

Python

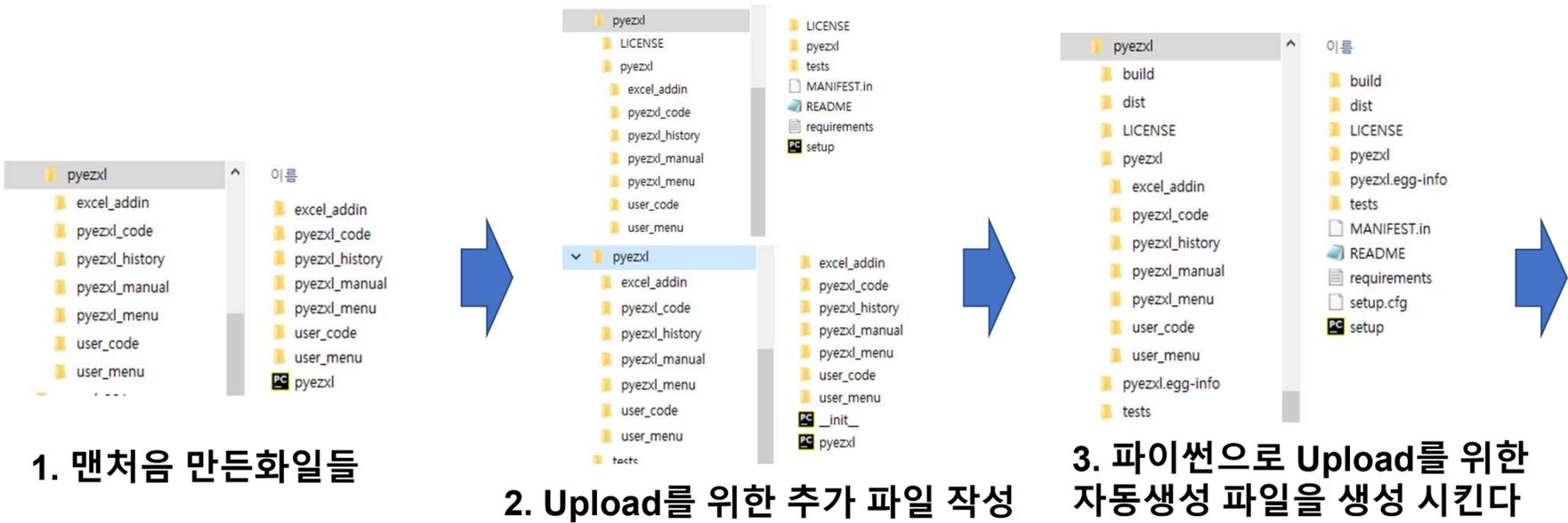
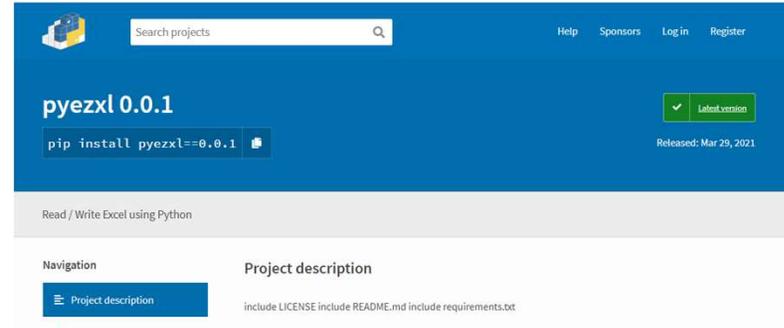
Pypi 업로드 파일 만들기

2021년 04월 19일
박 상 진

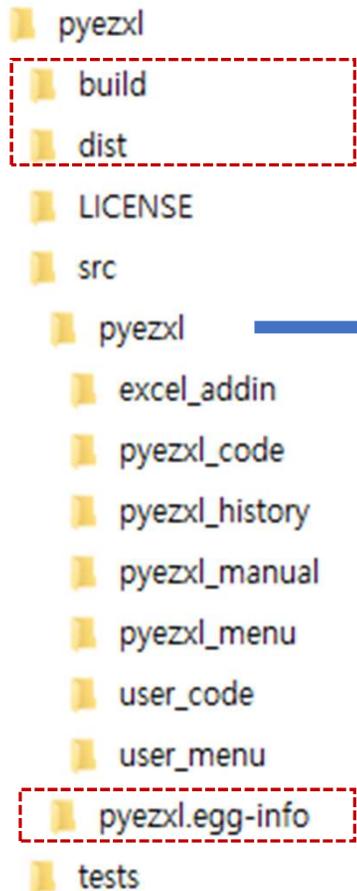
1. 기본설명

오른쪽과 같이 내가 만든 모듈을 "pip install pyezxl"를 이용하여 설치를 하고 싶다면

1. 만들 파일을 작성
2. Pypi에 Upload가능한 형태로 추가 요구되는 화일들을 작성
3. 변경할 패키지를 변환
4. Pypi에 upload



2. 전체적인 폴더 설명

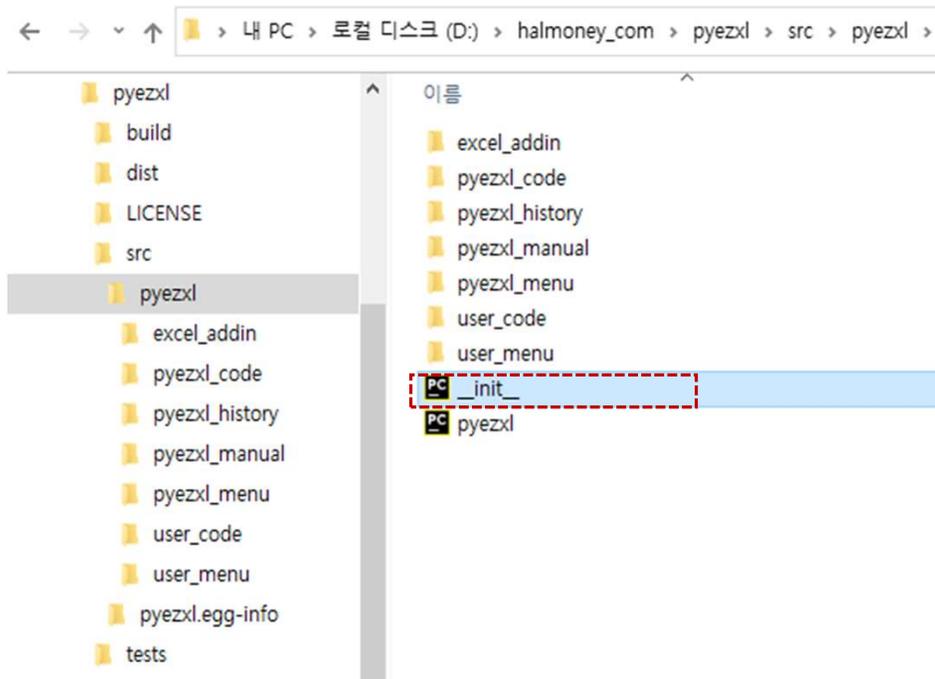


이 폴더는 패키지가 빌드될때 자동으로 생성되는 파일들이니 신경쓰지 마세요



pyezxl안의 __init__.py는 Upload할때 패키지 내용이 들어가는 곳

3. 주요 파일의 내용 -1



__init__.py는 패키지 내용이 들어가는 내용

```
# __init__.py  
# Copyright (C) 2021 (sjpkorea@yahoo.com) and contributors  
#
```

```
import inspect  
import os  
import sys
```

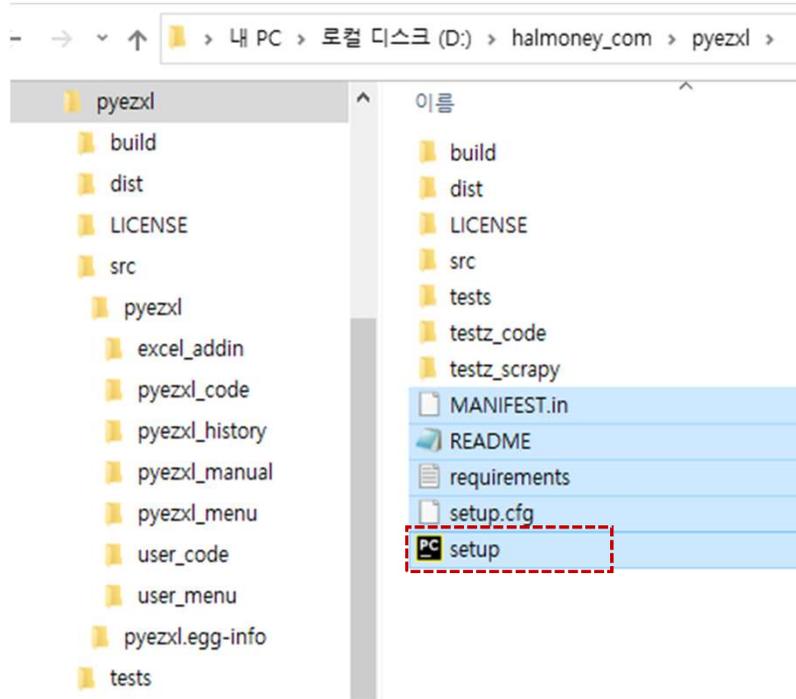
```
__version__ = '0.0.1'
```

```
real_path =  
os.path.dirname(os.path.abspath(__file__)).replace("\\", "/")  
sys.path.append(real_path)
```

```
try:  
    from pyezxl import pyezxl  
except ImportError as e:  
    print(e, " Please re-check.")  
    exit(1)
```

```
__all__ = [name for name, obj in locals().items()  
           if not (name.startswith('_') or inspect.ismodule(obj))]
```

4. 주요 파일의 내용 -2



Setup.py : 이 패키지에 대한 일반적인 설명이 들어있음

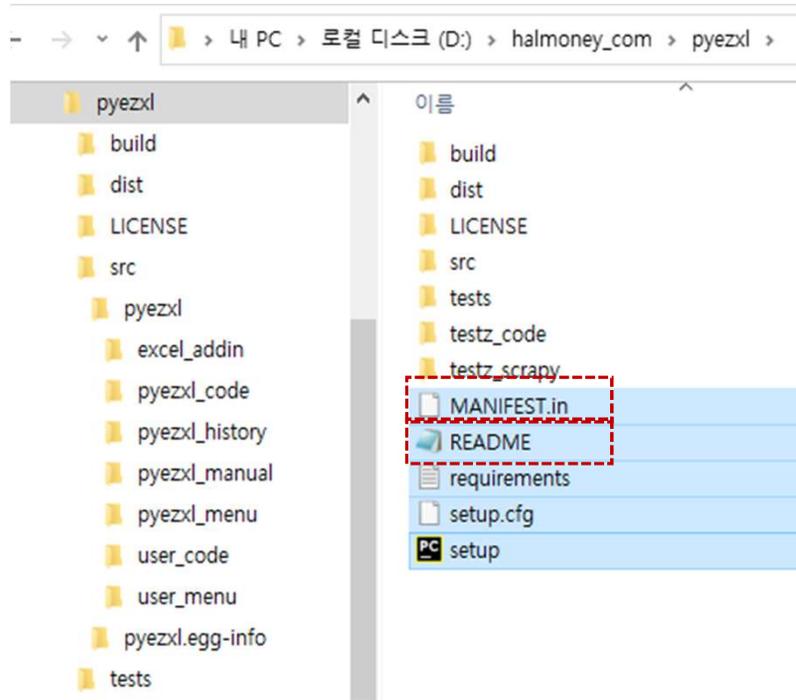
```
from setuptools import setup, find_packages

with open("README.md", "r") as fh:
    long_description = fh.read()

setup(name='pyezxl',
      version='0.0.6',
      url='https://github.com/sjpark/pyezxl',
      download_url='https://github.com/sjpark/pyezxl/archive/v0.0.6.tar.gz',
      author='sjpark',
      author_email='sjpkorea@yahoo.com',
      description='Read / Write Excel using Python',

      packages=find_packages("src"),
      package_dir={"": "src"},
      include_package_data=True,
      package_data={
          "pyezxl": ["excel_addin/*.***"],
          "pyezxl": ["pyezxl_code/*.***"],
          "pyezxl": ["pyezxl_history/*.***"],
          "pyezxl": ["pyezxl_manual/*.***"],
          "pyezxl": ["pyezxl_menu/*.***"],
          "pyezxl": ["user_code/*.***"],
          "pyezxl": ["user_menu/*.***"],
      },
      long_description=open('README.md').read(),
      install_requires=[],
      python_requires='>=3.5',
      zip_safe=False,
      classifiers=[
          'License :: OSI Approved :: MIT License'
      ],
    )
```

5. 주요 파일의 내용 -3



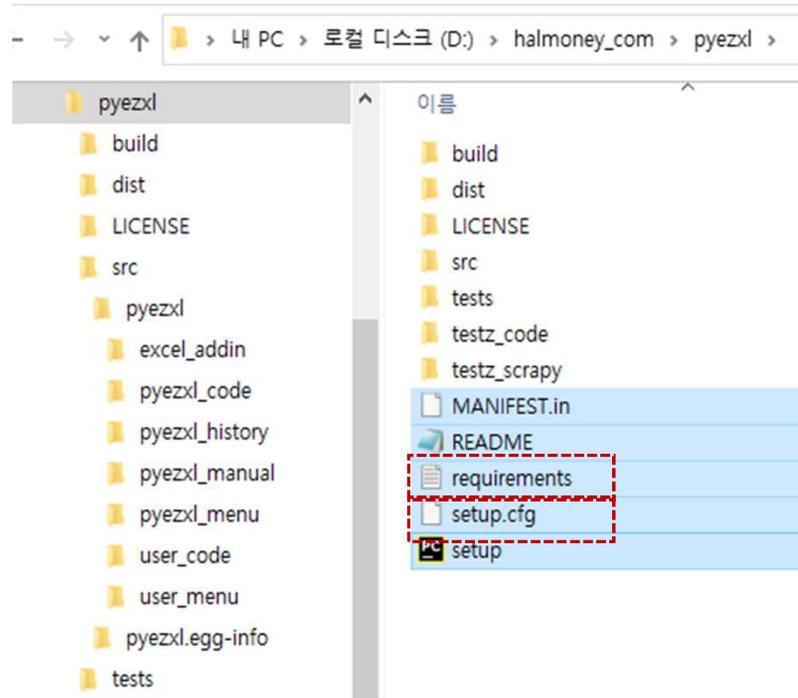
MANIFEST.in : 하위 내용들까지 같이 언급을 하셔야 전체가 다 올라갑니다
저도 처음에는 전부가다 올라가지 않아서 알아보니 여기에 올라갈 화일들을 언급해줘야 하더라구요...

```
include LICENSE
include README.md
include requirements.txt
include src/pyezxl/excel_addin/*.*
include src/pyezxl/pyezxl_code/*.*
include src/pyezxl/pyezxl_history/*.*
include src/pyezxl/pyezxl_manual/*.*
include src/pyezxl/pyezxl_menu/*.*
include src/pyezxl/user_code/*.*
include src/pyezxl/user_menu/*.*
```

Readme.md : 이 패키지에대한 일반적인 설명들

```
include LICENSE
include README.md
include requirements.txt
```

6. 주요 파일의 내용 -4

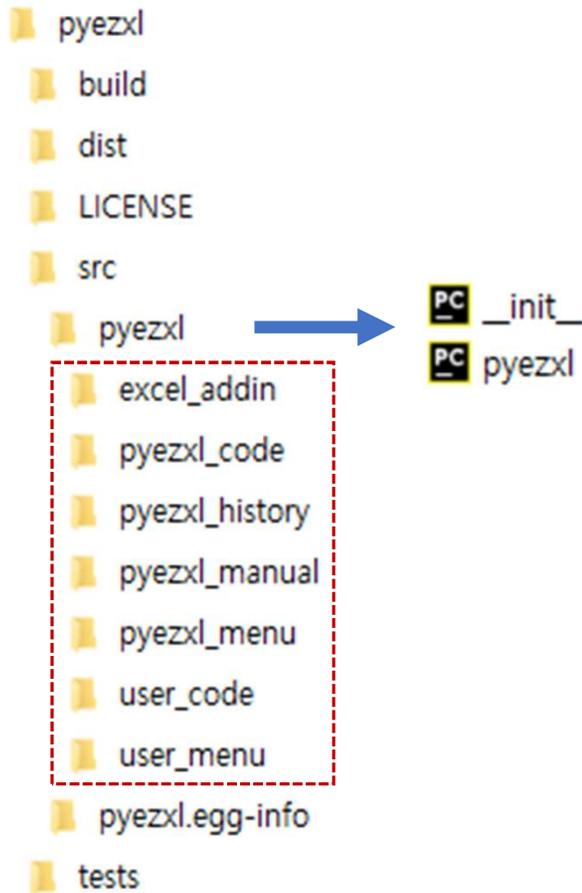


Requirements.txt

```
[metadata]  
description-file = README.md
```

Setup.cfg : 빈화일

7. 주요 파일의 내용 -4



이름	설명
pyezxl.py	이 패키지의 가장 중요한 파일로 Python으로 엑셀을 제어하기위한 파일이다
excel_addin(폴더)	안에 들어있는 엑셀화일을 엑셀이 설치된위치중 xlstart안에 넣으면 자동으로 엑셀이 실행될때 추가 메뉴가 뜨도록 하는 것이다
pyezxl_code(폴더)	위의 추가 메뉴를 누르면 실행되는 코드가 들어 있는것. 기본적으로 pyezxl.py를 사용한다
pyezxl_history(폴더)	만들어진 배경을 설명하는 부분이다
pyezxl_manual(폴더)	이 패키지에 대한 설명 매뉴얼이다
pyezxl_menu(폴더)	안에 txt화일을 만들어서 위의 excel_addin화일이 실행이되면, 이것과 아랫부분의 user_menu 안의 txt화일을 읽어와서 추가메뉴가 나타나게 하는것이다
user_code(폴더)	User_menu로 인하여 만들어진 자신만의 코드를 작성할수있는 공간이다
user_menu(폴더)	pyezxl_menu와 함께 자신이 만든 다시 사용가능한 코드를 만들어서 추가메뉴에 나타나게 하는것이다

8. 패키지 빌드

1) 빌드하기 위해 필요한 **setuptools**와 **wheel**을 설치

```
pip install setuptools wheel
```

(혹시 requests가 없다고 나오면 pip install request를 하기를 바란다)

2) 파일이 설치된 프로젝트 폴더로 이동

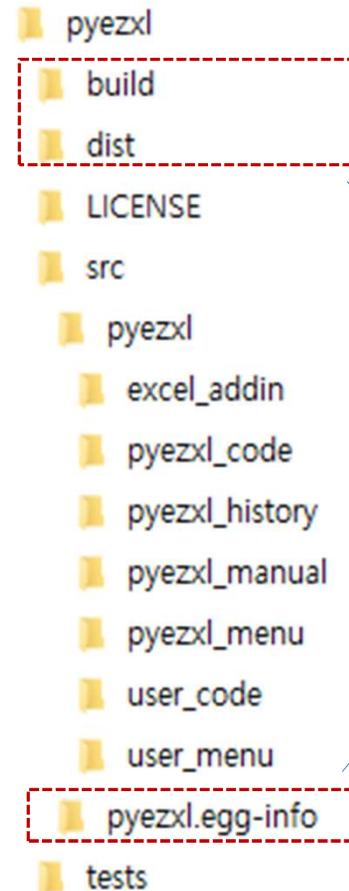
```
python setup.py sdist bdist_wheel
```

(bdist와 wheel사이에 언더바가 있음)

3) Upload를 한다

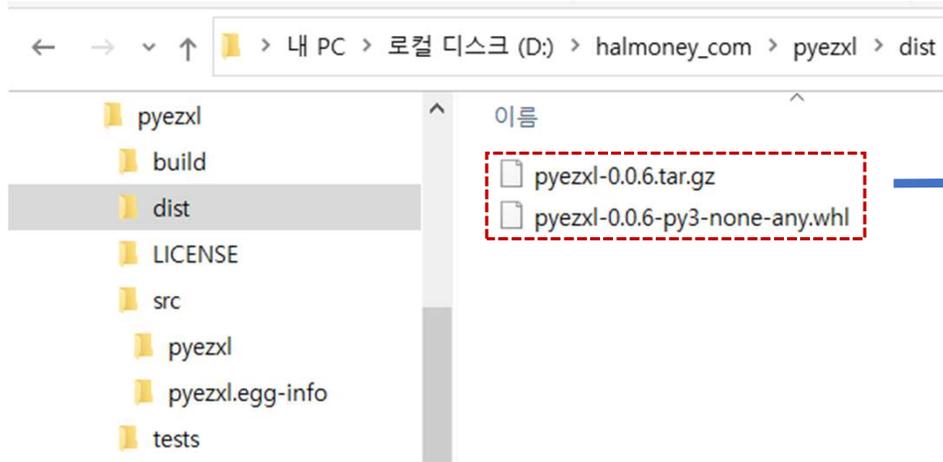
```
python -m twine upload --repository-url
```

```
https://upload.pypi.org/legacy/ dist/*
```



이 폴더는 패키지가 빌드될 때 자동으로 생성되는 파일들이니 신경쓰지 마세요

9. upload



이것이 압축을 해서 설치할때 사용되는것으로, 이러것을 7zip등을 이용해서 풀어본 후, 내가 원하는 모든 파일이 다 들어가있으면 성공한 것이다

아래의 코드와 같이 실행시키면, 위의 2개 파일이 올라가는 것이다

```
python -m twine upload --repository-url https://upload.pypi.org/legacy/ dist/*
```

```
PS D:\halmoney_com\pyezxl_002> python -m twine upload --repository-url https://upload.pypi.org/legacy/ dist/*
Uploading distributions to https://upload.pypi.org/legacy/
Enter your username: sjpark
Enter your password:
Uploading pyezxl-0.0.2-py3-none-any.whl
100%|#####| 11.8k/11.8k [00:02<00:00, 5.05kB/s]
Uploading pyezxl-0.0.2.tar.gz
100%|#####| 12.3k/12.3k [00:01<00:00, 8.80kB/s]

View at:
https://pypi.org/project/pyezxl/0.0.2/
PS D:\halmoney_com\pyezxl_002>
```



Version	Release date	Files	Options
0.0.2	2 minutes ago	2 files (1 Wheel, 1 Source)	Options
0.0.1	Mar 29, 2021	2 files (1 Wheel, 1 Source)	Options

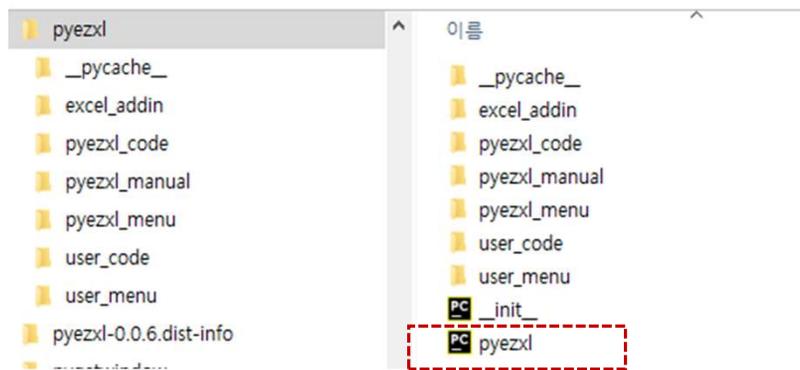
이렇게 다 완료가 되면 자신의 사이트에 upload가 완료된것을 볼수있다

Python

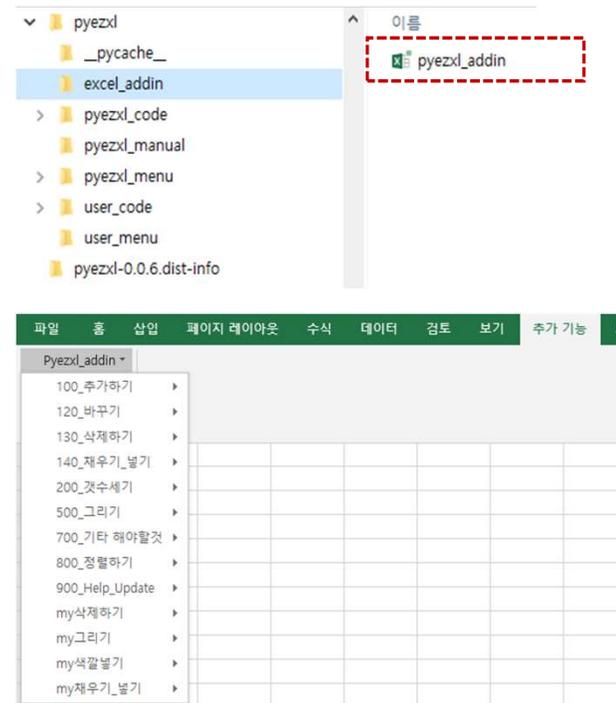
Pyezxl 패키지 소개

1. 소개

두가지 기능을 하는 것이다
하나는 엑셀의 VBA와 같은 기능을 어떻게 하면 쉽고 더욱 다양하게 사용할 수 있을까 에서 시작한 것으로 pyezxl.py모듈이 그것이며



다른 하나는 기본적으로 이 pyezxl.py와 python의 기능들을 이용해서 엑셀에 추가메뉴를 만들고 그것을 클릭하면 기능을 쉽게 사용하도록 만든 것이다 어찌 보면 사용 예라고 할수 도 있다

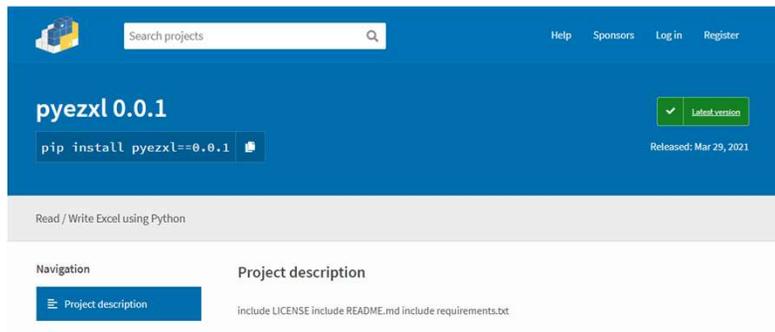


Python

Pyezxl 설치

1. 설치방법

1. Python을 설치한다
2. Win32com을 설치한다 (`pip install pywin32`)
3. pyezxl을 설치한다 (`pip install pyezxl`)



www.halmoney.com에서도 최신화일을 올려 놓을 예정입니다
니다

1. 설치된 Python패키지에서 excel_addin폴더안에 있는 엑셀화일을 엑셀이 설치된곳의 xlstart에 복사를 한다
2. 엑셀화일을 다시 실행하면 추가메뉴가 나타남



- ① 엑셀화일을 엑셀프로그램이 깔린곳의 xlstart안에 복사하여 넣는다
C:\Program Files\Microsoft Office\root\Office16\XLSTART
- ② 엑셀을 실행하면 그림과 같이 추가기능에 나타난다

Python

Pyezxl사용법

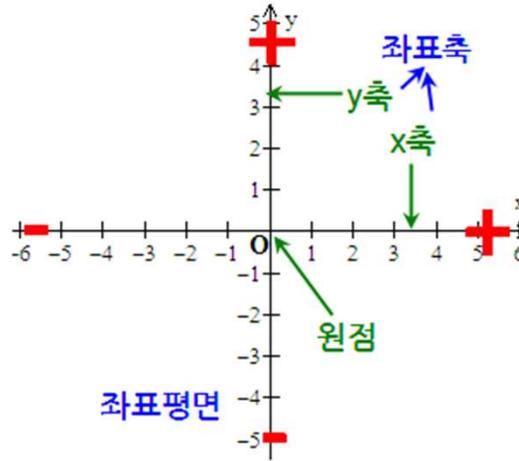
1. 코드 작성의 기본 설명

▣ 함수 이름들

- ① 함수의 이름은 가능하면 쉽게, 별도로 외우지 않아도 연상이 가능한 방법으로 하려고 하였다. 그래서 3 자리로 만들었으며, 어떤 이름은 약간 억지스러운 것도 있으니 이해 바랍니다
- ② 주소와 영역에 사용되는 것은 x,y 로 만들었으며, 이때 x 는 가로를 뜻하는 것으로 사용하였다. X 축을 뜻하면 가로이지만, x 좌표축의 표현 형식으로 보면 y 를 나타내야 하지만, 여기서는 여러 고민중, x 는 가로를 뜻하도록 하였으니 혼돈이 있으셔도 양해 바랍니다(즉, x 를 숫자로 표현할때 x 방향에 있는 몇번째로 할 것인지 xy 좌표계로 할것인지를 고민하였다는 뜻이다)
- ③ 함수의 기본 내용은 외형은 엑셀로 조정을 하고, 내용이 변경되는 부분은 python의 다른 모듈들을 이용하여 사용자가 처리하도록 하였다. 가장 중요한 읽고 쓰는 기능만 있어도, 엑셀의 자동화는 가능하다
- ④ 자주 사용하는 기능은 만들어서 넣었으며
- ⑤ 차후 값들을 변경하거나 sqlite등을 이용하여 개인적으로 사용하던 database를 별도의 함수나 파일로 만들어 다룰 예정이다

2. X, Y좌표에 대하여

	A	B	C	D	E	F	G
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							



- ① 기본적인 x축은 가로의 영역이다
- ② 그런데 왜그런지 엑셀에서의 좌표를 표기할때
- ③ 가로방향이 x로 보이는 것이 더 옳다는 것이다
- ④ 그래서 x는 더 가까운 것이다
- ⑤ 대부분의 자료는 세로 방향으로 많이 적으므로 x는 x방향으로 정의한다



1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
A	B	C	D	E	F	G	

1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
A	B	C	D	E	F	G	

X 방향

Y 방향

3. 함수의 이름 및 기타 이름 규칙

▣ 함수 이름

실행동사_적용대상_변화내용

(언더바로 3개를 연결한다)

* 실행동사

read, write, Insert, delete, change, set, copy, move 등

- database에서 사용하는 용어사용
- Set : 변경할 내용이 값이 아닌경우

* 적용대상

range, sheet, x, y, line 등

* 변화내용

name, color, value, end 등

▣ 주소사용방법

- 가로와 세로 표기 : x = 가로, row, y = 세로, column
- 한개의 셀 : cell사용
- 여러개의 셀 : range사용

예제)

B1 => [1,2]

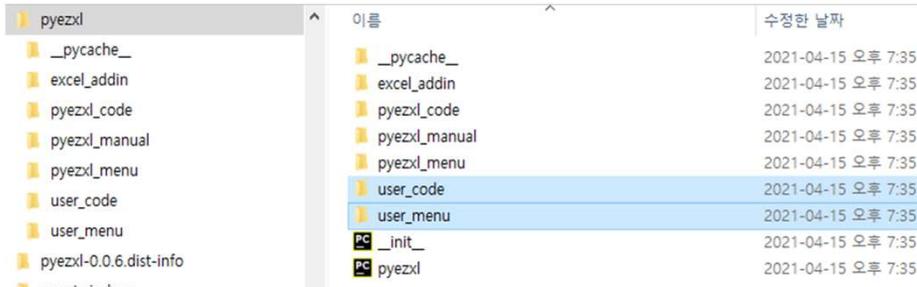
B1D3 => [1,2,3,4]

```
# 셀에 값을 쓰고 싶을 때 : write_cell_value (시트이름, 셀주소, 값)
# 영역에 값을 쓰고 싶을 때 : write_range_value (시트이름, range주소, 값)
# 현재 셀의 위치를 알고 싶을 때 : read_cell_address()
# 셀을 값이 있는 오른쪽 끝까지 이동시키고 싶을때 : move_right_end()
# 셀에 색깔을 넣고 싶을 때 : set_cell_color(시트이름, 셀주소, 값)
# 값을 복사하고 싶을때 : 값을 읽어온 후 -> 값을 쓰기
# 어디선가 읽어온 주소값을 사용하고 싶을때 : check_address_value(주소값)
```

4. 나의 코드를 만들어 사용하는 방법

■ 자신의 코드 작성

- ① 자신의 코드를 작성한다
- ② 이름은 차후에 모든 사람에게 공유가 가능하면서 고유의 이름을 위하여 "user_고유이름_버전_간단한설명.txt"으로 만들어서 사용하기를 권장한다
- ③ 만든 코드를 사용자 폴더(user_code)에 복사한다
- ④ 복사한 코드가 메뉴에 나올 수 있도록 메뉴화일에 만든 코드이름을 넣는다



이름	수정한 날짜
__pycache__	2021-04-15 오후 7:35
excel_addin	2021-04-15 오후 7:35
pyezxl_code	2021-04-15 오후 7:35
pyezxl_manual	2021-04-15 오후 7:35
pyezxl_menu	2021-04-15 오후 7:35
user_code	2021-04-15 오후 7:35
user_menu	2021-04-15 오후 7:35
__init__	2021-04-15 오후 7:35
pyezxl	2021-04-15 오후 7:35

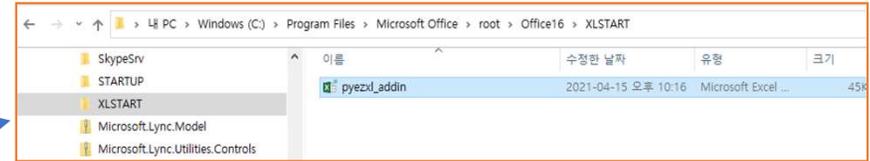
■ 자신의 코드 작성 (예)

- ① user_psj_01_sort_columns.py으로 코드 작성
- ② 만든 코드를 사용자 폴더(user_code)에 복사한다
- ③ 복사한 코드가 메뉴에 나올 수 있도록 user_menu부분을 수정한다

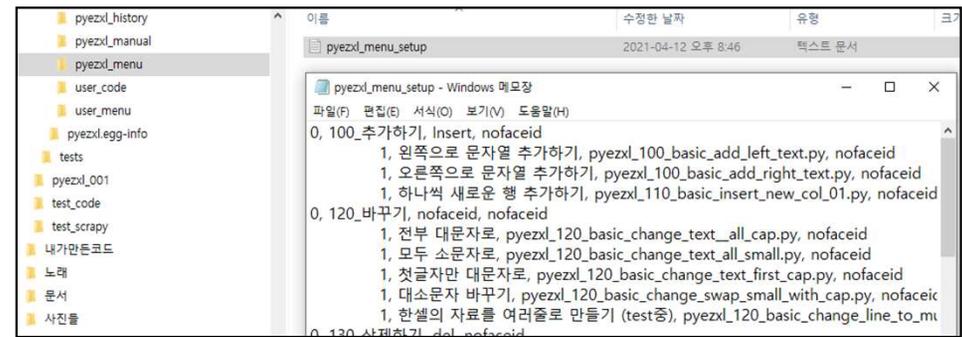
```
1 0,user_추가기능_1,del,nofaceid
2 →1,유저_테스트_11,user_test_11.py,nofaceid
3 →1,유저_테스트_12,user_test_12.py,nofaceid
4 0,user_추가기능_2,del,nofaceid
5 →1,유저_테스트_21,user_test_21.py,nofaceid
6 →1,유저_테스트_22,user_psj_01_sort_columns.py,nofaceid
```

5. 실행되는 순서

① 엑셀을 실행한다



② Xlstart에 들어있는 pyezxl_addin 파일이 실행된다

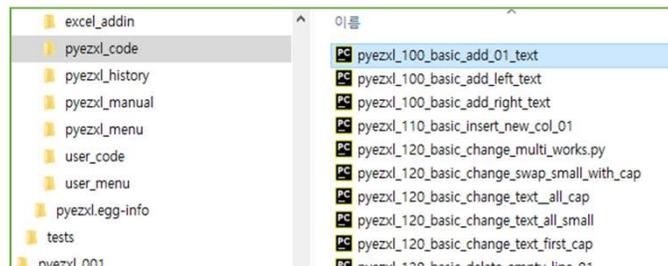


③ Pyezxli이 설치된 경로에서 pyezxl_menu와 user_menu를 읽어와서 엑셀의 추가메뉴를 만든다

④ 사용자가 엑셀의 추가메뉴를 누르면



⑤ 추가 메뉴가 만들어질때 연결이된 Python코드를 Pyezxli이 설치된 곳에서 찾아서 실행한다



6. 만든 이유

- 엑셀의 VBA는 사용하기 어려워 Python을 이용하여 쉽게 엑셀의 기능을 사용하도록 만들고 싶었다
- 이런 파이썬의 새로운 모듈을 만들어서 여러사람들이 쉽게 만들 수 있도록 하고 싶었다
- 또한 엑셀의 메뉴바를 만들어 기능을 쉽게 만들고 싶었다
- 누군가 만들어 놓은 좋은 코드를 쉽게 공유가 가능하도록 하고싶었다
- 공유는 버튼을 누르면 자동으로 실행되도록 하고싶었다(아직...)
- 자신만이 사용하고 싶은 코드는 따로 사용이 가능하도록 하고싶었다
- 다른나라의 언어로 편하게 사용가능하도록 변경 예정

7. 저작권 / 가격정책

* 저작권

이화일의 소스 코드 및 아이디어는 박상진과 그 가족들이 공통으로 만들어낸 창작프로그램으로 저작권이 보호된다

자신의 코드를 제외하고, 공식적으로 배포한 모든 코드와 프로그램은 저작자의 권한이다.

개인적인 용도로의 사용은 가능하며,

상업적용도로는 저작권자의 동의없이 일체의 사용을 금한다.

* 가격정책

등록용과 비등록용의 사용상 차이는 없으며,

등록용은 차후 개별 serial no를 별도로 송부 예정이며

등록비용은 향후 다른 프로젝트의 일부로 사용될 예정입니다

등록비용 : www.halmoney.com 방문 요망

Python

Pyezxl사용예

사용예

- * 실제 pyezxl_code안을 보면 실제 사용한 예들을 볼 수 있다
- * 여기서는 가장 기본이 되는 사용법을 한번 보도록 하는 것이다

- 1) 영역 [1,1,5,5]를 선택해보자
- 2) 선택한 영역에 색을 분홍색으로 하고
- 3) 숫자를 1부터 25까지 써보고
- 4) 2열에 새로운 열을 추가해 보고
- 5) 제일 첫번째 글자를 삭제해 본다

```
# -*- coding: utf-8 -*-  
  
import pyezxl  
excel = pyezxl.pyezxl("activeworkbook")  
activesheet_name = excel.read_activesheet_name()  
  
excel.set_range_select("",[1,1,5,5])  
color_no = excel.read_color_number("yellow")  
excel.set_range_color("",[1,1,5,5], color_no)  
  
[x_start, y_start, x_end, y_end] = [1,1,5,5]  
value=0  
for x in range(x_start, x_end + 1):  
    for y in range(y_start, y_end + 1):  
        value=value+1  
        excel.write_cell_value("", [x,y], value)  
  
excel.insert_line_y("",2)  
  
excel.write_cell_value("", [1, 1], "")
```

	A	B	C	D	E	F
1			2	3	4	5
2	6		7	8	9	10
3	11		12	13	14	15
4	16		17	18	19	20
5	21		22	23	24	25
6						25

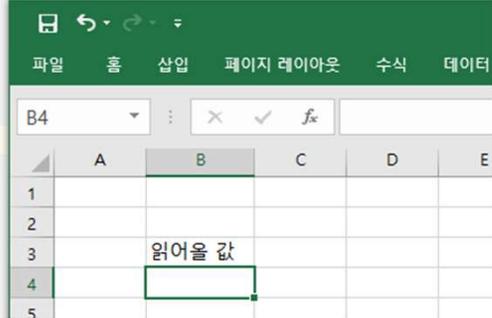
사용예

셀 (B3)의 값을 읽어오고 싶을 때

```
import pyezxl
excel=pyezxl.pyezxl()

print(excel.read_cell_value("",[3, 2]))
```

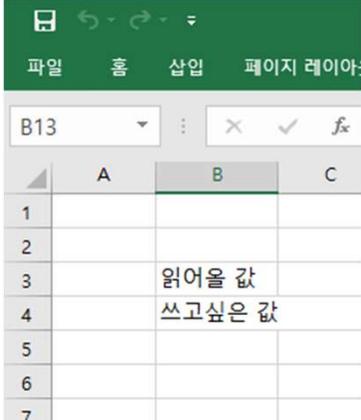
```
1 # -*- coding: utf-8 -*-
2
3
4 import pyeasyexcel
5 excel=pyeasyexcel.easyexcel()
6
7 print(excel.read_cell_value("",[3, 2]))
8
9
10
11
12
13
14
15
16
17
18
19
```



셀 (B4)에 값을 쓰고 싶을 때

```
import pyezxl
excel=pyezxl.pyezxl()

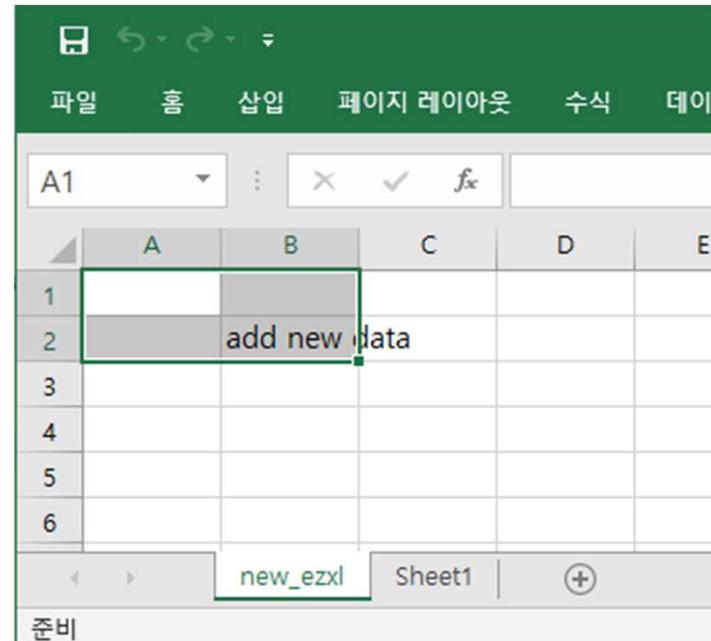
print(excel.read_cell_value("",[3, 2]))
excel.write_cell_value("",[4, 2], "쓰고싶은 값")
```



사용예

새로운시트를 하나 만든 시트의 이름을
“new_ezxl”로 만들고, 영역을 선택하는 것

```
# -*- coding: utf-8 -*-  
  
import pyezxl  
excel=pyezxl.pyezxl()  
  
excel.insert_sheet_new()  
excel.change_sheet_name("", "new_ezxl")  
  
excel.write_cell_value("",[2, 2], "add new data")  
  
range = "$A$1:$B$2"  
excel.select_sheet_range("", range)  
  
excel.read_range_select()
```



사용예 -3

연속된 자료중 같은 값을 삭제하는것

```
import pyezxl
excel = pyezxl.pyezxl("activeworkbook")
activesheet_name = excel.read_activesheet_name()

x_start, y_start, x_end, y_end= excel.read_range_select()

#선택한 영역중 연속된 같은자료 삭제
flag_same = 0
old_data=""
for y in range(y_start, y_end+1):
    for x in range(x_start, x_end+1):
        if flag_same == 0:
            cell_value = excel.read_cell_value(activesheet_name,[x, y])
            cell_value_down = excel.read_cell_value(activesheet_name,[x+1, y])
            if cell_value == cell_value_down:
                excel.write_cell_value(activesheet_name, [x+1, y], "")
                flag_same = 1
            else:
                flag_same = 0
```

	A
1	aaa
2	aaa
3	bbb
4	bbb
5	bbb
6	bbb
7	ccc
8	cc
9	d
10	d
11	dd
12	ddd
13	가나다
14	가나
15	사람
16	사랑



	A
1	aaa
2	
3	bbb
4	
5	
6	
7	ccc
8	cc
9	d
10	
11	dd
12	ddd
13	가나다
14	가나
15	사람
16	사랑