Spring Day 6

▼ Spring

File Upload

1. **실제 서버 경로 찾기**

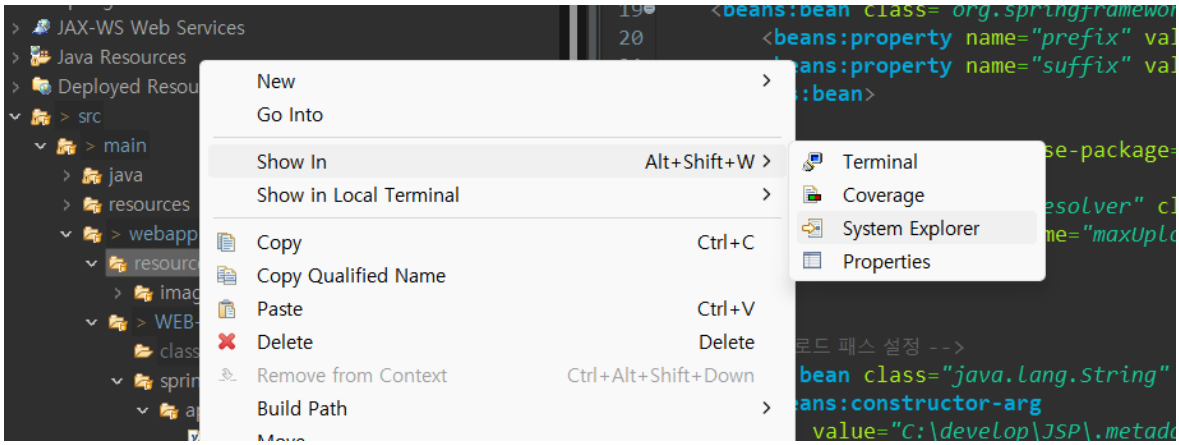
```
request.getSession().getServletContext().getRealPath("/")
```

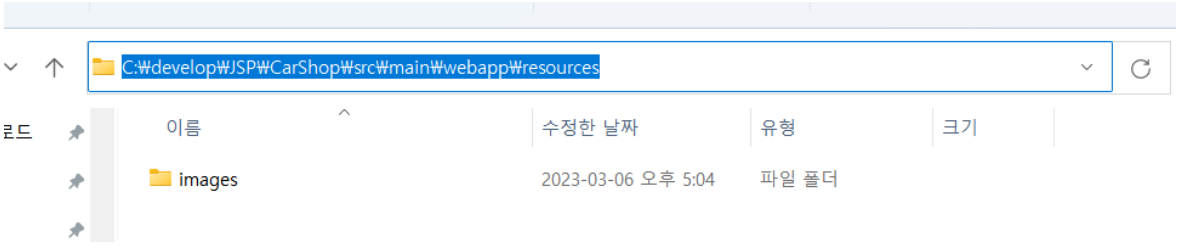
C:₩develop₩JSP₩.metadata₩.plugins₩org.eclipse.wst.server.core₩tmp0₩wtpwebapps₩CarShop₩

2. **servlet-context.xml**

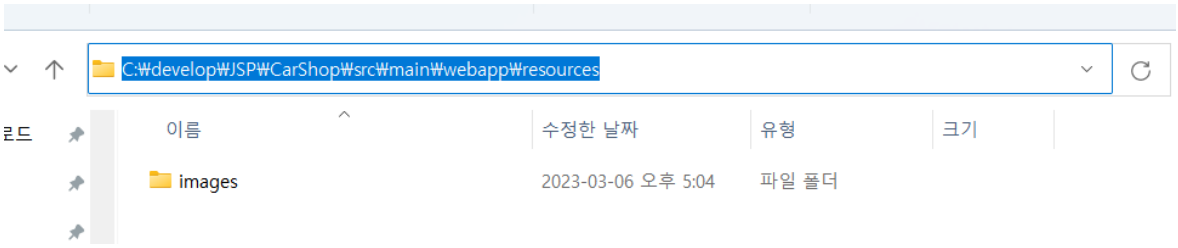
```
<!-- 업로드 패스 설정 -->
  <beans:bean class="java.lang.String" id="uploadPath">
    <beans:constructor-arg
        value="C:\develop\JSP\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\CarShop\resources" /> <!-- 호스팅 할 때
     <!--value="C:\develop\JSP\CarShop\src\main\webapp\resources" />--> <!-- 로컬에서 작업할 때 -->
  </beans:bean>

  <!-- 일반 파일 업로드 경로 -->
  <resources mapping="/images/**" location="/resources/images/" />
```

System Explorer(로컬 작업 경로 찾기)





로컬 작업 경로



실제 운영 서버의 작업 경로(Cafe24 등의 서버에서는 주소형태가 운영체제에 따라 달라진다.)

C:₩develop₩JSP₩.metadata₩.plugins₩org.eclipse.wst.server.core₩tmp0₩wtpwebapps₩CarShop₩

3. **Controller**

```
  @Resource(name = "uploadPath")
  private String uploadPath;

  @PostMapping("/admin/add")
  public String submitAddNewCar(@ModelAttribute("NewCar") CarDTO car, HttpServletRequest request) {

    MultipartFile carimage = car.getCarimage();
    String saveName = carimage.getOriginalFilename();

//    String uploadpath = request.getRealPath("/resources/images");

//    File saveFile = new File("C:\\upload", saveName);
    File saveFile = new File(uploadPath + "\\images", saveName);
//    System.out.println(saveFile.getPath());


    if (carimage != null && !carimage.isEmpty()) {
      try {
        carimage.transferTo(saveFile);
      } catch (Exception e) {
//        throw new RuntimeException("차량 이미지 업로드가 실패했습니다.");
        e.printStackTrace();
      }
    }


    carService.setNewCar(car);
    return "redirect:/cars";
  }
```

4. **JSP 설정**

/images/ 로 설정해도 경로찾기가 가능하다.

```
<%--                <img src='<c:url value="/resources/images/${car.getCarimage().getOriginalFilename()}"/>'/> --%>
                <img src='<c:url value="/images/${car.getCarimage().getOriginalFilename()}"/>'/>
```

cars.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<head>
  <title>cars</title>
  <%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
      <meta charset="utf-8">

</head>
<body class="text-center">
<%@ include file="header.jsp" %>
<div class="alert alert-dark" role="alert">
<div class="container"><h1>차량 보기(김도영)</h1></div>
</div>


  <div class="container">
    <div class="row" align="center">

      <c:forEach items="${carList}" var="car">
        <div class="col-md-4">



<%--        <img src='<c:url value="/resources/images/${car.getCarimage().getOriginalFilename()}"/>'/> --%>
        <img src='<c:url value="/images/${car.getCarimage().getOriginalFilename()}"/>'/>
        <h3>${car.cid}</h3>
        <p>${car.cname}
        <p>${car.cprice}만원
        <p><a href="/car?id=${car.cid }" class="btn btn-Secondary" role="button">상세보기</a></p>
      <%-- <a href='<c:url value="/car?id=${car.cid }"/>' class="btn btn-Secondary" role="button"> --%>

        </div>


      </c:forEach>

    </div>
  </div>



</body>
</html>
```

Exception(예외 처리)

> 💡 **내부 브라우저에서는 확인이 안되기 때문에 외부 브라우저를 이용하자.**

주요 예외 처리(404 Error는 따로 설정하여야 한다.)

ErrorController.java

```
package com.carshop.controller;

import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.servlet.ModelAndView;

@ControllerAdvice
public class ErrorController {

  @ExceptionHandler(RuntimeException.class)
  private ModelAndView handleErrorCommon(Exception e) {
    ModelAndView modelAndView = new ModelAndView();
    modelAndView.addObject("exception", e);
    modelAndView.setViewName("errorCommon");

    return modelAndView;




  }
}
```

errorCommon.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>errorCommon.jsp</title>
</head>
<body>
<h1>${exception}</h1>
<br>
예외처리
</body>
</html>
```

404 Error 처리

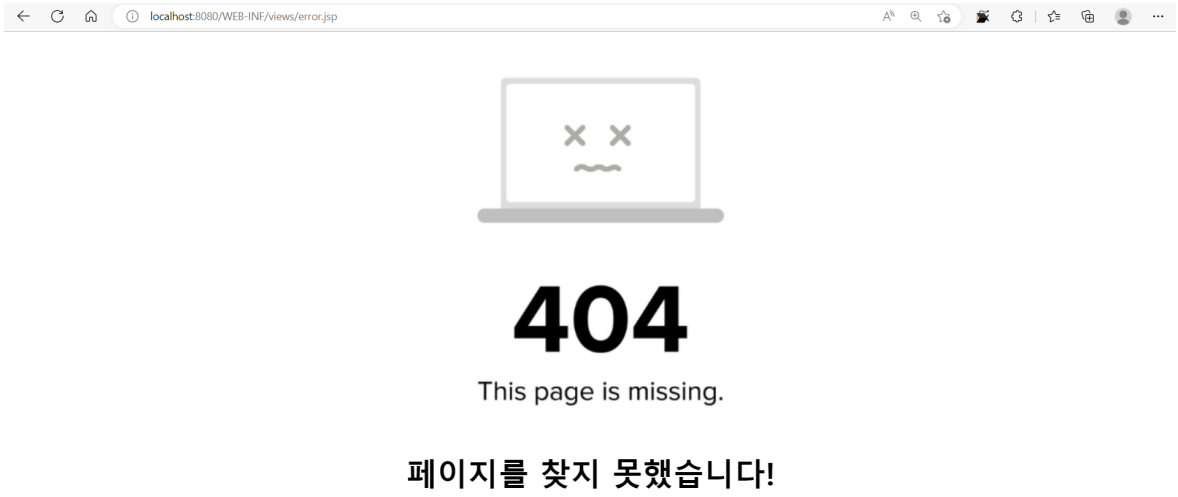
web.xml

```
<error-page>
    <error-code>404</error-code>
    <location>/WEB-INF/views/error.jsp</location>
  </error-page>
```

error.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>404 page</title>
</head>
<body>
<div style="text-align: center;">
<img src="/images/404.png">

<h1>페이지를 찾지 못했습니다!</h1>
</div>
```

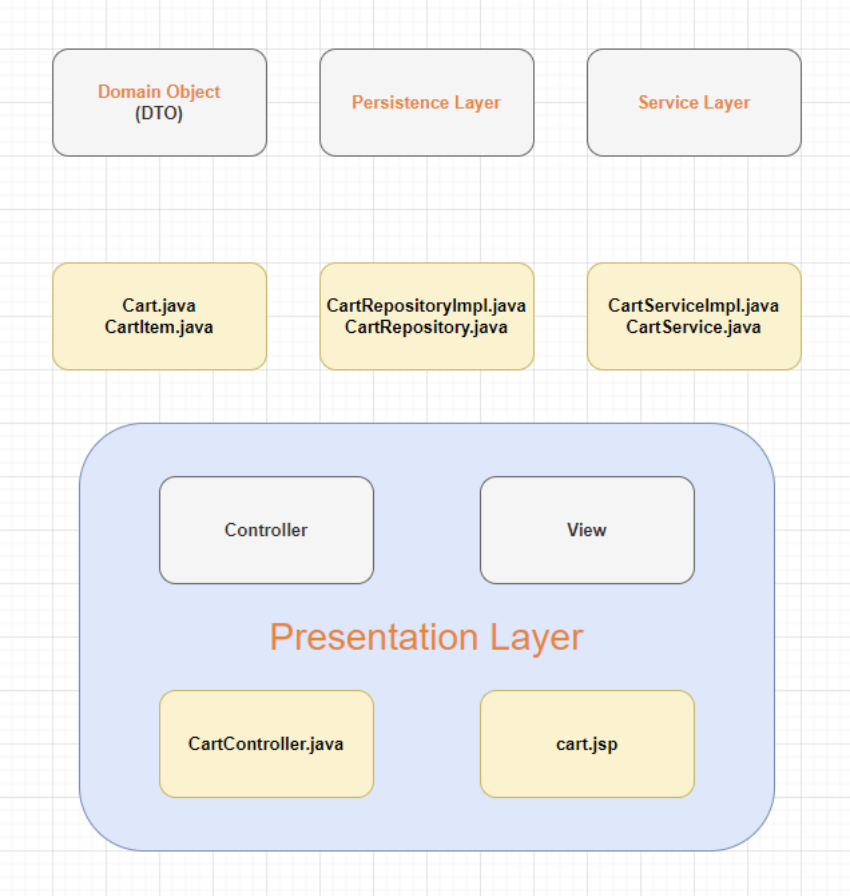
```
</body>
</html>
```



▼ **etc.**

[Tomcat 에러]Several ports (8005, 8080, 8009) required by Tomcat v9.0 Server at localhost are already in use. The server may already be ru
톰캣을 구동시켰을 때 생긴 포트 오류이다.나는 이클립스가 비정상적으로 종료가 된 후에 톰캣을 구동시켰을 때 발생했다. Several ports (8005, 8080, 8009) required by
process may be using the port. To start this server you will need to stop the other process or change the port number(s). Tomcat이 사용하고 있는 기본 포트(808
https://to-dy.tistory.com/59

장바구니 만들기



CartItem.java

```
package com.carshop.controller;

public class CartItem {

  private CarDTO car;   //자동차
  private int quantity; //자동차 개수
  private int totalPrice; //자동차 가격

  public CarDTO getCar() {
    return car;
  }
  public void setCar(CarDTO car) {
    this.car = car;
  }
  public int getQuantity() {
    return quantity;
  }
  public void setQuantity(int quantity) {
    this.quantity = quantity;
  }
  public int getTotalPrice() {
    return totalPrice;
  }
  public void setTotalPrice(int totalPrice) {
    this.totalPrice = totalPrice;
  }

  public CartItem() {
    super();
  }

  public CartItem(CarDTO car) {
    super();
    this.car = car;
    this.quantity = 1;
    this.totalPrice = Integer.valueOf(car.getCprice());
  }
```

```
  @Override
  public int hashCode() {
    final int prime = 31;
    int result = 1;
    result = prime * result + ((car == null) ? 0 : car.hashCode());
    result = prime * result + quantity;
    result = prime * result + totalPrice;
    return result;
  }
  @Override
  public boolean equals(Object obj) {
    if (this == obj)
      return true;
    if (obj == null)
      return false;
    if (getClass() != obj.getClass())
      return false;
    CartItem other = (CartItem) obj;
    if (car == null) {
      if (other.car != null)
        return false;
    } else if (!car.equals(other.car))
      return false;
    if (quantity != other.quantity)
      return false;
    if (totalPrice != other.totalPrice)
      return false;
    return true;
  }

  public void updateTotalPrice() {
    int price = Integer.valueOf(car.getCprice());
    totalPrice = price * this.quantity;

  }




}
```

Cart.java

```
package com.carshop.controller;

import java.util.HashMap;
import java.util.Map;

public class Cart {

  private String cartId;
  private Map<String, CartItem> cartItems;
  private int grandTotal;

  public Cart() {
    cartItems = new HashMap<String, CartItem>();
  }

  public Cart(String cartId) {
    this();
    this.cartId = cartId;
  }

  public String getCartId() {
    return cartId;
  }

  public void setCartId(String cartId) {
    this.cartId = cartId;
  }

  public Map<String, CartItem> getCartItems() {
    return cartItems;
  }

  public void setCartItems(Map<String, CartItem> cartItems) {
    this.cartItems = cartItems;
  }

  public int getGrandTotal() {
    return grandTotal;
```

```
  }

  public void setGrandTotal(int grandTotal) {
    this.grandTotal = grandTotal;
  }

  @Override
  public int hashCode() {
    final int prime = 31;
    int result = 1;
    result = prime * result + ((cartId == null) ? 0 : cartId.hashCode());
    return result;
  }

  @Override
  public boolean equals(Object obj) {
    if (this == obj)
      return true;
    if (obj == null)
      return false;
    if (getClass() != obj.getClass())
      return false;
    Cart other = (Cart) obj;
    if (cartId == null) {
      if (other.cartId != null)
        return false;
    } else if (!cartId.equals(other.cartId))
      return false;
    return true;
  }

  public void updateGrandTotal() {
    grandTotal = 0;
    for (CartItem item : cartItems.values()) {
      grandTotal = grandTotal + item.getTotalPrice();
    }
  }



}
```

CartRepository.java

```
package com.carshop.controller;

public interface CartRepository {

  Cart create(Cart cart);
  Cart read(String cartId);


}
```

CartRepositoryImpl.java

```
package com.carshop.controller;

import java.util.HashMap;
import java.util.Map;

public class CartRepositoryImpl implements CartRepository {

  private Map<String, Cart> listOfCarts;

  public CartRepositoryImpl() {
    listOfCarts = new HashMap<String, Cart>();
  }



  @Override
  public Cart create(Cart cart) {
    if(listOfCarts.keySet().contains(cart.getCartId())) {
      throw new IllegalArgumentException(String.format("장바구니를 생성할 수 없습니다. 자동차 ID가 이미 존재합니다.", cart.getCartId()));
    }else {
```

```
      listOfCarts.put(cart.getCartId(), cart);
    }
    return cart;
  }

  @Override
  public Cart read(String cartId) {
    // cartId 를 이용하여 장바구니에 등록된 모든 정보를 가져온다.
    return listOfCarts.get(cartId);
  }

}
```

CartService.java

```
package com.carshop.controller;

public interface CartService {

  Cart create(Cart cart);
  Cart read(String cartId);

}
```

CartServiceImpl.java

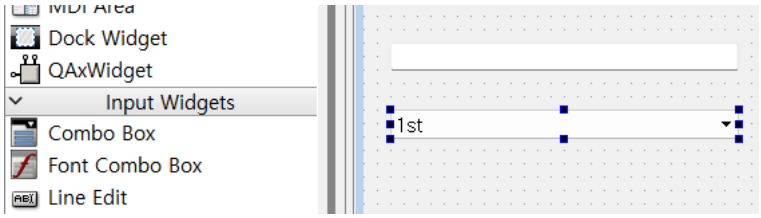
```
package com.carshop.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class CartServiceImpl implements CartService {

  @Autowired
  private CartRepository CartRepository;

  @Override
  public Cart create(Cart cart) {
    return CartRepository.create(cart);
  }

  @Override
  public Cart read(String cartId) {
    return CartRepository.read(cartId);
  }

}
```

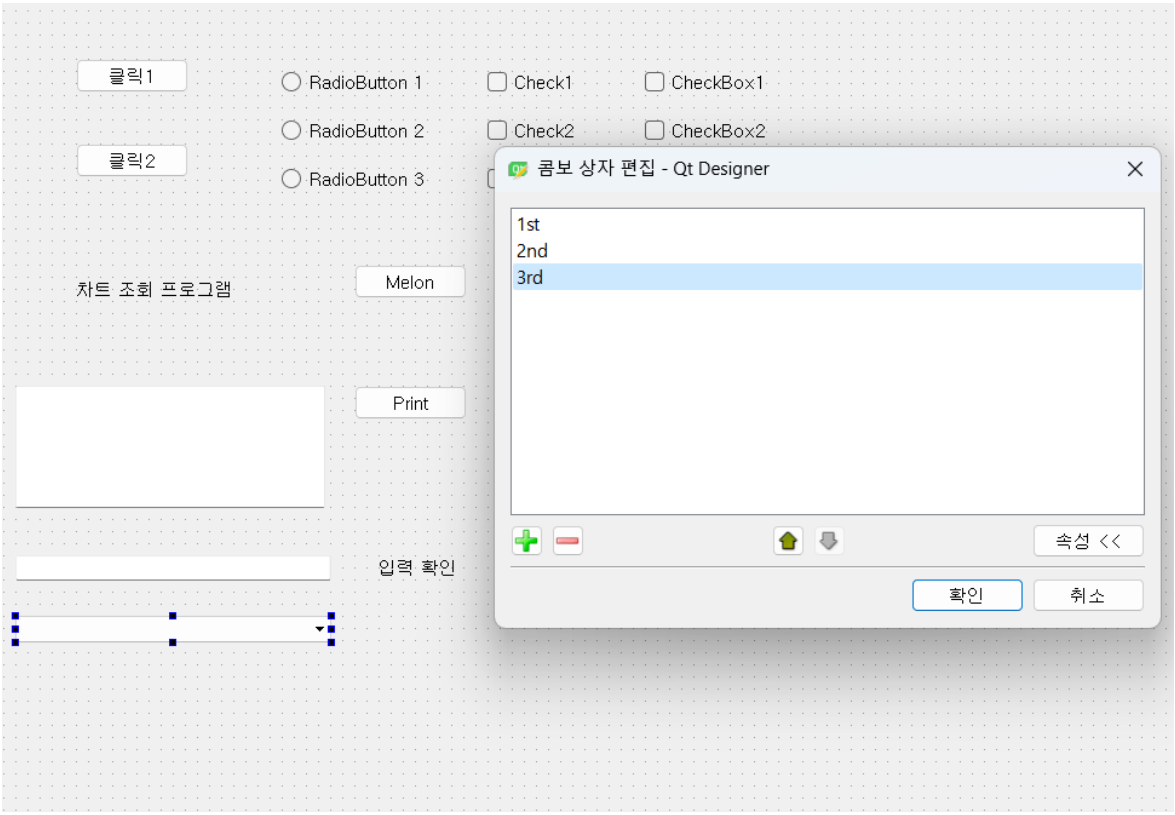
▼ Python

https://doc.qt.io/qtforpython-5/PySide2/QtWidgets/QComboBox.html

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/c60189c7-de28-40db-aff7-4ed2f70c6ac0/Untitled.pdf

ComboBox

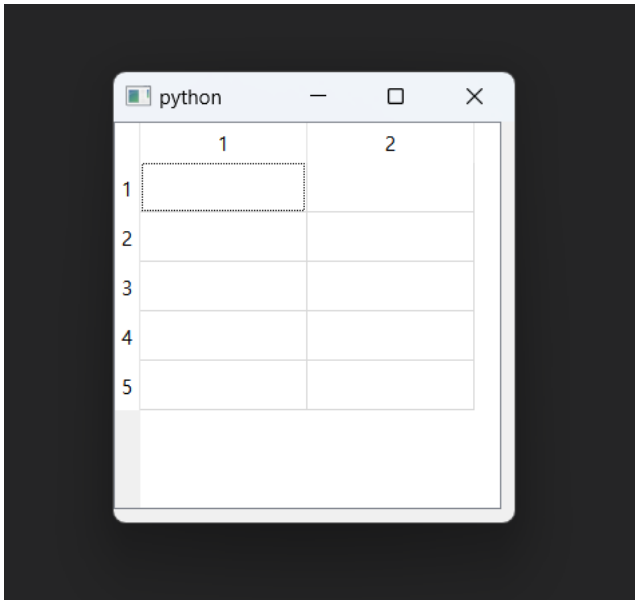





```
# ComboBox

import sys
from PyQt5.QtWidgets import *
from PyQt5 import uic

from_class = uic.loadUiType("mywindow8.ui")[0]

class MyWindow(QMainWindow, from_class):

    def __init__(self):
        super().__init__()
        self.setupUi(self)

        # 단추 클릭했을 때 콤보 박스 확인
        # self.comboBox
        self.PrintItem.clicked.connect(self.printFunction)

    def printFunction(self) :
        print(self.comboBox.currentText())

app  = QApplication(sys.argv)
window = MyWindow()
window.show()
app.exec_()
```

Table Widget

```
# Table Widget

import sys
from PyQt5.QtWidgets import *
from PyQt5 import uic


class MyWindow(QMainWindow):

    def __init__(self):
        super().__init__()
        self.setGeometry(800, 200, 300, 300)

        self.tableWidget = QTableWidget(self)
        self.tableWidget.resize(290, 290)

        # 열은 반드시 넣어주어야 한다.
        self.tableWidget.setColumnCount(2)

        self.tableWidget.setRowCount(5)



app  = QApplication(sys.argv)
window = MyWindow()
window.show()
app.exec_()
```



💡 **Table Widget 의 열을 지정하지 않으면 에러가 발생한다.**

열을 지정하지 않은 경우

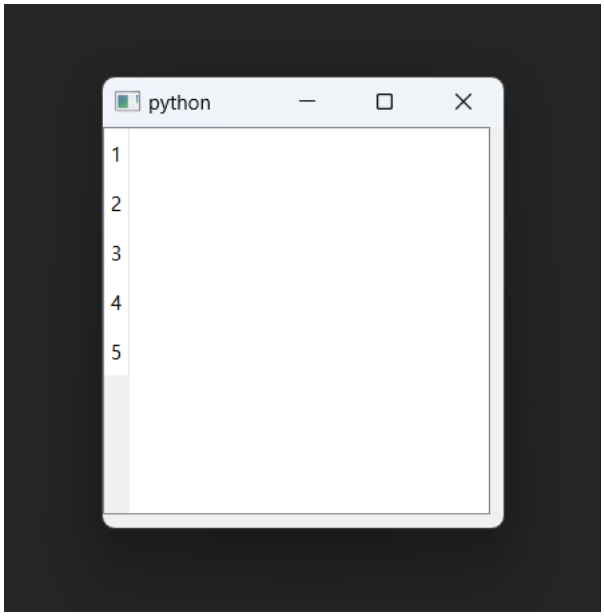


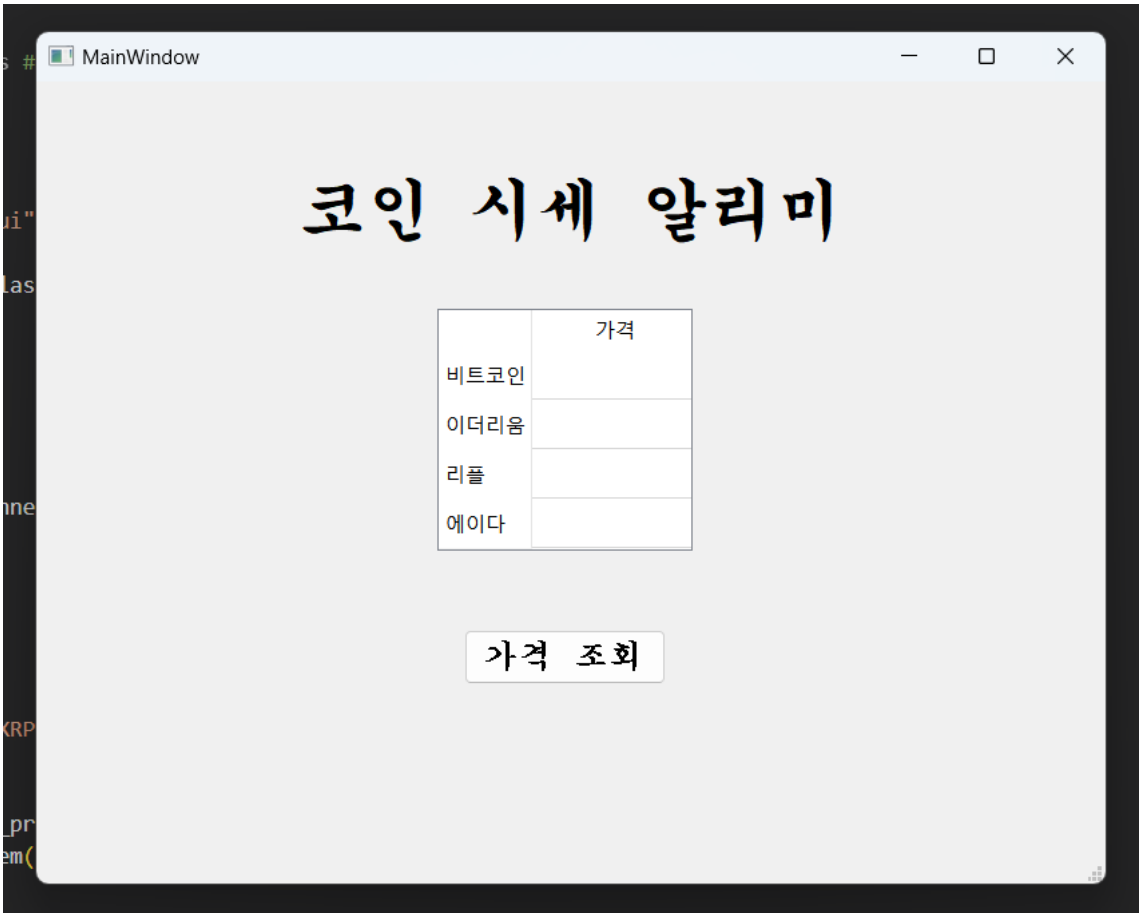
Table Widget 데이터 넣기

```
# Table Widget

import sys
from PyQt5.QtWidgets import *
from PyQt5 import uic


class MyWindow(QMainWindow):

    def __init__(self):
        super().__init__()

        # 윈도우 창
        self.setGeometry(800, 200, 300, 300)

        # 윈도우에 테이블위젯 넣기
        self.tableWidget = QTableWidget(self)
        self.tableWidget.resize(290, 290)
        # self.tableWidget.setColumnCount(2)
        self.tableWidget.setRowCount(5)

        # 테이블위젯에 데이터 넣기(행, 열, 넣을 값)
        self.tableWidget.setItem(0, 0, QTableWidgetItem("비트코인"))
        self.tableWidget.setItem(0, 1, QTableWidgetItem("31000000"))

        self.tableWidget.setItem(1, 0, QTableWidgetItem("이더리움"))
        self.tableWidget.setItem(1, 1, QTableWidgetItem("2100000"))

        self.tableWidget.setItem(2, 0, QTableWidgetItem("리플"))

app  = QApplication(sys.argv)
window = MyWindow()
window.show()
app.exec_()
```

▼ 현재 날씨 알리미(실습)

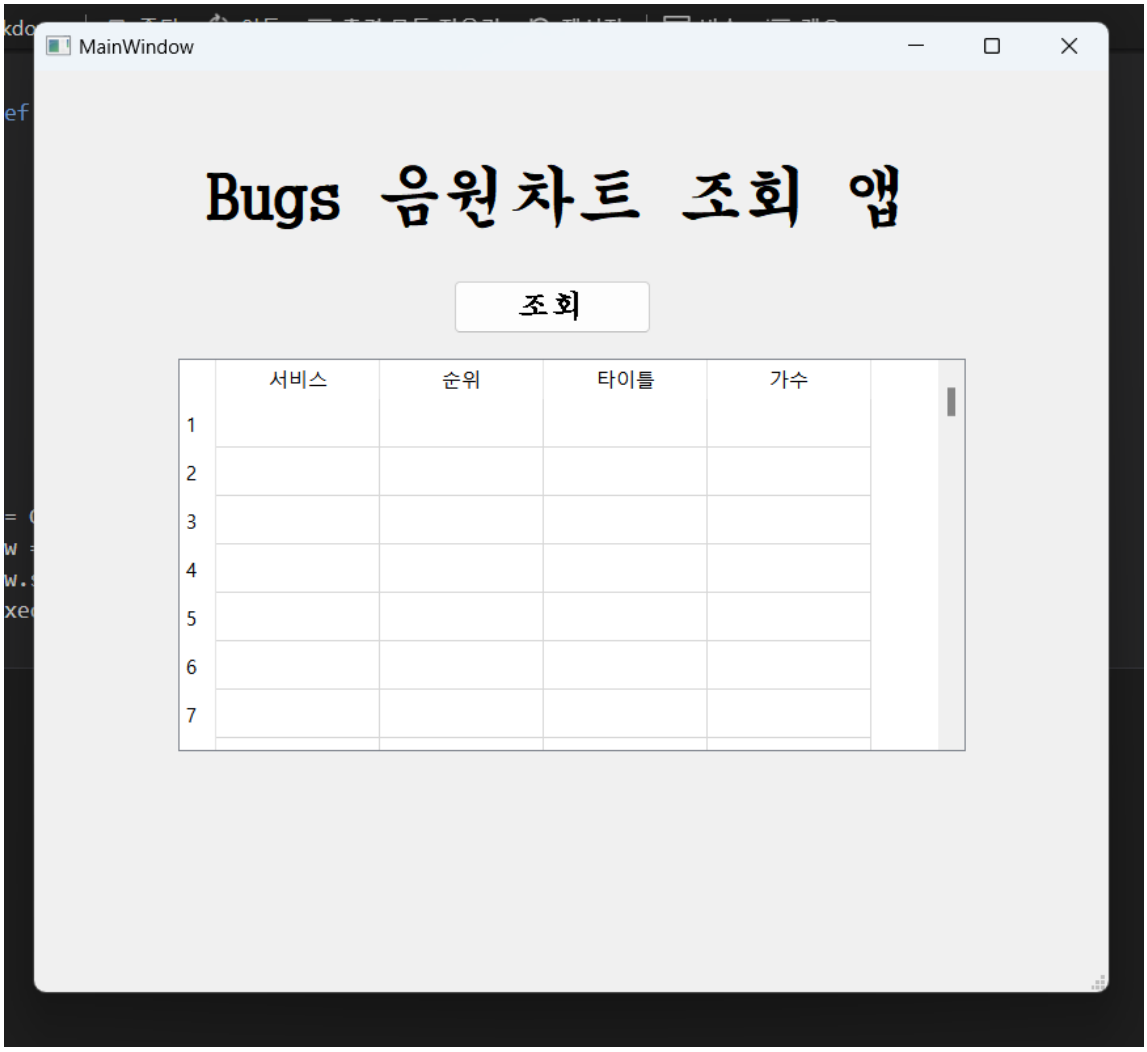

```
# 현재 온도 조회 앱

import sys
from PyQt5.QtWidgets import *
from PyQt5 import uic

import requests
from bs4 import BeautifulSoup as bs # 데이터 분석을 용이하게 정제

from_class = uic.loadUiType("weather.ui")[0]

class MyWindow(QMainWindow, from_class):

    def __init__(self):
        super().__init__()
        self.setupUi(self)

        # 단추를 누르면
        self.pushButton.clicked.connect(self.printFunction)

        # 조회 후 칸에 출력
    def printFunction(self) :
        txt = requests.get('https://weather.naver.com/')
        html = bs(txt.text)
        temp = html.select('strong.current')[0].text.strip()
        self.lineEdit.setText(temp[5:])

app  = QApplication(sys.argv)
window = MyWindow()
window.show()
app.exec_()
```

▼ 코인 시세 알리미(실습)



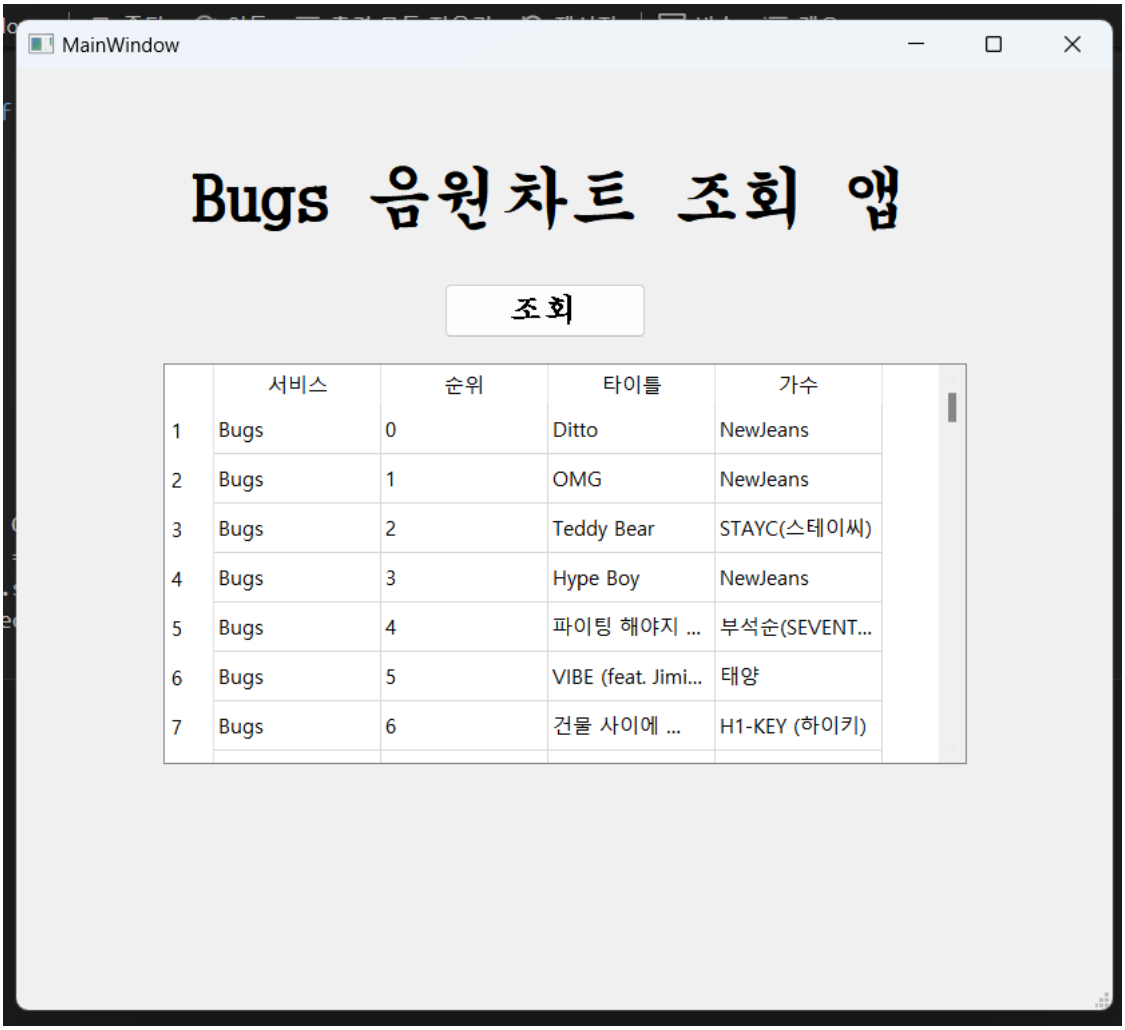

```
# 비트코인 시세 알리미

import sys
from PyQt5.QtWidgets import *
from PyQt5 import uic

import requests
from bs4 import BeautifulSoup as bs # 데이터 분석을 용이하게 정제
import pybithumb as pb



from_class = uic.loadUiType("coin.ui")[0]

class MyWindow(QMainWindow, from_class):

    def __init__(self):
        super().__init__()
        self.setupUi(self)

        # 단추를 누르면
        self.pushButton.clicked.connect(self.printFunction)



        # 조회 후 칸에 출력
    def printFunction(self) :

        tickers = ['BTC', 'ETH', 'XRP', 'ADA']

        for i, ticker in enumerate(tickers):
            price = pb.get_current_price(ticker)
            self.tableWidget.setItem(i, 0, QTableWidgetItem(str(price)))

app  = QApplication(sys.argv)
window = MyWindow()
window.show()
app.exec_()
```

▼ **Bugs 음원차트 조회 앱(실습)**

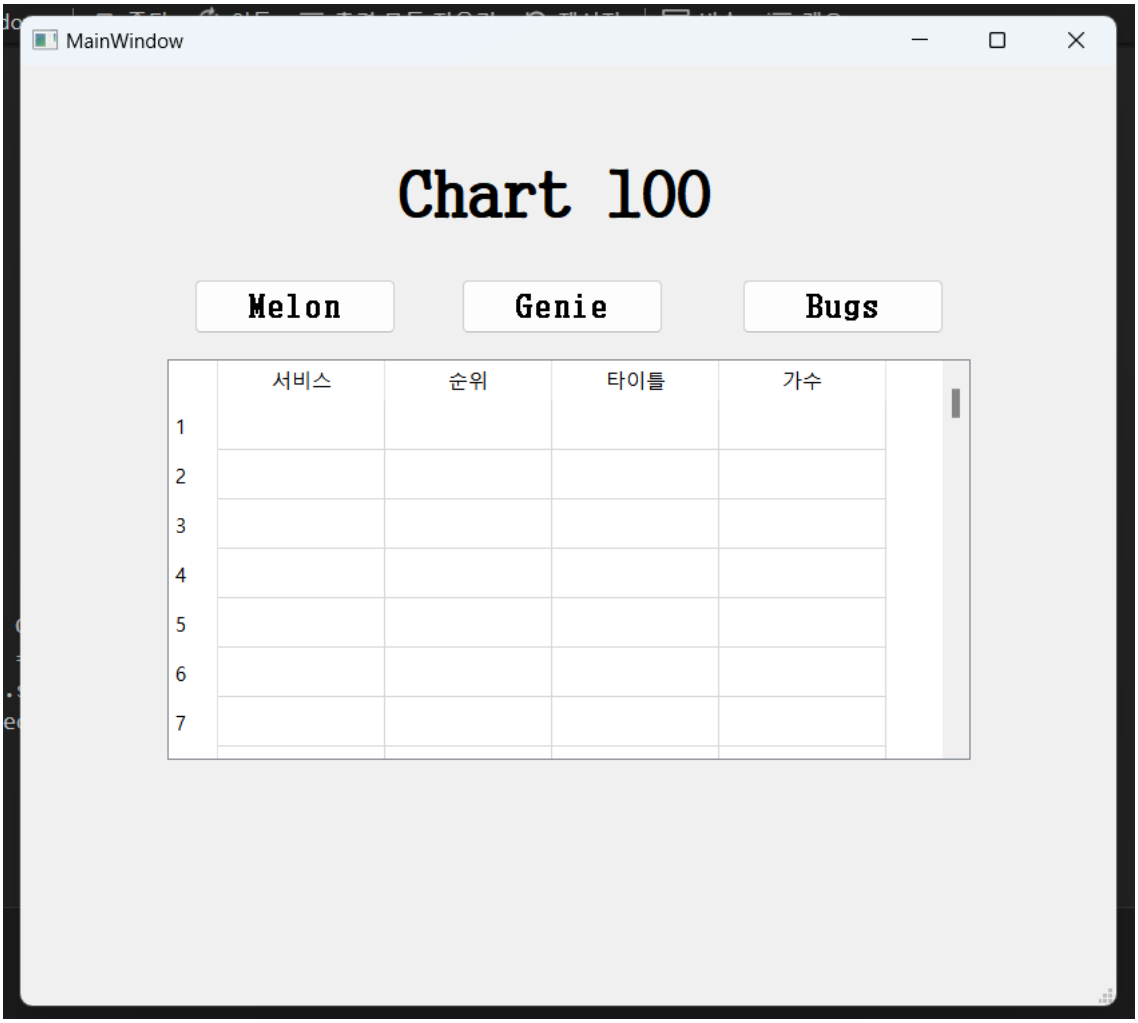



```
# Bugs 음원차트 조회 앱

import sys
from PyQt5.QtWidgets import *
from PyQt5 import uic

import requests
from bs4 import BeautifulSoup as bs # 데이터 분석을 용이하게 정제
import pybithumb as pb



from_class = uic.loadUiType("chart.ui")[0]

class MyWindow(QMainWindow, from_class):

    def __init__(self):
        super().__init__()
        self.setupUi(self)

        # 단추를 누르면
        self.pushButton.clicked.connect(self.printFunction)



        # 조회 후 칸에 출력
    def printFunction(self) :
        url = "https://music.bugs.co.kr/chart"
        txt = requests.get(url)
        html = bs(txt.text)
        songs = html.select('table.byChart > tbody > tr')
        for i, song in enumerate(songs):
            title = song.select('p.title > a')[0].text
```

```
            singer = song.select('p.artist > a')[0].text
            self.tableWidget.setItem(i, 0, QTableWidgetItem("Bugs"))
            self.tableWidget.setItem(i, 1, QTableWidgetItem(str(i)))
            self.tableWidget.setItem(i, 2, QTableWidgetItem(str(title)))
            self.tableWidget.setItem(i, 3, QTableWidgetItem(str(singer)))

app  = QApplication(sys.argv)
window = MyWindow()
window.show()
app.exec_()
```

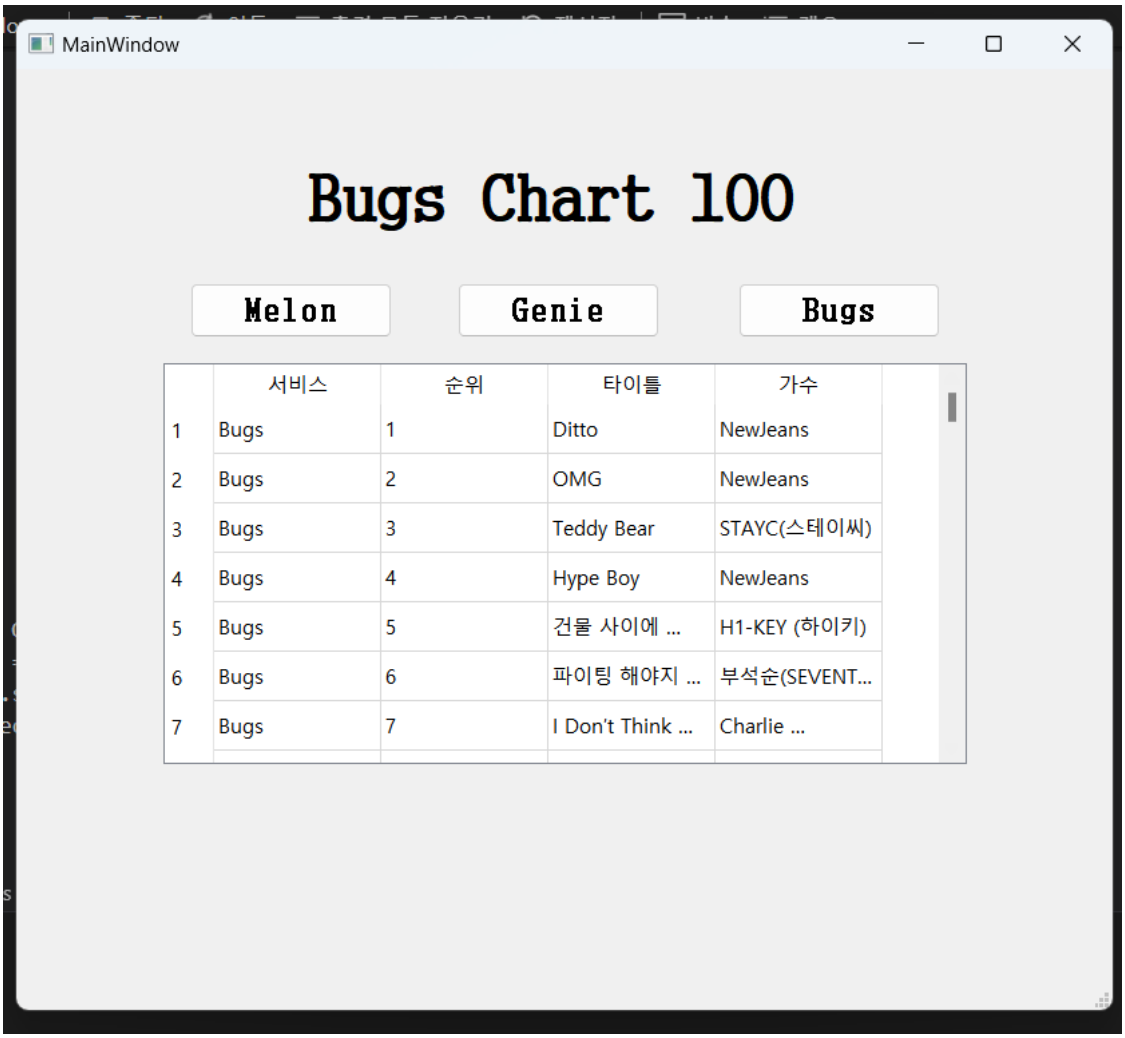


▼ 통합 음원차트 조회 앱(실습)




```
# 음원차트 조회 앱

import sys
from PyQt5.QtWidgets import *
from PyQt5 import uic
from PyQt5.QtCore import *

import requests
from bs4 import BeautifulSoup as bs # 데이터 분석을 용이하게 정제
from selenium import webdriver      # 웹 브라우저 컨트롤(크롬)


from_class = uic.loadUiType("musicChart.ui")[0]

class MyWindow(QMainWindow, from_class):

    def __init__(self):
        super().__init__()
        self.setupUi(self)

        self.timer1 = QTimer(self)
        self.timer2 = QTimer(self)
        self.timer3 = QTimer(self)

        self.timer1.timeout.connect(self.MelonFunction)
        self.timer2.timeout.connect(self.GenieFunction)
        self.timer3.timeout.connect(self.BugsFunction)

        # 단추를 누르면
        self.Melon.clicked.connect(self.MelonFunction)
        self.Genie.clicked.connect(self.GenieFunction)
        self.Bugs.clicked.connect(self.BugsFunction)
```

```
    def MelonFunction(self) :
        self.timer2.stop()
        self.timer3.stop()
        self.label.setText("Melon Chart 100")
        driver = webdriver.Chrome('chromedriver.exe')
        driver.get("https://www.melon.com/chart/index.htm")
        txt = driver.page_source
        html = bs(txt) # 벅스 문법과 약간 다름
        songs = html.select('tbody > tr')
        for i, song in enumerate(songs):
            title = song.select('div.rank01 > span > a')[0].text
            singer = song.select('div.rank02 > a')[0].text
            self.tableWidget.setItem(i, 0, QTableWidgetItem("Melon"))
            self.tableWidget.setItem(i, 1, QTableWidgetItem(str(i+1)))
            self.tableWidget.setItem(i, 2, QTableWidgetItem(str(title)))
            self.tableWidget.setItem(i, 3, QTableWidgetItem(str(singer)))

        self.timer1.start(5000)
        print("Melon")


    def GenieFunction(self) :
        self.timer1.stop()
        self.timer3.stop()
        self.label.setText("Genie Chart 100")
        # self.tableWidget.clear()
        driver = webdriver.Chrome('chromedriver.exe')
        driver.get("https://www.genie.co.kr/chart/top200?ditc=D&ymd=20230227&hh=16&rtm=Y&pg=1")
        txt = driver.page_source
        html = bs(txt) # 벅스 문법과 약간 다름

        songs = html.select('tbody > tr')

        for i, song in enumerate(songs):
            title = song.select('td > a.title')[0].text.strip()
            singer = song.select('td > a.artist')[0].text.strip()
            self.tableWidget.setItem(i, 0, QTableWidgetItem("Genie"))
            self.tableWidget.setItem(i, 1, QTableWidgetItem(str(i+1)))
            self.tableWidget.setItem(i, 2, QTableWidgetItem(str(title)))
            self.tableWidget.setItem(i, 3, QTableWidgetItem(str(singer)))

        driver = webdriver.Chrome('chromedriver.exe')
        driver.get("https://www.genie.co.kr/chart/top200?ditc=D&ymd=20230227&hh=16&rtm=Y&pg=2")
        txt = driver.page_source
        html = bs(txt) # 벅스 문법과 약간 다름

        songs = html.select('tbody > tr')

        for i, song in enumerate(songs):
            i = i + 50
            title = song.select('td > a.title')[0].text.strip()
            singer = song.select('td > a.artist')[0].text.strip()
            self.tableWidget.setItem(i, 0, QTableWidgetItem("Genie"))
            self.tableWidget.setItem(i, 1, QTableWidgetItem(str(i+1)))
            self.tableWidget.setItem(i, 2, QTableWidgetItem(str(title)))
            self.tableWidget.setItem(i, 3, QTableWidgetItem(str(singer)))

        self.timer2.start(5000)
        print("Genie")


    def BugsFunction(self) :
        self.timer1.stop()
        self.timer2.stop()
        self.label.setText("Bugs Chart 100")
        url = "https://music.bugs.co.kr/chart"
        txt = requests.get(url)
        html = bs(txt.text)
        songs = html.select('table.byChart > tbody > tr')
        for i, song in enumerate(songs):
            title = song.select('p.title > a')[0].text
            singer = song.select('p.artist > a')[0].text
            self.tableWidget.setItem(i, 0, QTableWidgetItem("Bugs"))
            self.tableWidget.setItem(i, 1, QTableWidgetItem(str(i+1)))
            self.tableWidget.setItem(i, 2, QTableWidgetItem(str(title)))
            self.tableWidget.setItem(i, 3, QTableWidgetItem(str(singer)))

        self.timer3.start(5000)
        print("Bugs")


app  = QApplication(sys.argv)
window = MyWindow()
window.show()
app.exec_()
```


MainWindow
Bugs Chart 100
Melon
Genie
Bugs
서비스
순위
타이틀
가수
1 Bugs 1 Ditto NewJeans
2 Bugs 2 OMG NewJeans
3 Bugs 3 Teddy Bear STAYC(스테이씨)
4 Bugs 4 Hype Boy NewJeans
5 Bugs 5 건물 사이에 ... H1-KEY (하이키)
6 Bugs 6 파이팅 해야지 ... 부석순(SEVENT...
7 Bugs 7 I Don't Think ... Charlie ...