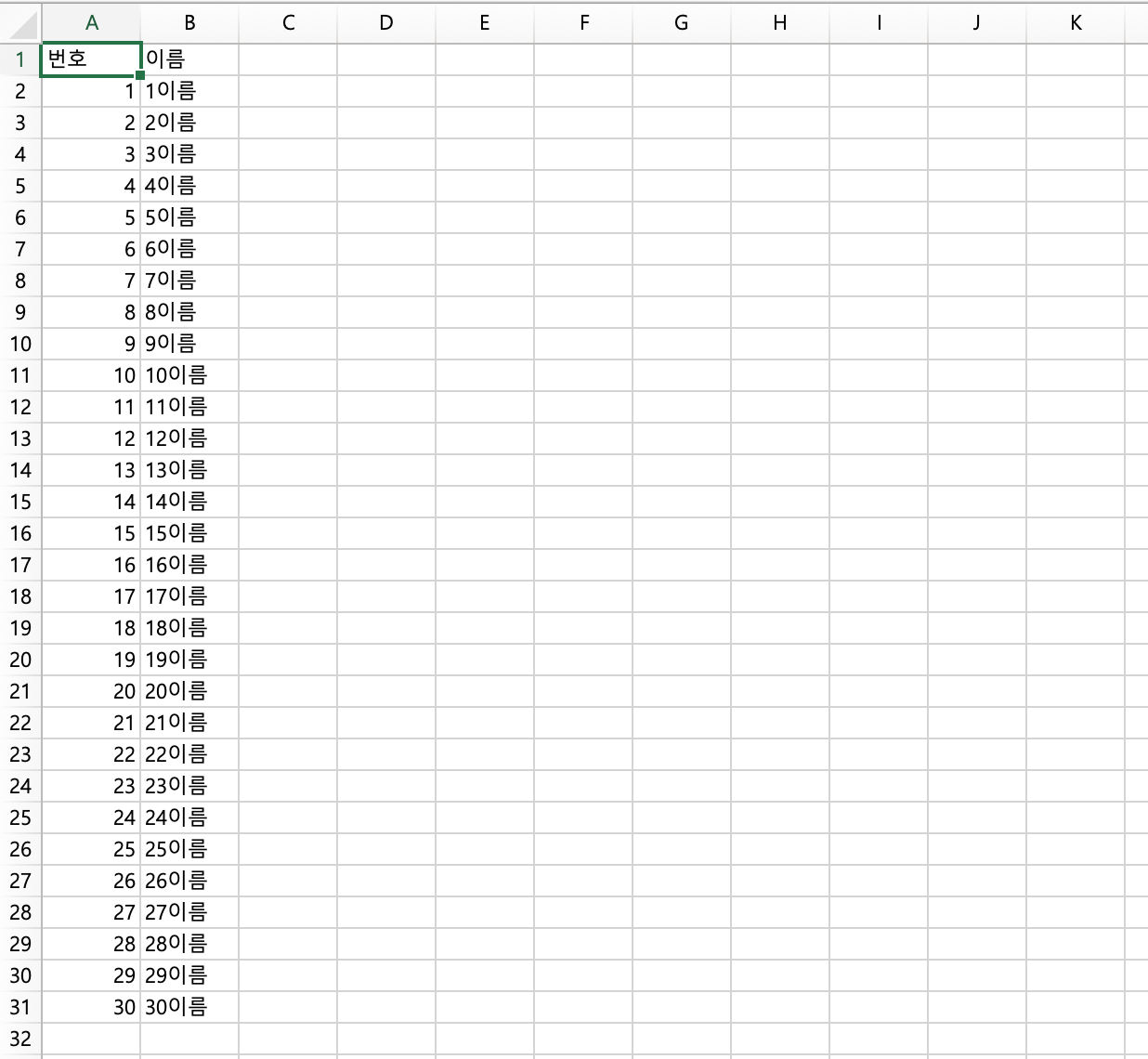
정확이 엑셀파일로 이상없이 만들어진 것을 확인한다.



**엑셀파일 MySQL DB Insert**

이번에는 반대로 엑셀파일의 데이터를 MySQL DB로 Insert할거다.

엑셀파일을 열어 30개의 데이터만 남기고 파일을 저장한다.



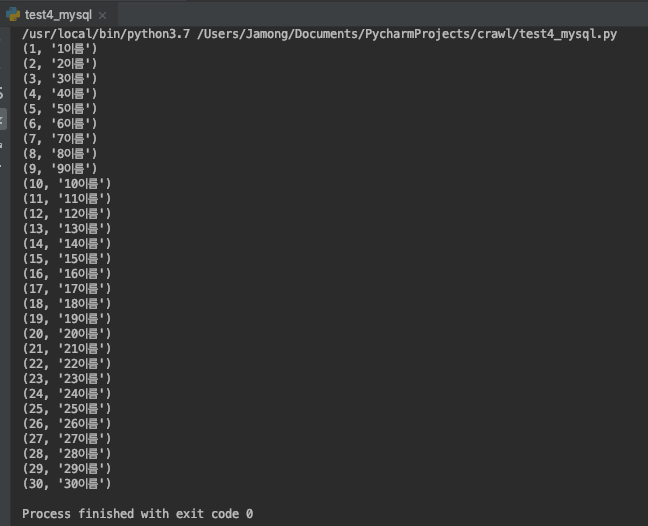
위에서 작성한 함수로 현재 DB에 들어있는 데이터를 모두 삭제한 후

엑셀 파일을 DB로 Insert해보자.

iter를 사용한 이유는 엑셀파일의 첫번째 행을 빼고 반복문을 돌릴려고 사용했다.

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | #엑셀파일 DB Insert  def insert\_excel\_to\_db():      conn = pymysql.connect(host='localhost', user='root', password='비밀번호', db='python', charset='utf8')      try:          with conn.cursor() as curs:              sql = 'insert into test values(%s, %s)'                wb = load\_workbook('/Users/Jamong/Desktop/숫자.xlsx',data\_only=True)              ws = wb['Sheet']                iter\_rows = iter(ws.rows)              next(iter\_rows)              for row in iter\_rows:                  curs.execute(sql, (row[0].value, row[1].value))              conn.commit()      finally:          conn.close()          wb.close()      if \_\_name\_\_ == "\_\_main\_\_":      delete\_all()      insert\_excel\_to\_db()      select\_all() | [cs](http://colorscripter.com/info#e) |

-결과



select\_all함수를 이용해서 현재  DB의 목록을 Select해보니 작성한 엑셀 데이터가 이상없이 Insert된것을 확인했다.