# 암호화폐 트레이딩 교육

Day 3 (2/2)

HTS 만들어 보기

Kangwuk Heo
calvin.heo@gmail.com

# Contents

**HTS 만들어 보기 3**

**SECTION 1: 실시간 호가 UI ..... 4**

**SECTION 2: 실시간 개요 UI ..... 17**

## Overview

암호화폐 트레이딩 교육 과정에서, 실습을 통해서, 트레이딩이 무엇인지, 트레이딩을 어떤 방식으로 구성하고 진행하는지를 배울 수 있습니다.

# HTS 만들어 보기

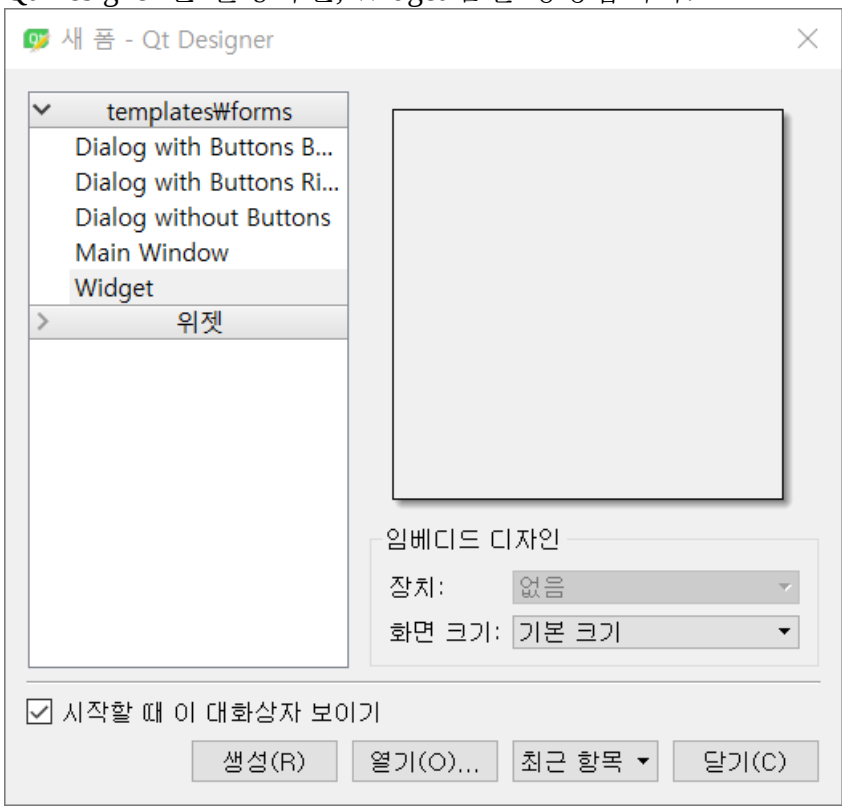
**Section 1** 은 실시간 호가 UI 을 만들어 봅니다.
**Section 2** 은 실시간 개요 UI 을 만들어 봅니다.

## Section 1: 실시간 호가 UI

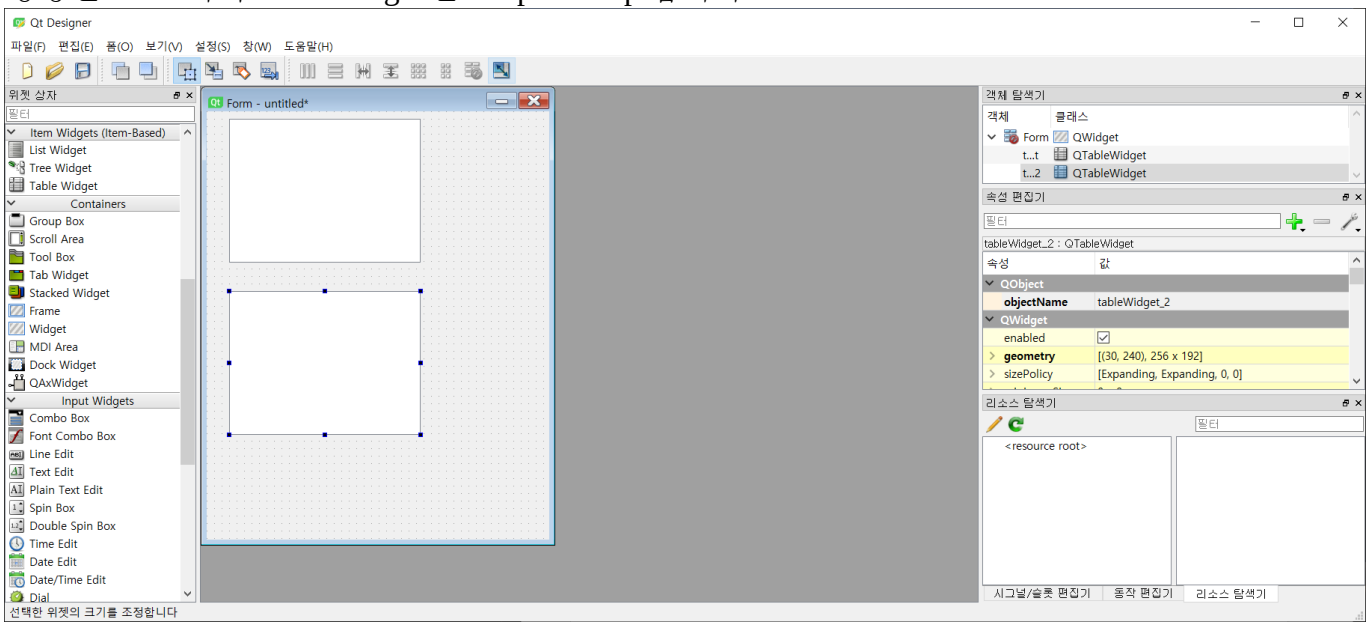
가상화폐 거래소의 거래소 페이지에 볼수 있는 매도호가와 매수호가의 리스트를 tablewidget 을 통해서 구성하고, QProgessBar 를 통해서 수량을 표시합니다.

__1. Qt Designer 에서 실시간 호가 정보를 보여주기 위한 UI 를 생성합니다.

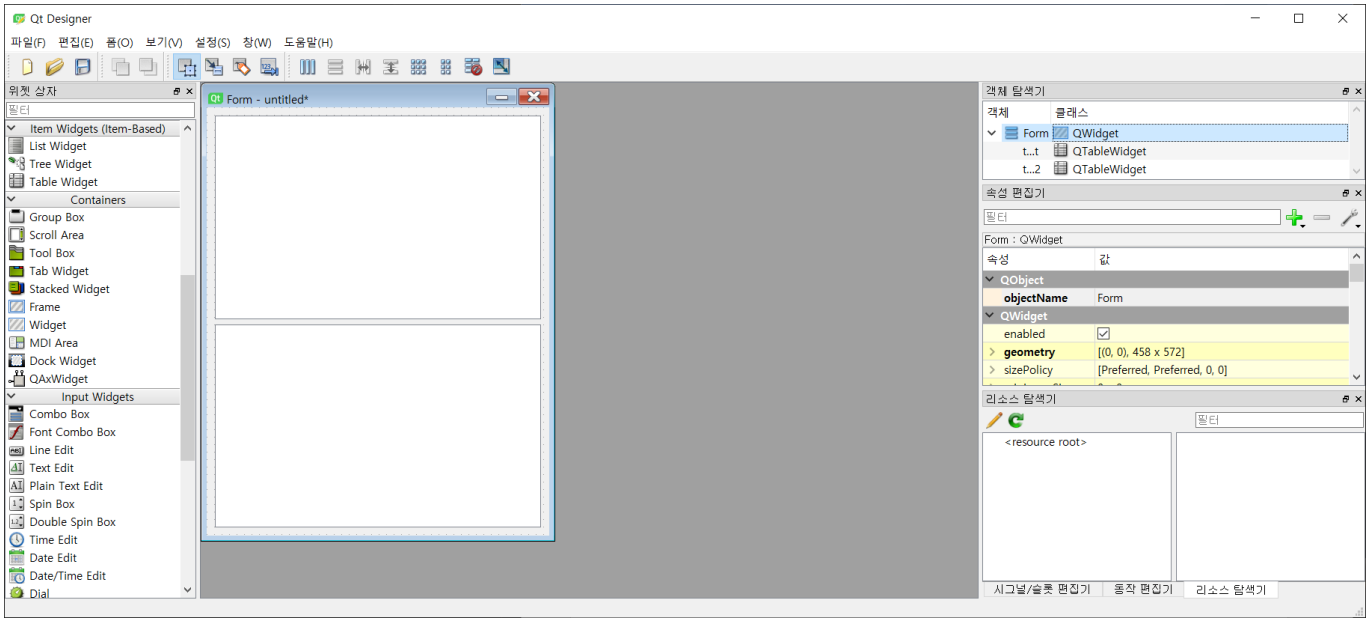
Qt Designer 를 실행하면, Widget 폼을 생성합니다.



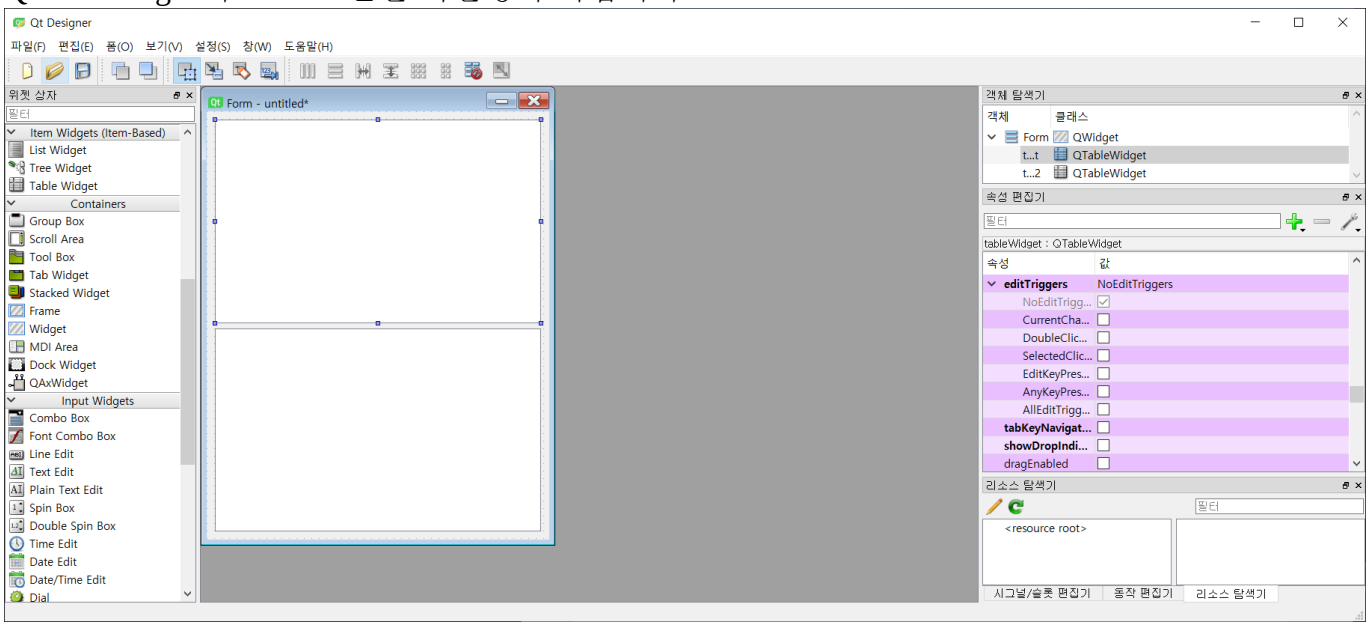

생성된 From 에서 TableWidget 를 Drap & Drop 합니다.



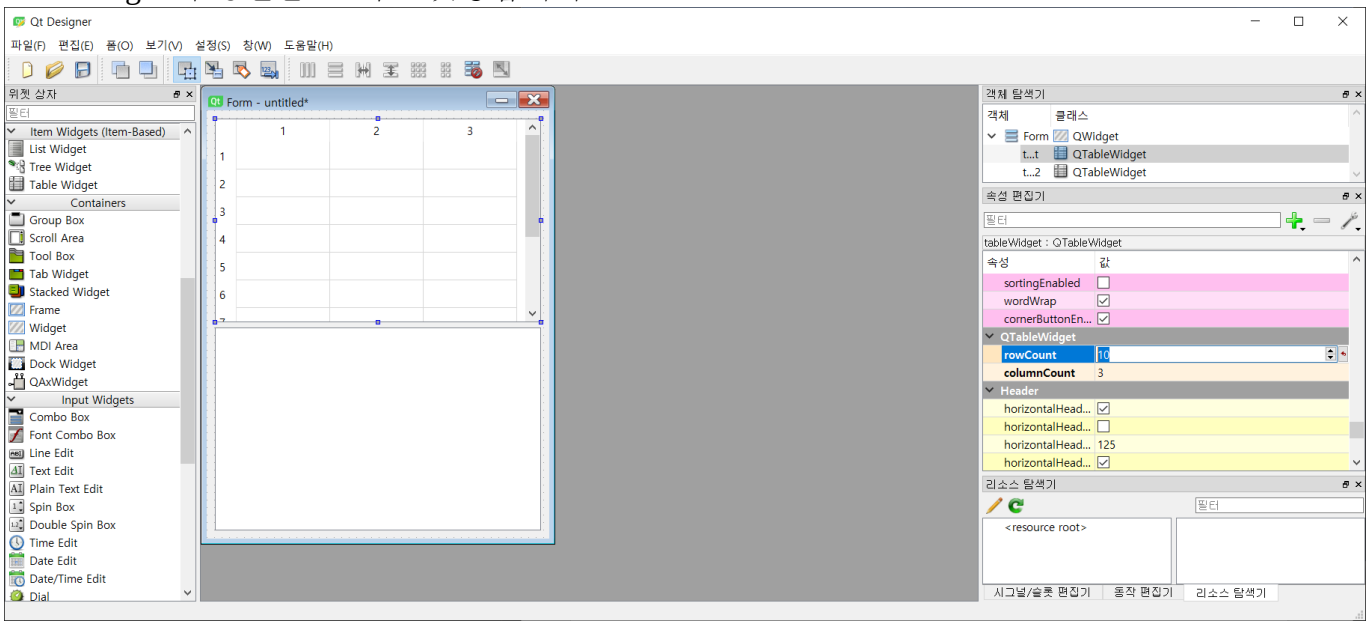
Form 을 배치에서 수직 배치로 전환합니다.



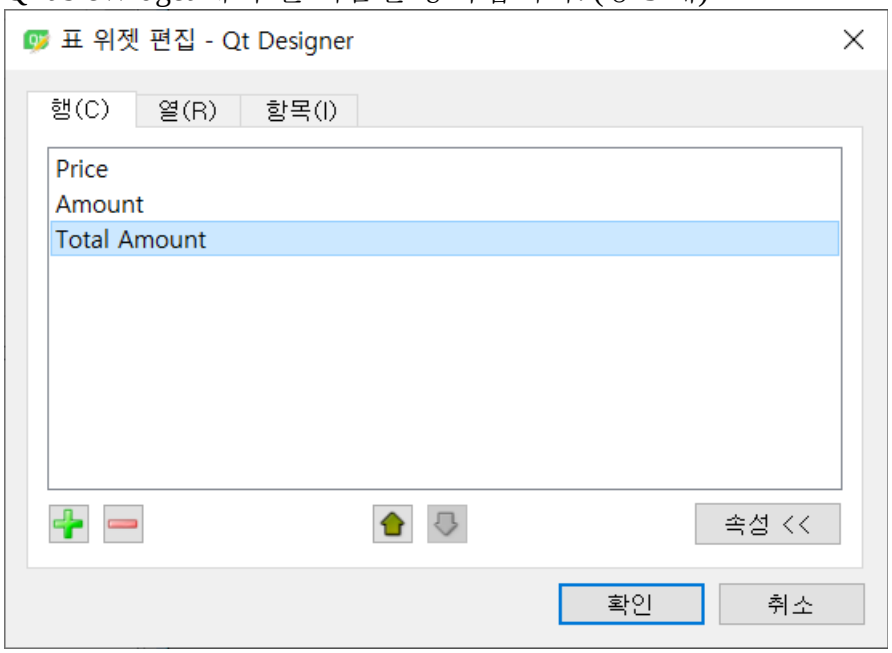

QTableWidget 의 Actions 들을 비활성화 시킵니다.



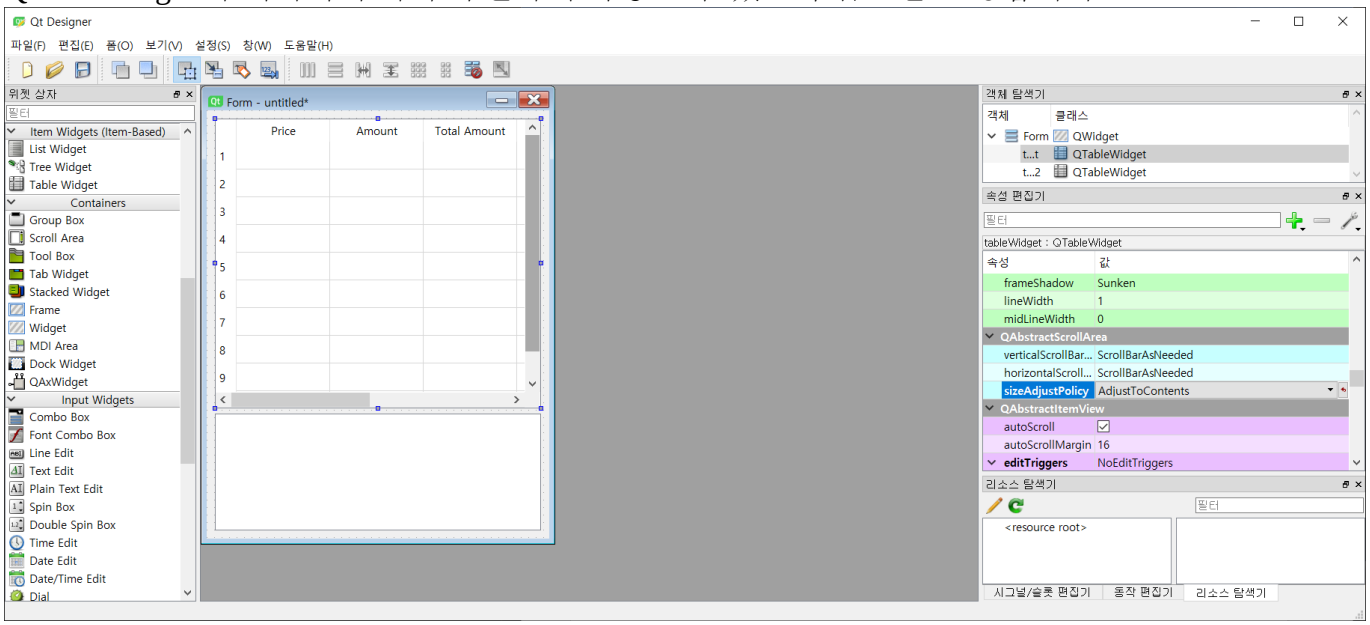
Tablewidget 의 행열을 10 개로 셋팅합니다.



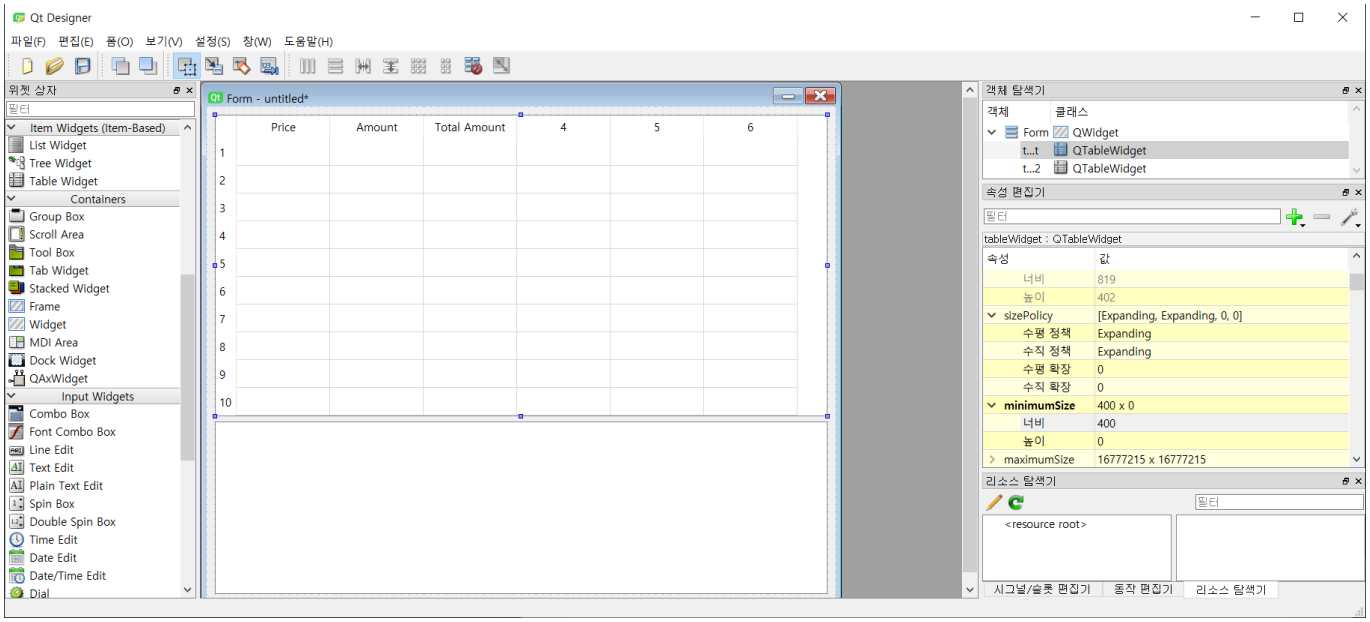

QTableWidget 에서 열 이름을 등록합니다. (총 3 개)



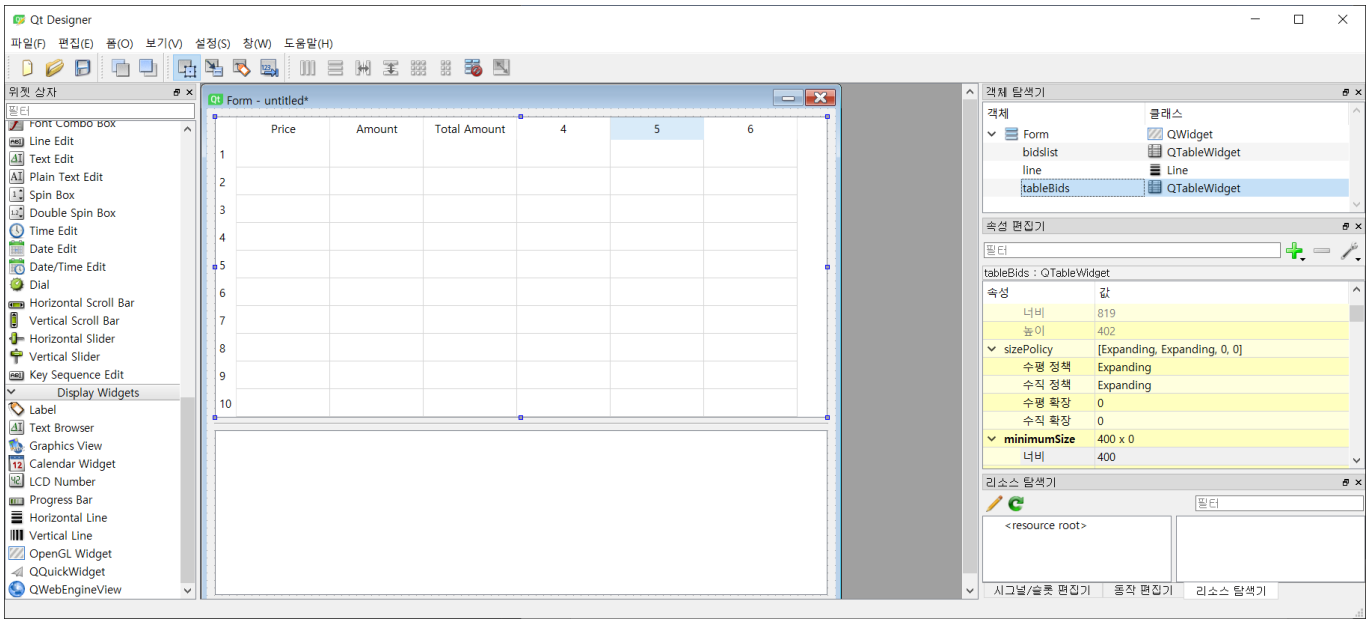
QTableWidget 이 데이터에 따라 유연하게 확장될 수 있도록 옵션을 변경합니다.




Form 의 여백을 0 로 셋팅합니다.

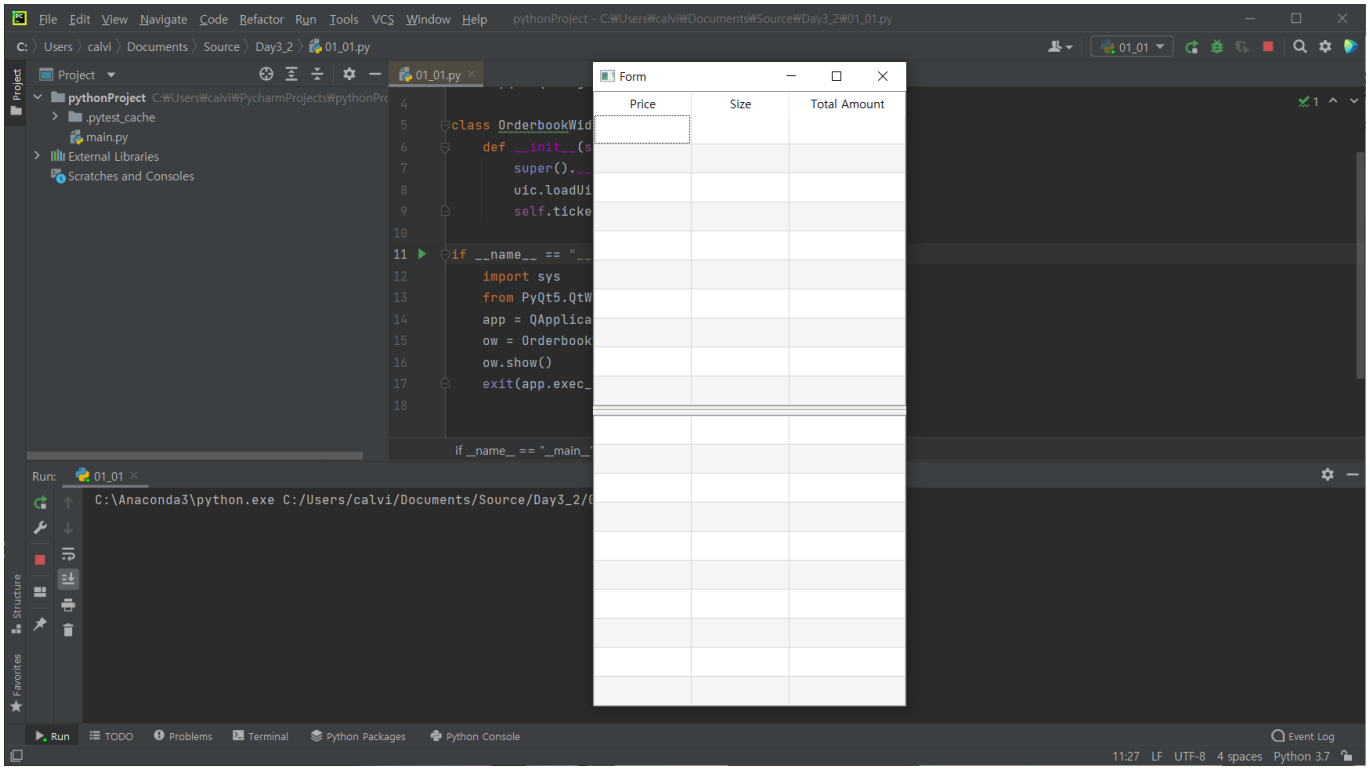


TableWidget 의 name 을 TableAsks 와 TableBids 로 변경합니다.




__2. 실시간 호가 UI 를 파이썬에서 정상적으로 연결되는지를 확인합니다. .

```
import sys
from PyQt5 import uic
from PyQt5.QtWidgets import QWidget

class OrderbookWidget(QWidget):
    def __init__(self, ticker="BTC"):
        super().__init__()
        uic.loadUi("orderbook.ui", self)
        self.ticker = ticker

if __name__ == "__main__":
    import sys
    from PyQt5.QtWidgets import QApplication
    app = QApplication(sys.argv)
    ow = OrderbookWidget()
    ow.show()
    exit(app.exec_())
```


File Edit View Navigate Code Refactor Run Tools VCS Window Help
pythonProject - C:₩Users₩calvi₩Documents₩Source₩Day3_2₩01_01.py
C: Users calvi Documents Source Day3_2 01_01.py
Project
pythonProject C:₩Users₩calvi₩PycharmProjects₩pythonPro
.pytest_cache
main.py
External Libraries
Scratches and Consoles
01_01.py
class OrderbookWid
def __init__(s
super().__
uic.loadUi
self.ticke
if __name__ == "__
import sys
from PyQt5.QtW
app = QApplica
ow = Orderbook
ow.show()
exit(app.exec_
if __name__ == "__main__"
Form
Price
Size
Total Amount
Run: 01_01
C:\Anaconda3\python.exe C:/Users/calvi/Documents/Source/Day3_2/
Run
TODO
Problems
Terminal
Python Packages
Python Console
Event Log
11:27 LF UTF-8 4 spaces Python 3.7

__3. QtableWidget 에 데이터를 저장하기 위한 공간을 생성하고, 거래 대금을 표현하기 위한 QprogressBar 를 생성합니다. .

```
import sys
from PyQt5 import uic
from PyQt5.QtWidgets import QWidget
# ----------------- 추 가 ------------------
from PyQt5.QtWidgets import QTableWidgetItem, QProgressBar
from PyQt5.QtCore import Qt
# ----------------------------------------

class OrderbookWidget(QWidget):
    def __init__(self, ticker="BTC"):
        super().__init__()
        uic.loadUi("order.ui", self)
        self.ticker = ticker


        for i in range(self.tableBids.rowCount()):
            # 매도호가
      item_0 = QTableWidgetItem(str(""))
            item_0.setTextAlignment(Qt.AlignRight | Qt.AlignVCenter)
            self.tableAsks.setItem(i, 0, item_0)

            item_1 = QTableWidgetItem(str(""))
            item_1.setTextAlignment(Qt.AlignRight | Qt.AlignVCenter)
            self.tableAsks.setItem(i, 1, item_1)

            item_2 = QProgressBar(self.tableAsks)
            item_2.setAlignment(Qt.AlignRight | Qt.AlignVCenter)
            item_2.setStyleSheet("""
                QProgressBar {background-color : rgba(0, 0, 0, 0%);border :
1}
                QProgressBar::Chunk {background-color : rgba(255, 0, 0,
50%);border : 1}
            """)
            self.tableAsks.setCellWidget(i, 2, item_2)

            # 매수호가
      item_0 = QTableWidgetItem(str(""))
            item_0.setTextAlignment(Qt.AlignRight | Qt.AlignVCenter)
            self.tableBids.setItem(i, 0, item_0)

            item_1 = QTableWidgetItem(str(""))
            item_1.setTextAlignment(Qt.AlignRight | Qt.AlignVCenter)
            self.tableBids.setItem(i, 1, item_1)

            item_2 = QProgressBar(self.tableBids)
            item_2.setAlignment(Qt.AlignRight | Qt.AlignVCenter)
            item_2.setStyleSheet("""
                QProgressBar {background-color : rgba(0, 0, 0, 0%);border :
1}
                QProgressBar::Chunk {background-color : rgba(0, 255, 0,
40%);border : 1}
            """)
            self.tableBids.setCellWidget(i, 2, item_2)
        # ----------------------------------------
```

```
if __name__ == "__main__":
    import sys
    from PyQt5.QtWidgets import QApplication
    app = QApplication(sys.argv)
    ow = OrderbookWidget()
    ow.show()
    exit(app.exec_())
```

매도호가 (tableAsk) 테이블에 열별 저장될 문자열 객체를 생성하고, 호가 잔량을 시각화 하기 위한 QprogressBar 객체를 생성하여, 출력되는 텍스트를 연계 작업을 진행합니다.

그리고 매수호가 (tableBids) 테이블에 연계를 위한 객체를 생성합니다.

__4. 가상화폐 거래소로부터 데이터를 가져오기 위한 로직을 추가합니다. .

```
class OrderbookWorker(QThread):
    dataSent = pyqtSignal(dict)

    def __init__(self, ticker):
        super().__init__()
        self.ticker = ticker
        self.alive = True

    def run(self):
        while self.alive:
            data  = pybithumb.get_orderbook(self.ticker, limit=10)
            time.sleep(0.05)
            self.dataSent.emit(data)

    def close(self):
        self.alive = False
```

가상화폐 거래소의 공개 API 를 호출하여(초당 20 번 호출), Orderbook 을 매수 / 매도정보를 각각 10 개씩 가져오는 로직을 추가합니다.

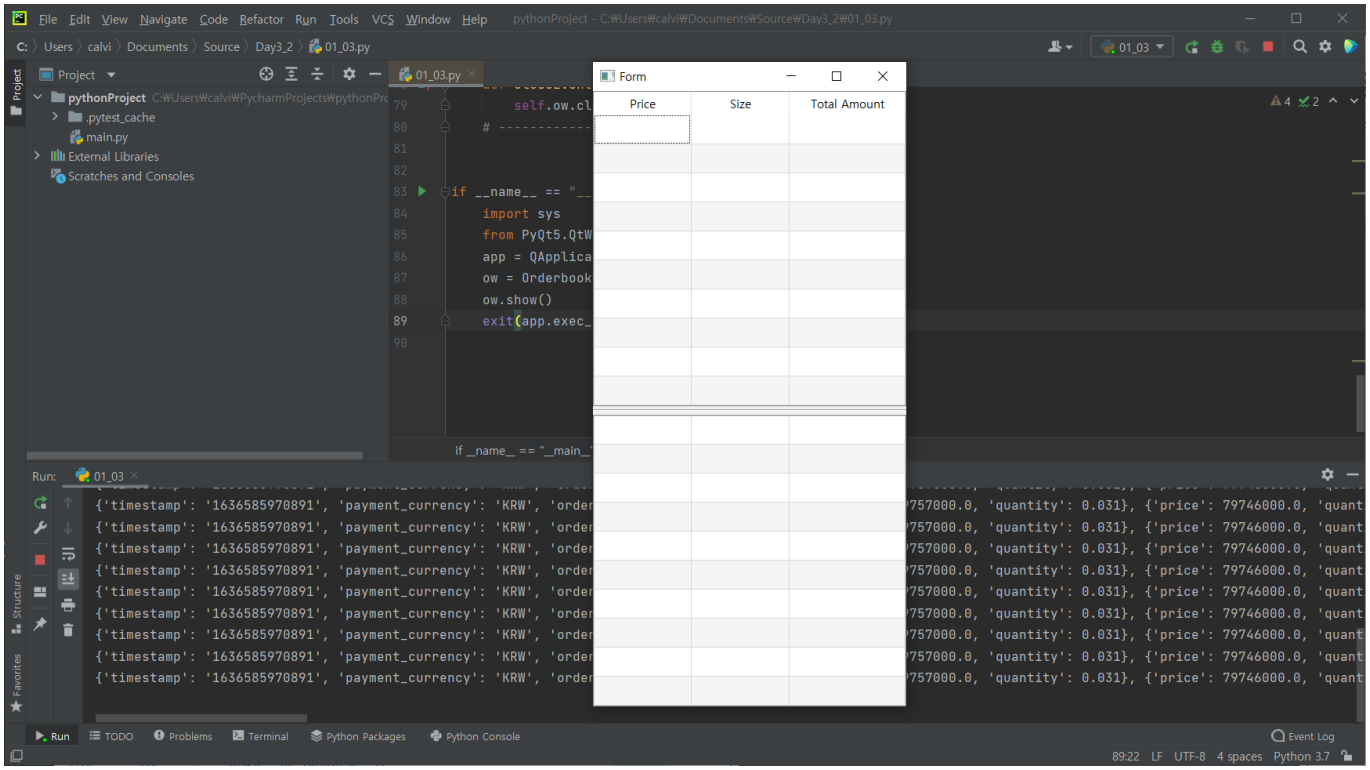
딕셔너리를 전달하는 dataSent signal 을 정의합니다.

데이터를 얻어오는 Qthread 를 실행합니다.

QThread 에서 시그널이 전송되면 updateDate 가 실행됩니다.

Qthread 의 종료를 처리하기 위해 Qwidget 의 메서드를 오버라이딩하여, 메인위젯이 종료될 때 closeEvent 메서드가 실행됩니다.

실행을 하게되면, console 에 Orderbook 정보가 보이게 됩니다.


File Edit View Navigate Code Refactor Run Tools VCS Window Help
pythonProject - C:₩Users₩calvi₩Documents₩Source₩Day3_2₩01_03.py
C: Users calvi Documents Source Day3_2 01_03.py
Project
pythonProject C:₩Users₩calvi₩PycharmProjects₩pythonPro
.pytest_cache
main.py
External Libraries
Scratches and Consoles
01_03.py
79 self.ow.cl
80 # ----------
81
82
83 if __name__ == "__
84 import sys
85 from PyQt5.QtW
86 app = QApplica
87 ow = Orderbook
88 ow.show()
89 exit(app.exec_
90
if __name__ == "__main__"
Form
Price
Size
Total Amount
Run: 01_03
{'timestamp': '1636585970891', 'payment_currency': 'KRW', 'order
757000.0, 'quantity': 0.031}, {'price': 79746000.0, 'quant
Run
TODO
Problems
Terminal
Python Packages
Python Console
Event Log
89:22 LF UTF-8 4 spaces Python 3.7

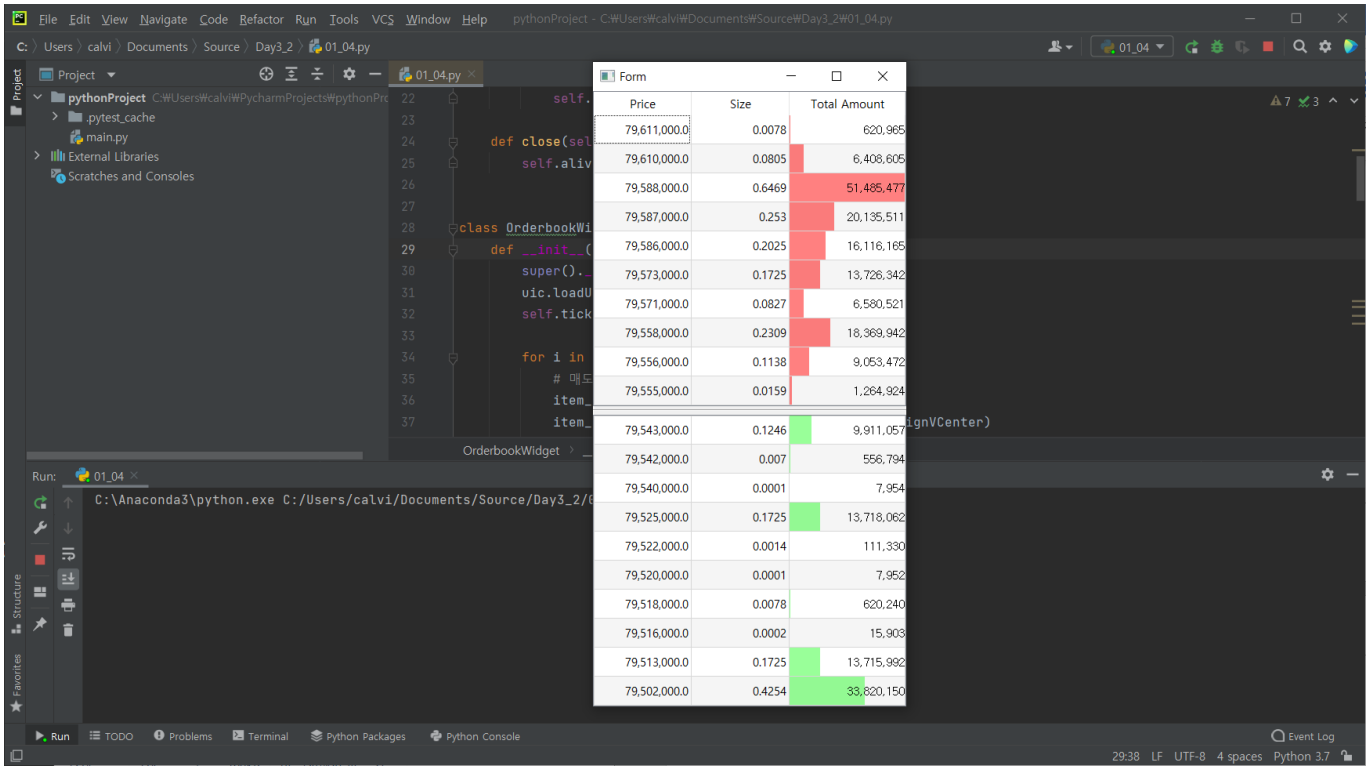
__5. 가상화폐 거래소로부터 가져오는 Orderbook 정보를 UI 에 연동하는 작업을 진행합니다.

```
def updateData(self, data):

    tradingBidValues = [ ]
    for v in data['bids']:
        tradingBidValues.append(int(v['price'] * v['quantity']))
    tradingAskValues = [ ]
    for v in data['asks'][::-1]:
        tradingAskValues.append(int(v['price'] * v['quantity']))
    maxtradingValue = max(tradingBidValues + tradingAskValues)

    for i, v in enumerate(data['asks'][::-1]):
        item_0 = self.tableAsks.item(i, 0)
        item_0.setText(f"{v['price']:,}")
        item_1 = self.tableAsks.item(i, 1)
        item_1.setText(f"{v['quantity']:,}")
        item_2 = self.tableAsks.cellWidget(i, 2)
        item_2.setRange(0, maxtradingValue)
        item_2.setFormat(f"{tradingAskValues[i]:,}")
        item_2.setValue(tradingAskValues[i])

    for i, v in enumerate(data['bids']):
        item_0 = self.tableBids.item(i, 0)
        item_0.setText(f"{v['price']:,}")
        item_1 = self.tableBids.item(i, 1)
        item_1.setText(f"{v['quantity']:,}")
        item_2 = self.tableBids.cellWidget(i, 2)
        item_2.setRange(0, maxtradingValue)
        item_2.setFormat(f"{tradingBidValues[i]:,}")
        item_2.setValue(tradingBidValues[i])
```

로직상에 UI 에 데이터를 전달하는 역할을 담당하는 updateData 메소드에 로직을 추가합니다.

가격과 수량을 곱한 총액을 리스트에 추가한 뒤에 전체 데이터의 최대값을 계산해 maxTradingValue 변수에 저장하고 호가와 수량을 차례로 출력합니다.

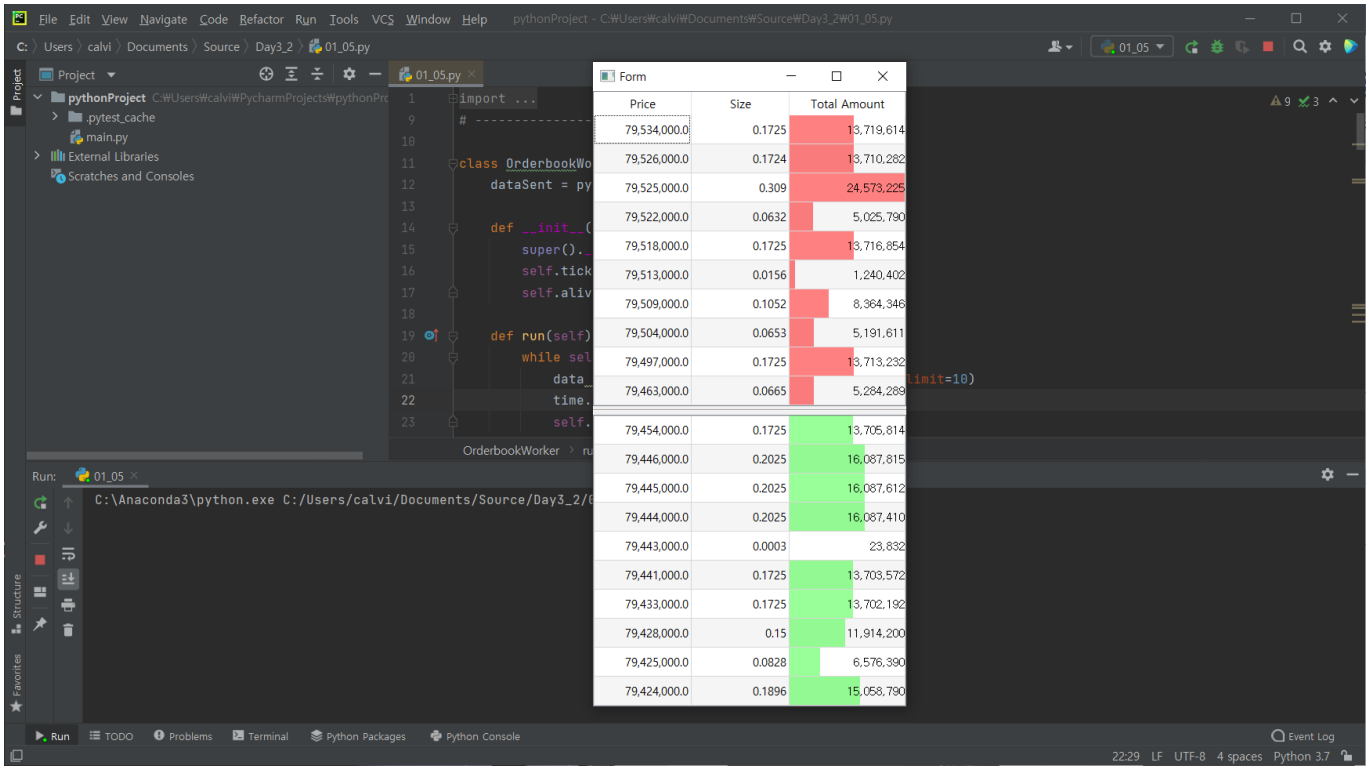
그리고 총액의 최대가를 100%로 설정하고 현재가를 QprogressBar 에 출력합니다.




__6. 실시간 호가 UI 를 파이썬에서 정상적으로 연결되는지를 확인합니다. .

```
import sys
from PyQt5 import uic
from PyQt5.QtWidgets import QWidget

class OrderbookWidget(QWidget):
    def __init__(self, ticker="BTC"):
        super().__init__()
        uic.loadUi("orderbook.ui", self)
        self.ticker = ticker

if __name__ == "__main__":
    import sys
    from PyQt5.QtWidgets import QApplication
    app = QApplication(sys.argv)
    ow = OrderbookWidget()
    ow.show()
    exit(app.exec_())
```

Qprogressive Bar 를 부드럽게 동작할 수 있도록 Animation 기능을 추가합니다.




## 연습문제

BTC 가 아닌 다른 가상화폐 (XRP, ETC, ETH)를 순차적으로 Ticker 정보를 변경하며, 호가정보를 확인해 봅니다.

## Section 2: 실시간 개요 UI

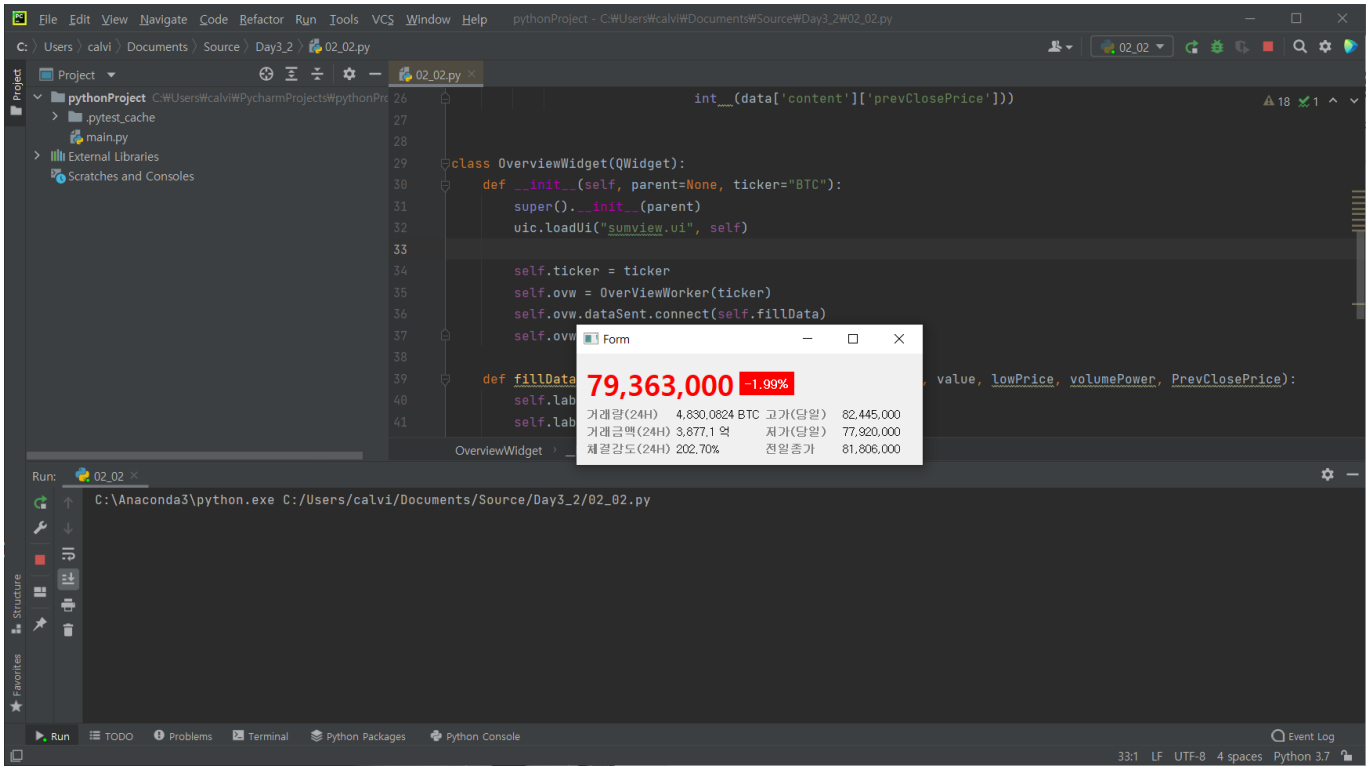
가상화폐 거래소 웹페이지에서 가상화폐의 전반적인 가격정보를 요약해서 보여주는 영역이 제공됩니다.

__1. Qt Designer 를 통해서 가상화폐의 실시간 개요정보 UI 를 만듭니다.

__2. 생성한 UI 를 파이썬에서 호출하는 로직을 실행합니다. .

```
import sys
from PyQt5 import uic
from PyQt5.QtWidgets import QWidget

class OverviewWidget(QWidget):
    def __init__(self, parent=None, ticker="BTC"):
        super().__init__(parent)
        uic.loadUi("sumview.ui", self)

if __name__ == "__main__":
    import sys
    from PyQt5.QtWidgets import QApplication
    app = QApplication(sys.argv)
    ob = OverviewWidget()
    ob.show()
    exit(app.exec_())
```

정상적으로 UI 이 보이는지를 확인합니다.

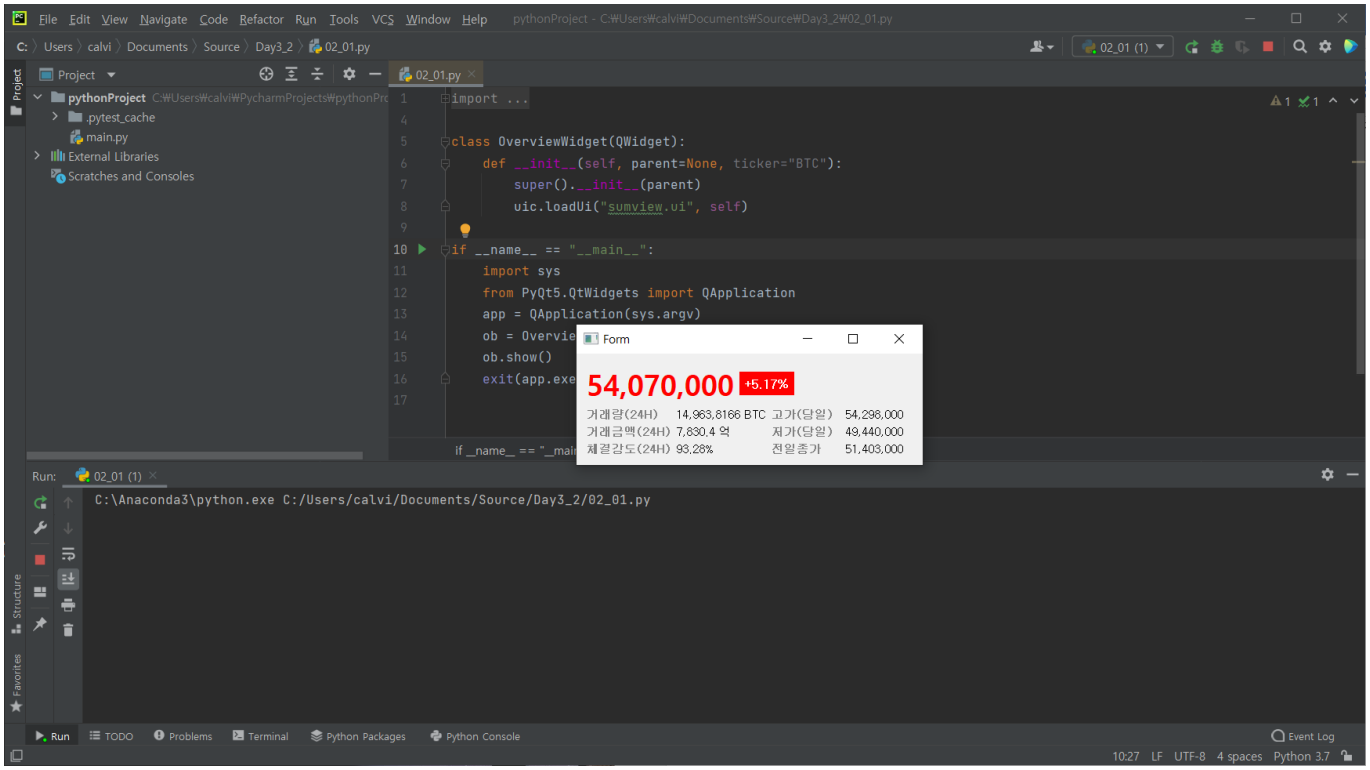



__3. OverviewWorker 를 생성하고 시그널에 연결한 슬롯을 정의하고, 전달된 데이터를 Label 에 출력합니다.

```
import sys
from PyQt5 import uic
from PyQt5.QtWidgets import QWidget
from PyQt5.QtCore import Qt, QThread, pyqtSignal
from pybithumb import WebSocketManager

class OverViewWorker(QThread):
    dataSent = pyqtSignal(int, float, float, int, float, int, float, int)

    def __init__(self, ticker):
        super().__init__()
        self.ticker = ticker
        self.alive = True

    def run(self):
        wm = WebSocketManager("ticker", [f"{self.ticker}_KRW"], ["24H"])
        while self.alive:
            data = wm.get()
            self.dataSent.emit(int   (data['content']['closePrice'     ]),
                               float(data['content']['chgRate'       ]),
                               float(data['content']['volume'        ]),
                               int   (data['content']['highPrice'    ]),
                               float(data['content']['value'         ]),
                               int   (data['content']['lowPrice'     ]),
                               float(data['content']['volumePower'   ]),
                               int   (data['content']['prevClosePrice']))
```

```
class OverviewWidget(QWidget):
    def __init__(self, parent=None, ticker="BTC"):
        super().__init__(parent)
        uic.loadUi("sumview.ui", self)

        self.ticker = ticker
        self.ovw = OverViewWorker(ticker)
        self.ovw.dataSent.connect(self.fillData)
        self.ovw.start()

    def fillData(self, currPrice, chgRate, volume, highPrice, value,
lowPrice, volumePower, PrevClosePrice):
        self.label_1.setText(f"{currPrice:,}")
        self.label_2.setText(f"{chgRate:+.2f}%")
        self.label_4.setText(f"{volume:,.4f} {self.ticker}")
        self.label_6.setText(f"{highPrice:,}")
        self.label_8.setText(f"{value/100000000:,.1f} 억")
        self.label_10.setText(f"{lowPrice:,}")
        self.label_12.setText(f"{volumePower:.2f}%")
        self.label_14.setText(f"{PrevClosePrice:,}")


if __name__ == "__main__":
    import sys
    from PyQt5.QtWidgets import QApplication
    app = QApplication(sys.argv)
    ob = OverviewWidget()
    ob.show()
    exit(app.exec_())
```

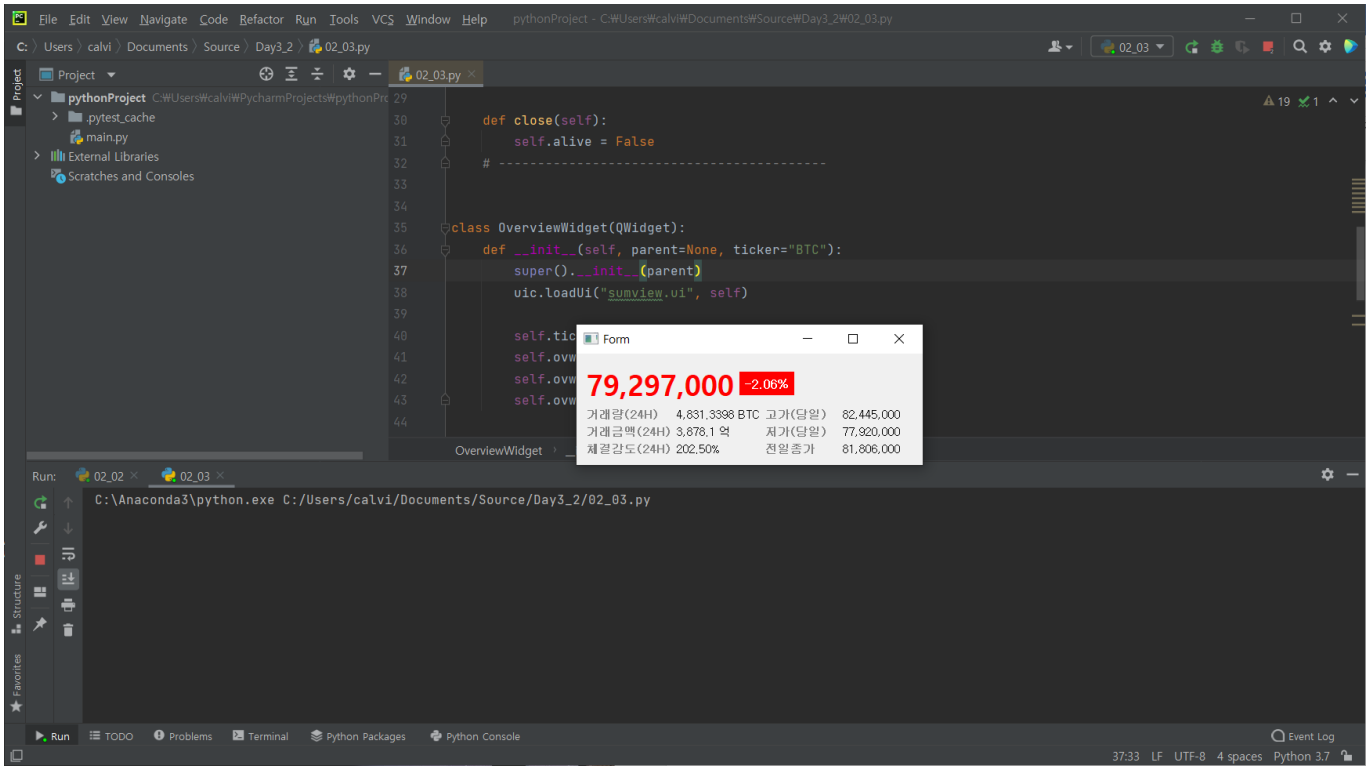
MID 는 자정기준으로 현재가와 등락률 표시합니다.




__4. Qthread 의 종료 처리를 위해 로직을 추가하고, 실시간 현재가와 호가창 관련 close 이벤트를 Overridng 처리를 합니다.

```
class OverViewWorker(QThread):
    dataSent = pyqtSignal(int, float, float, int, float, int, float, int)

    def __init__(self, ticker):
        super().__init__()
        self.ticker = ticker
        self.alive = True

    def run(self):
        wm = WebSocketManager("ticker", [f"{self.ticker}_KRW"], ["24H"])
        while self.alive:
            data = wm.get()
            self.dataSent.emit(int   (data['content']['closePrice'     ]),
                               float(data['content']['chgRate'       ]),
                               float(data['content']['volume'        ]),
                               int   (data['content']['highPrice'    ]),
                               float(data['content']['value'         ]),
                               int   (data['content']['lowPrice'     ]),
                               float(data['content']['volumePower'   ]),
                               int   (data['content']['prevClosePrice']))
    # ------------------------ -----------------
        wm.terminate()

    def close(self):
        self.alive = False
    # ----------------------------------------
```

```python
class OverviewWidget(QWidget):
    def __init__(self, parent=None, ticker="BTC"):
        super().__init__(parent)
        uic.loadUi("sumview.ui", self)

        self.ticker = ticker
        self.ovw = OverViewWorker(ticker)
        self.ovw.dataSent.connect(self.fillData)
        self.ovw.start()

    # ----------------------------------------
    def closeEvent(self, event):
        self.ovw.close()
     # ----------------------------------------
```



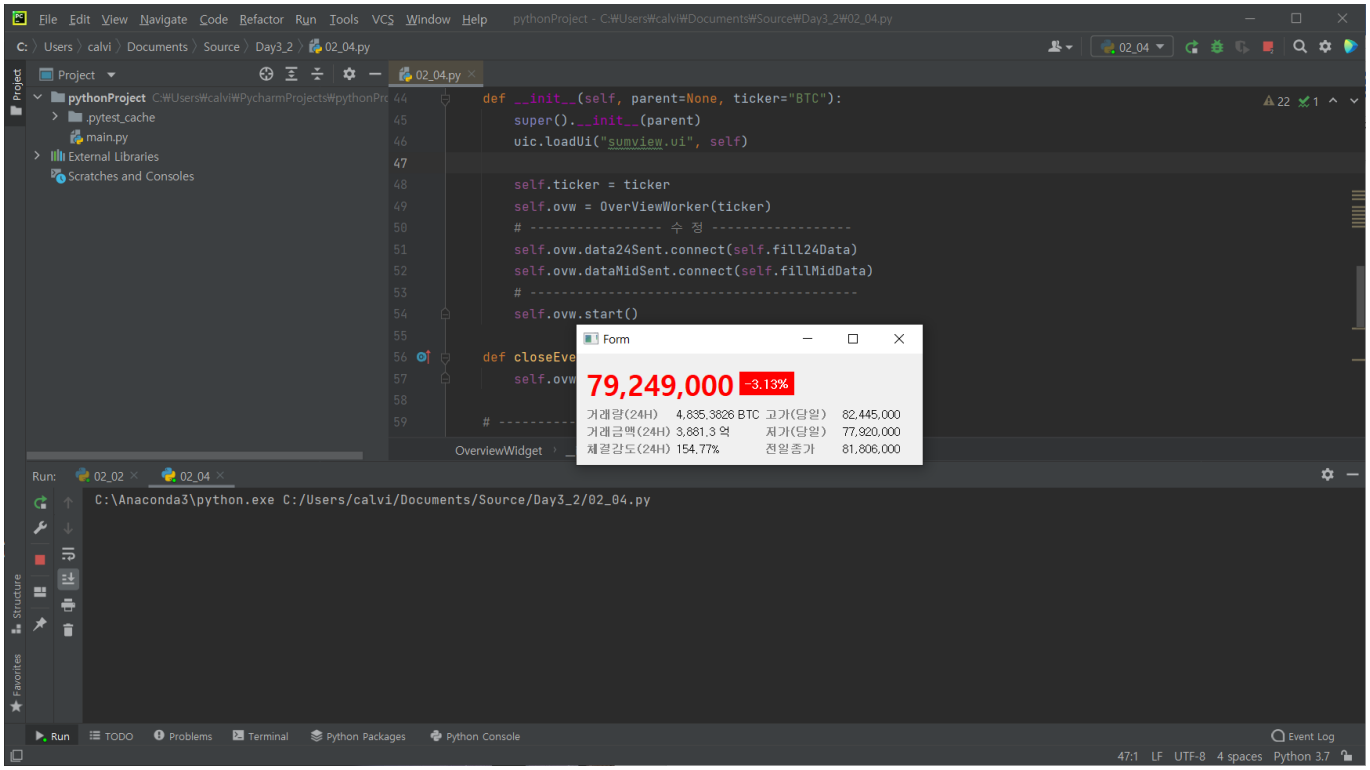
__5. 웹소켓의 Ticktype 파라미터를 수정해서[“24H”, “MD”]를 모두 구독하는 절차를 추가하고, 각각 별도의 Signal 을 통해 처리하는 로직을 추가합니다.

.

```
class OverViewWorker(QThread):
    # ----------------- --- ------------------
    data24Sent = pyqtSignal(int, float, int, float, int, int)
    dataMidSent = pyqtSignal(int, float, float)
    # ----------------------------------------

    def __init__(self, ticker):
        super().__init__()
        self.ticker = ticker
        self.alive = True

    def run(self):
        # ----------------- ---- ------------------
        wm = WebSocketManager("ticker", [f"{self.ticker}_KRW"], ["24H",
"MID"])
        while self.alive:
            data = wm.get()

            if data['content']['tickType'] == "MID":
                self.dataMidSent.emit(int
(data['content']['closePrice'    ]),

float(data['content']['chgRate'       ]),

float(data['content']['volumePower'   ]))
            else:
                self.data24Sent.emit(int
(data['content']['closePrice'    ]),

float(data['content']['volume'        ]),
                                     int
(data['content']['highPrice'     ]),

float(data['content']['value'         ]),
                                     int
(data['content']['lowPrice'      ]),
                                     int
(data['content']['prevClosePrice']))
        # ----------------------------------------

        wm.terminate()

    def close(self):
        self.alive = False
```

```python
class OverviewWidget(QWidget):
    def __init__(self, parent=None, ticker="BTC"):
        super().__init__(parent)
        uic.loadUi("sumview.ui", self)

        self.ticker = ticker
        self.ovw = OverViewWorker(ticker)
        # ----------------- 수정 ------------------
        self.ovw.data24Sent.connect(self.fill24Data)
        self.ovw.dataMidSent.connect(self.fillMidData)
        # ----------------------------------------
        self.ovw.start()

    def closeEvent(self, event):
        self.ovw.close()


    # ----------------- --- ------------------
    def fill24Data(self, currPrice, volume, highPrice, value, lowPrice,
PrevClosePrice):
        self.label_1.setText(f"{currPrice:,}")
        self.label_4.setText(f"{volume:,.4f} {self.ticker}")
        self.label_6.setText(f"{highPrice:,}")
        self.label_8.setText(f"{value/100000000:,.1f} 억")
        self.label_10.setText(f"{lowPrice:,}")
        self.label_14.setText(f"{PrevClosePrice:,}")

    def fillMidData(self, currPrice, chgRate, volumePower):
        self.label_1.setText(f"{currPrice:,}")
        self.label_2.setText(f"{chgRate:+.2f}%")
        self.label_12.setText(f"{volumePower:.2f}%")
    # ----------------------------------------
```

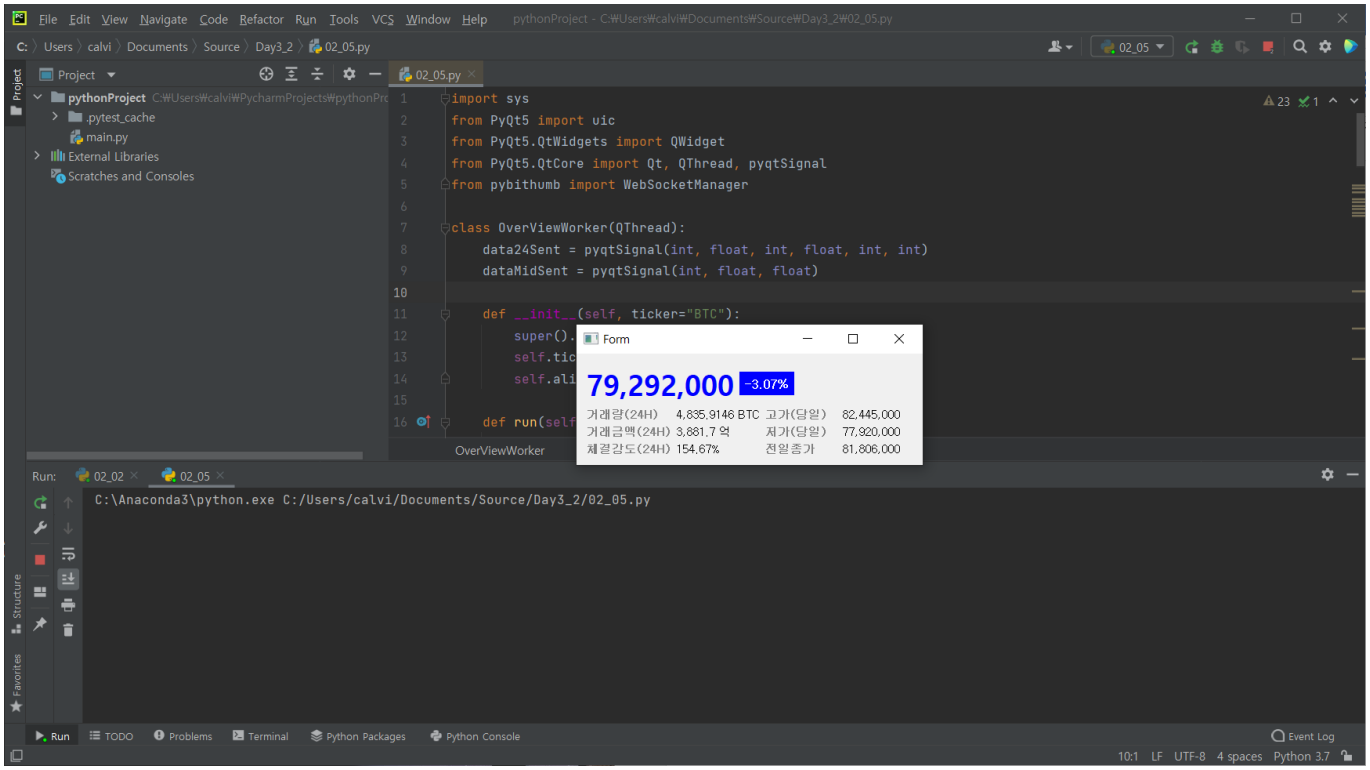
24H 이벤트와 MID 이벤트를 위한 시그널을 정의하고, 모두 구독하고,
웹서버가 전송한 데이터가 MID 일 경우, 현재가는 최신정보를 표현하기 위해 양쪽에 다 추가합니다.


File Edit View Navigate Code Refactor Run Tools VCS Window Help
pythonProject - C:₩Users₩calvi₩Documents₩Source₩Day3_2₩02_04.py
C: Users calvi Documents Source Day3_2 02_04.py
Project
pythonProject C:₩Users₩calvi₩PycharmProjects₩pythonPro
.pytest_cache
main.py
External Libraries
Scratches and Consoles
02_04.py
def __init__(self, parent=None, ticker="BTC"):
super().__init__(parent)
uic.loadUi("sumview.ui", self)
self.ticker = ticker
self.ovw = OverViewWorker(ticker)
# ---------- 수 정 ----------
self.ovw.data24Sent.connect(self.fill24Data)
self.ovw.dataMidSent.connect(self.fillMidData)
# ----------
self.ovw.start()
def closeEve
self.ovw
# ----------
OverviewWidget
Form
79,249,000
-3.13%
거래량(24H) 4,835.3826 BTC
고가(당일) 82,445,000
거래금액(24H) 3,881.3 억
저가(당일) 77,920,000
체결강도(24H) 154.77%
전일종가 81,806,000
Run: 02_02 02_04
C:\Anaconda3\python.exe C:/Users/calvi/Documents/Source/Day3_2/02_04.py
Run TODO Problems Terminal Python Packages Python Console
Event Log
47:1 LF UTF-8 4 spaces Python 3.7

__6. 현재가와 등락률에 따른 구분을 color 를 반영하여 처리합니다. (하한가 : 파란색, 상한가 : 빨간색)

```
class OverviewWidget(QWidget):
    def __init__(self, parent=None, ticker="BTC"):
        super().__init__(parent)
        uic.loadUi("sumview.ui", self)

        self.ticker = ticker
        self.ovw = OverViewWorker(ticker)
        self.ovw.data24Sent.connect(self.fill24Data)
        self.ovw.dataMidSent.connect(self.fillMidData)
        self.ovw.start()

    def closeEvent(self, event):
        self.ovw.close()

    def fill24Data(self, currPrice, volume, highPrice, value, lowPrice,
PrevClosePrice):
        self.label_1.setText(f"{currPrice:,}")
        self.label_4.setText(f"{volume:,.4f} {self.ticker}")
        self.label_6.setText(f"{highPrice:,}")
        self.label_8.setText(f"{value/100000000:,.1f} 억")
        self.label_10.setText(f"{lowPrice:,}")
        self.label_14.setText(f"{PrevClosePrice:,}")
        # ----------------- --- ------------------
        self.__updateStyle()
        # ----------------------------------------

    def fillMidData(self, currPrice, chgRate, volumePower):
        self.label_1.setText(f"{currPrice:,}")
        self.label_2.setText(f"{chgRate:+.2f}%")
        self.label_12.setText(f"{volumePower:.2f}%")
        # ----------------- ---------------------
        self.__updateStyle()
        # ----------------------------------------


    # ----------------- --- ------------------
    def __updateStyle(self):
        if '-' in self.label_2.text():
            self.label_1.setStyleSheet("color:blue;")
            self.label_2.setStyleSheet("background-color:blue;color:white")
        else:
            self.label_1.setStyleSheet("color:red;")
            self.label_2.setStyleSheet("background-color:red;color:white")
    # ----------------------------------------
```

스타일 업데이트 기능을 통해서 적용하며, _updateStyle() 메소드를 통해서 적용합니다.
Label2 에서 등락률이 표시되며, setStyleSheet 메소드를 통해서 적용됩니다.




## 연습문제

BTC 가 아닌 다른 가상화폐 (XRP, ETC, ETH)를 순차적으로 Ticker 정보를 변경하며, 가상화폐의 개요정보를 확인해 봅니다.