Python libraries

Andy Kim

COM (from Microsoft)

COM (Component Object Model) enables the code to access to external instances of other programs

```
import win23com.client


excel = win32com.client.Dispatch("Excel.Application")
excel.Visible = True
wb = excel.Workbooks.Add()
ws = wb.Worksheets("Sheet 1")
ws.Cells(1,1).Value = "hello world"
wb.SaveAs('C"\\Users\\andy.kim\\test.xlsx')
excel.quit()
```

- Pandas can make it easier to communicate with Excel
- Daesin / eBest

OCX (from Microsoft too)

OCX (Object Linking and Embedding Custom Contrl) can be accessed using QAxContainer Module from PyQt package

- Kiwoom

PyQt

PyQt is a GUI framework binding of Qt (written in C++)

```
import sys
from PyQt5.QtWidgets import *
```

```
app = QApplications(sys.argv)  # sys.argv contains path to current code
label = QPushButton("Quit")
label.show()
app.exec_()   # Event loop
```

Another example:

```
import sys
from PyQt5.QtWidgets import *

class MyWindow(QMainWindow):
    def __init__(self):
        super().__init__()  # calling 'init' of parent class; note no 'self'
        self.setWindowTitle("PyStock")
        self.setGeometry(300,300,300,400)

        btn1 = QPushButton("Click me", self)
        btn1.move(20, 20)
        btn1.clicked.connect(self.btn1_clicked)  # 'clicked' event is generated, so it needs to defi

    def btn1_clicked(self):
        QMesssageBox.about(self, "message", "clicked")



if __name__ == "__main__":
    app = QApplication(sys.argv)
    mywindow = MyWindow()
    mywindow.show()
    app.exec_()
```

Kiwoom calling

```
# ... inside of __init__(self) ...
    self.kiwoom = QAxWidget("KHOPENAPI.KHOpenAPICtrl.1")  # CLSID or ProgID to be found in Windows r
    self.kiwoom.dynamicCall("CommConnect()")
    ret = self.kiwoom.dynamicCall("GetConnectStatus()")
    self.kiwoom.OnEventConnect.connect(self.event_connect)
    account_num = self.kiwoom.dynamicCall("GetLoginInfo(QString)", ["ACCNO"])
    # "ACCNO" is input to QString It has to be the list format ["ACCNO"] even with only one element
```

```
# Event Handler of OnEventConnect
def event_connect(self, err_code):
    if err_code == 0:
        pass
```

QtDesigner is a GUI design tool

Pandas

Data structure library; mainly used structures are Series, DataFrame

- Series: Python list with index

```
from pandas import Series

a = Series([10, 20, 30], index=['a', 'b', 'c'])
b = Series([100, 200, 300], index=['c', 'b', 'a'])
```

- DataFrame: 2x2 Matrix with column and row index

```
from pandas import DataFrame

dict1 = {'a': [1, 2, 3]
         'b': [4, 5, 6]
         'c': [7, 8, 9]}

df1 = DataFrame(dict1, index=['x', 'y', 'z'])
```

Refer to manual for various operators of Series and DataFrame

pandas_datareader

pandas_datareader is a datareader that returns DataFrame. It supports numerous financial institutioins - refer to its documentation on the web

matplotlib

matplotlib is a Python 2D plotting library; similar to Matlab

```
import matplotlib.pyplot as plt
plt.plot(df1.index, df1['a'])
```

```
plt.show()
```

zipline

Zipline is an open-source algorithmic trading simulator written in Python - refer to doc