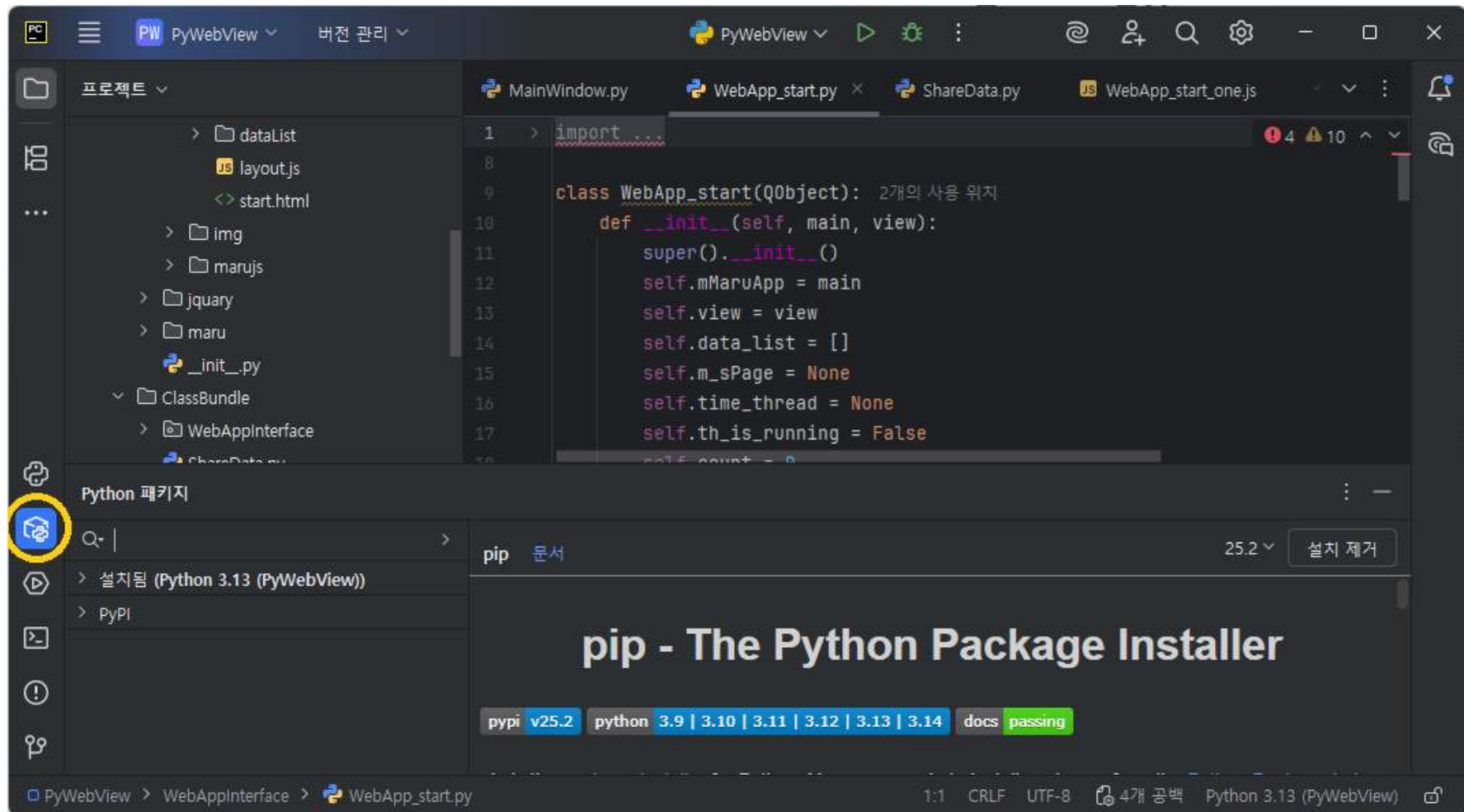


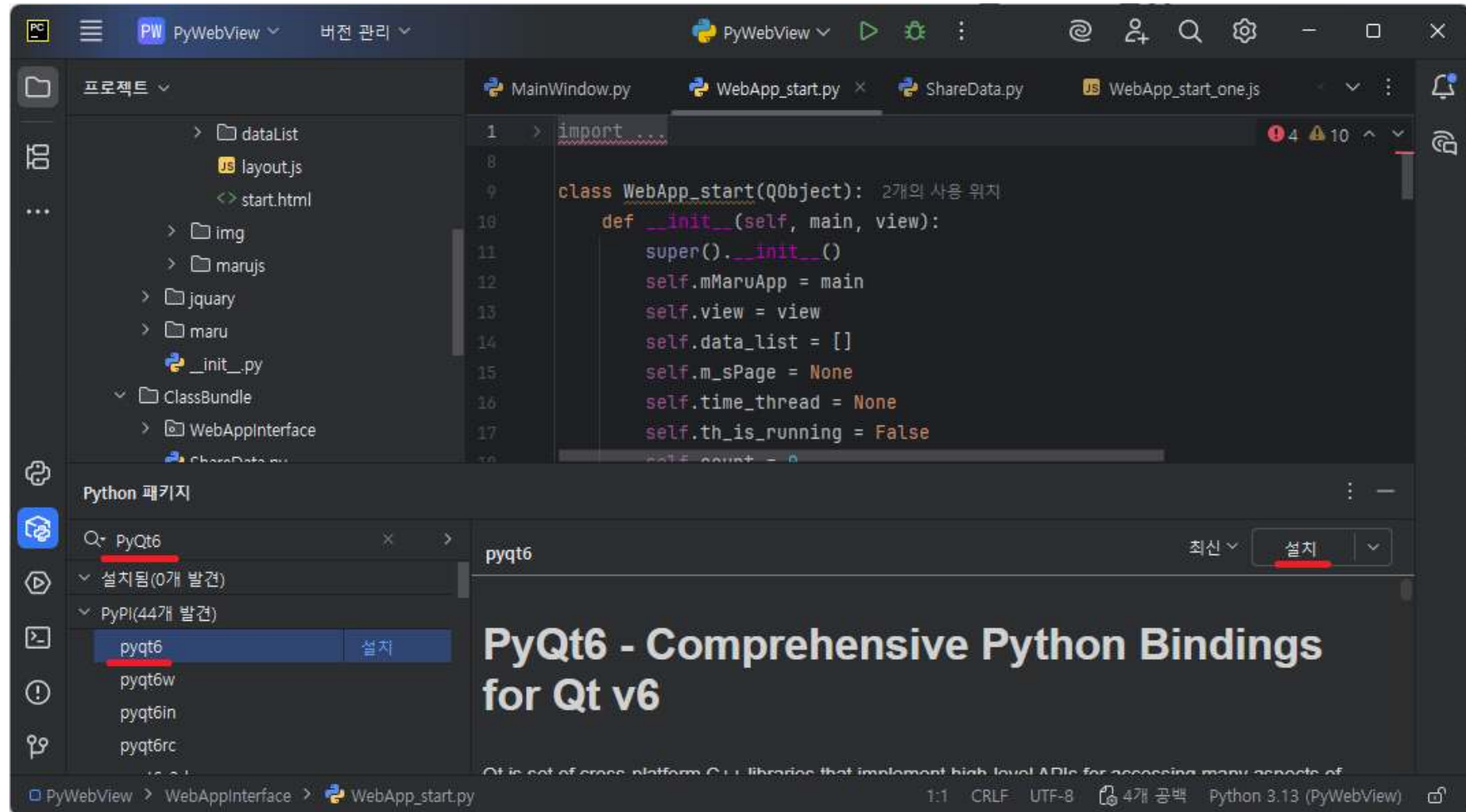
Py Maru Source 구동

1. Github에서 다운받은 후 PyCham에 Project 불러온 후 패키지버튼 누름



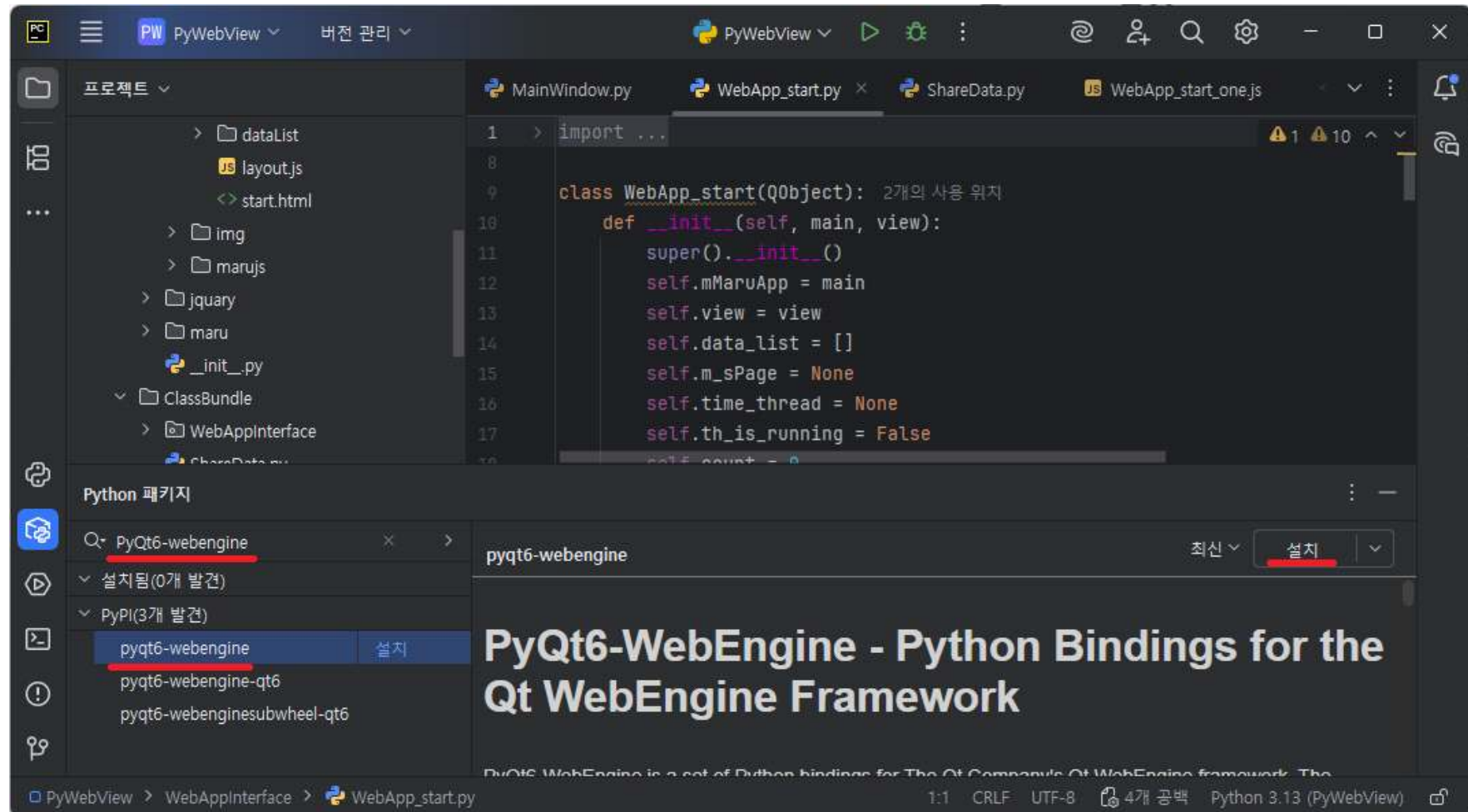
Py Maru Source 구동

2. PyQt6 최신버전 설치



Py Maru Source 구동

3. PyQt6-webengine 설치



Py Maru Source 구동

4. 2개의 package 설치 후

The screenshot shows the PyCharm IDE interface. The top toolbar includes buttons for running (a green play icon) and debugging (a green bug icon). The left sidebar displays the project structure for 'PyWebView', showing folders like '.venv', 'Lib', 'Scripts', 'assets', 'form', and 'css'. The main editor window shows the 'MainWindow.py' file with the following Python code:

```
1 import sys
2 import os
3 import json
4 import datetime
5
6 from PyQt6.QtWidgets import QApplication, QMainWindow
7 from PyQt6.QtWebEngineWidgets import QWebEngineView
8 from PyQt6.QtWebEngineCore import QWebEngineProfile, QWebEngineSettings
9 from PyQt6.QtWebChannel import QWebChannel
```

Below the editor, the 'Python 패키지' (Python Packages) panel is open. It shows a list of installed packages for Python 3.13 (PyWebView). The 'pyqt6' package is highlighted with a red box. The list includes:

Package Name	Version
pip	25.2
pyqt6	6.9.1
pyqt6-qt6	6.9.2
pyqt6-sip	13.10.2
pyqt6-webengine	6.9.0
pyqt6-webengine-qt6	6.9.2

The 'pyqt6' package details are shown on the right, including the title 'PyQt6 - Comprehensive Python Bindings for Qt v6' and a description: 'Qt is set of cross-platform C++ libraries that implement high-level APIs for accessing many aspects of modern desktop and mobile systems. These include location and positioning services, multimedia, NFC and Bluetooth connectivity, a Chromium based web browser, as well as traditional UI development.'

Maru Platform

Html 내부 구조

```
<!DOCTYPE html>
<html>
<head>
  <script src="../../maru/qwebchannel.js"> </script> <!--Py 추가 -->
</head>
<body>
  <div data-role="page" id="page-aaaaa" data-page="aaaaa">
    <div data-role="top"> </div> <!-- 화면 윗부분 디자인-->
    <div data-role="body" id="aaaaa-body">
      <div data-role="content" id="aaaaa_111" data-loc="aaaaa">

        <!-- 가운데 화면 디자인-->

      </div>
    </div>
    <div data-role="botton"> </div> <!-- 화면 아랫부분 디자인-->
  </div>
</body>
</html>
```

- * top, botton은 다른 임의의 단어로 교체가능
- * <div data-role="top"> </div> 삭제 가능
- * assets -> html -> layout.js에 등록

* aaaa 와 bbbbb 는 각 한 화면을 나타냄

assets 파일 구조

```
↓ assets
↓ form
  → css
  → fonts
  ↓ html
    ↓ aaaaa
      aaaaa.top.html
      aaaaa.botton.html
      aaaaa_222.html
    ↓ bbbbb
      bbbbb.left.html
      bbbbb.right.html
      bbbbb_111.html
      bbbbb_222.html
    aaaaa.html
    layout.js
  ↓ marujs
    ↓ aaaaa
      WebApp_aaaaa.js
      WebApp_aaaaa_111.js
      WebApp_aaaaa_222.js
    ↓ bbbbb
      WebApp_bbbbb.js
      WebApp_bbbbb_111.html
      WebApp_bbbbb_222.html
  ↓ source
  → js
  → maru
```

- * source ← 그림파일 모음
- * js ← JQuery 모음

Py Maru -- 1. MainWindow.py의 구조

MainWindow.py

```
class MainWindow(QMainWindow):
    js_signal = pyqtSignal(datetime.datetime)
    def __init__(self):
        super().__init__()
        self.jObject = {}
        self.jArray = []

        self.setWindowTitle("HTML만으로 Python 함수 실행")
        self.setGeometry(100, 100, 800, 600)
        self.channel_ready = False

        self.browser = QWebEngineView()
        self.setCentralWidget(self.browser)
        # Make the window fullscreen
        self.showFullScreen()

        s = QWebEngineProfile.defaultProfile().settings()
        s.setAttribute(QWebEngineSettings.WebAttribute.FullScreenSupportEnabled, True)
        s.setAttribute(QWebEngineSettings.WebAttribute.AllowRunningInsecureContent, True)
```

PRJ 파일 구조

```
↓ project
→ ClassBundle
↓ WebAppInterface
    WApp_aaaaa.cs
    WApp_bbbbbb.cs
→ assets
    MainWindow.py
```


Py Maru -- 1. MainWindow.py의 구조2

MainWindow.py

```
# QWebChannel과 Python 객체 연결
self.channel = QWebChannel()
self.wa_start = WebApp_start(self, self.browser)

# HTML 파일 로드 후, 로드 완료 시그널에 함수 연결
self.browser.loadFinished.connect(self.onLoadFinished)

# "Maru_start"라는 이름으로 등으로 Python 객체를 등록합니다.
self.channel.registerObject("Maru_app", self)
self.channel.registerObject("Maru_start", self.wa_start)
self.browser.page().setWebChannel(self.channel)

# HTML 파일 경로 설정
#html_file_path = os.path.abspath("web_channel.html")
html_file_path = os.path.abspath("./assets/form/html/start.html")
# 웹 페이지를 엽니다.
#self.browser.setUrl(QUrl("./web_channel.html")) # [4]
try:
    with open(html_file_path, "r", encoding="utf-8") as f:
        html_content = f.read()

    # setHtml()을 사용하여 HTML 문자열을 로드
    self.browser.setHtml(html_content, baseUrl=QUrl.fromLocalFile(html_file_path))

except FileNotFoundError:
    print(f"오류: '{html_file_path}' 파일을 찾을 수 없습니다.")
```

Py Maru -- 3. Button의 동작 흐름

html

```
<p class="SYS-BTN-TOUCH" ></p>
```

WebApp_aaaaa.js or WebApp_aaaaa_xxx.js

```
var add_Evnt = {  
  "btn": {  
    "TOUCH": {  
      Request: function (e, PageBlock) {  
        window['Maru_' + oPageInfo.page].btn_TOUCH_CLICK();  
      },  
    },  
  },  
};
```

WApp_aaaaa.py

```
def btn_TOUCH_CLICK (self)  
{  
    .  
    .  
    .  
}
```


C# Maru -- 4. javascript <--> MainWindow.py

html

```
<span class="SYS-TXT-TIMER_DAY">2025.04.25.</span>
```

MainWindow.py

```
def text_put(self, _sjson:str):  
    js_code = f"""Text_put('{_sjson}');"""  
    #print(js_code)  
    self.browser.page().runJavaScript(js_code)
```

WebApp.site.js

```
var Text_put = function (json) {  
    var Vars = JSON.parse(json);  
    for (var i in Vars) {  
        ele = $(' .SYS-TXT-' + i.toUpperCase());  
        if (ele.length) {  
            if (ele.is("input"))  
                ele.val(Vars[i]);  
            else  
                ele.html(Vars[i]); // ele.text( Vars[i] );  
        }  
    }  
}
```