

HICC 개발 입문 세미나

## 3주차. 웹 프레임워크

# HTML 개념 설명



## 내용, 구조

HTML은 웹 페이지에 올라갈 내용과 전체 페이지의 구조를 담당하는 언어입니다.

HTML을 통해 제목, 문단, 이미지, 링크, 테이블, 폼 등 다양한 요소를 사용해 페이지에 콘텐츠를 배치할 수 있습니다.

웹의 가장 기본이 되는 언어로서, 브라우저는 HTML 코드를 해석하여 웹 페이지에 띄워주는 역할을 합니다.

# CSS 개념 설명



디자인

CSS는 **HTML** 요소들의 디자인을 지정하고, 웹페이지의 레이아웃을 구성하는 역할을 합니다.

해당 식당에 책상 및 의자 몇 개를 둘지, 식당의 구조를 어떻게 정의할지, 메뉴 보드를 둘지 말지, 둔다면 어떤 내용을 적을지 등등을 담당하는게 HTML이라면,

어떻게 생긴 책상과 의자를 어디에 둘지, 식당에서 어느 위치에 있는게 주방이고, 어느정도 평수가 화장실인지, 메뉴 보드를 둔다면 어디에 둘 것이고 어떻게 꾸밀 것인지 등등을 담당하는게 CSS입니다.

# JS 개념 설명

JavaScript



동작, 상호작용

JavaScript는 **사용자와 인터렉션을 통한 여러가지 동작을 구현하고 서버와 상호작용**을 하는 역할을 지닙니다.

해당 식당에 책상 및 의자 몇 개를 둘지, 식당의 구조를 어떻게 정의할지, 메뉴 보드를 둘지 말지, 둔다면 어떤 내용을 적을지 등등을 담당하는게 HTML이라면,

JS는 손님이 왔을 때 어떤 식으로 맞이할지, 손님의 요청에 어떤 서비스를 제공할지, 손님이 주문한 내용을 주방(서버)에 전달하거나 완성된 음식을 손님에게 서빙하는 웨이터와 비슷한 역할을 합니다.

# 마크업 언어와 프로그래밍 언어



마크업 언어

단순히 입력된 코드를 브라우저가 화면에 띄워주는  
것만 가능

식당의 구조, 인테리어 등 시각적 도구에 비유 가능



JavaScript



프로그래밍 언어

웹에서 사용자와 실시간으로 상호작용 가능하게  
해줌

종업원, 주방 요리사 등 소통 가능한 사람에 비유  
가능 (실시간으로 명령이 가능하다)

# 프로그래밍 언어란?



JavaScript



JS는 프론트엔드에 최적화된 프로그래밍 언어

컴퓨터에게 명령이나 연산을 시킬 목적으로 설계되어 기계와 의사소통을 할 수 있게 해주는 언어를 뜻한다.

쉽게 말해 컴퓨터에게 명령을 내릴 수 있는 도구이다.

# Python 언어의 특징



다 필요없고  
Python은 쉽고 직관적이고 간결하다!  
프로그래밍에 입문하기 가장 적절한 언어

```
class HelloWorld {  
    static public void main( String args[] ) {  
        System.out.println( "Hello, world!" );  
    }  
}
```



```
print("Hello, world!")
```

# Python의 변수

프로그래밍 언어에는 변수라는 개념이 있는데, 데이터를 저장하기 위한 **이름이 붙은** 공간이라고 보면 됩니다. 쉽게 말해, 어떤 값을 저장하고 나중에 그 값을 꺼내 쓸 수 있게 해주는 **그릇** 같은 역할을 합니다.

해당 변수가 어떤 유형의 데이터를 저장할 수 있는지를 나타내는 것이 "자료형"이라는 개념인데, 파이썬에서는 변수에 저장되는 값에 맞춰서 자동으로 자료형이 설정됩니다.

```
# 문자열
name = "John"
print("Hello, " + name) # 출력: Hello, John

# 숫자
a = 5
b = 3
sum = a + b
print(sum) # 출력: 8

# 리스트
fruits = ["apple", "banana", "orange"]
print(fruits[1]) # 출력: banana

# 사전
person = {"name": "John", "age": 25}
print(person["name"]) # 출력: John
```



# Python의 함수

프로그래밍 언어에는 또한 함수라는 개념이 있는데, **여러 작업을 묶어서 이름을 붙인 코드 블록**입니다. 어떤 입력값(매개변수)을 받아서 **정해진 작업을 수행하고**, 그 결과를 출력(반환)할 수도 있습니다.

간단히 말해, 함수를 사용하면 **반복되는 작업을 하나로 묶어 재사용**할 수 있어서 코드가 더 깔끔하고 효율적이 됩니다.

함수의 이름을 적절히 붙이면, 해당 함수가 어떤 기능을 하는지 정확히 몰라도 적절히 사용할 수 있습니다.

아래와 같이 함수를 직접 정의할 수도 있고, Python 언어 자체에서 이미 정의해놓은 수많은 함수들을 사용할 수도 있습니다.

```
# 함수
def greet(name):
    print("Hello, " + name)

greet("Alice") # 출력: Hello, Alice
```

# Python의 모듈

모듈은 다른 파이썬 프로그램에서 불러와 사용할 수 있도록 만든 파이썬 파일이다.  
Django에선 프로그램을 개발할 때 수많은 모듈을 사용한다.

```
# 모듈 (math 모듈 사용 예제)
import math

print(math.pi) # 출력: 3.141592653589793
print(math.sqrt(16)) # 출력: 4.0

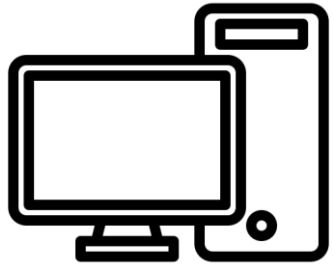
# 패키지 (패키지 내부의 모듈 사용 예제)
import random

print(random.randint(1, 10)) # 1부터 10 사이의 랜덤한 정수 출력
```

import 파이썬파일 코드를 통해, 다른 파이썬 파일을 불러와 사용할 수 있다.

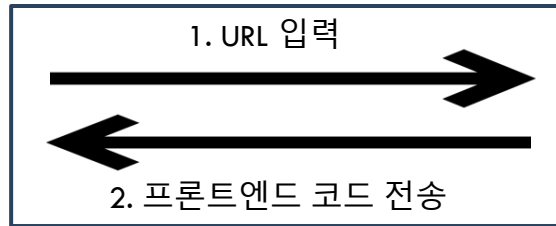
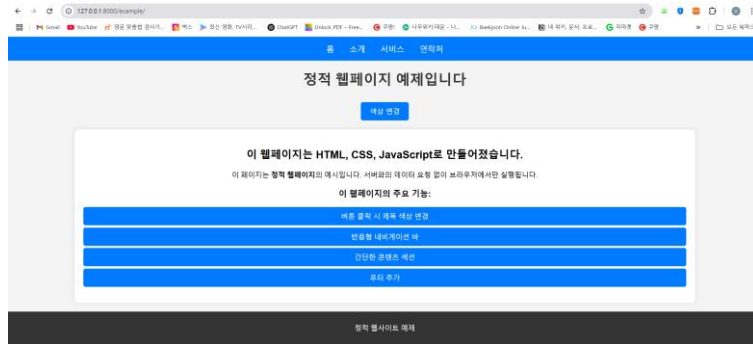
from 파이썬파일 import 함수 이런 방식으로도 가능한데,  
from math import sqrt << 해당 코드를 넣으면, print(sqrt(16))  
이런 식으로 앞에 모듈 이름을 생략하고 바로 함수를 사용할 수 있다.

# 웹 세계에서 일어나는 일 (정적)

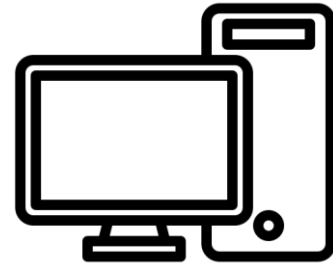


내 컴퓨터

3. 브라우저에서 코드 실행



해당 부분을 python 프로그래밍 언어로 구현 해야함



서버

# 웹 프레임워크



- Django는 파이썬으로 작성되는 웹 프레임워크입니다.
- 다른 프레임워크에 비해 난이도가 매우 낮은 것이 특징이며, 프로젝트 입문자들이 사용하기에 적합합니다.
- Instagram, Pinterest 같은 유명한 프로그램도 Django를 통해 개발되었습니다.
- Model – Template – View 의 구조를 따르고 있습니다.

# 웹 프레임워크 설명



도넛 가게를 창업한다고 가정합시다, 도넛 메뉴도 직접 연구해야하고 브랜딩도 필요하고, 이름도 지어야 하고, 디자인도 해야합니다. 포스 단말기도 세팅해야하고 내부 인테리어도 필요합니다.

하지만 크리스피크럼도넛 프랜차이즈에 들어가면 언급한 것들 대부분을 하지 않아도 됩니다. **프레임워크는 '프랜차이즈'입니다.** 가게 사장은 최대한 도넛 판매에만 집중할 수 있습니다.

# 웹 프레임워크 설명



웹 프로그램 하나를 만들기 위해서는 매우 많은 기능을 구현해야 합니다. 해당 사진과 같은 기능들이 예시인데, 이러한 기능들을 모두 백지 상태에서 직접 구현하려면 굉장히 많은 시간과 어려움이 있을 것입니다.

**웹 프레임워크**란 이러한 매우 자주 사용되는 기능들을 미리 구현하여 웹 개발자들의 편의성을 높여주는 프로그램입니다.

앞에서 언급한 "모듈"들을 웹 프로그램 개발에 편리한 것들만 모아놓은 밀키트같은 존재를 웹 프레임워크라고 보시면 됩니다.

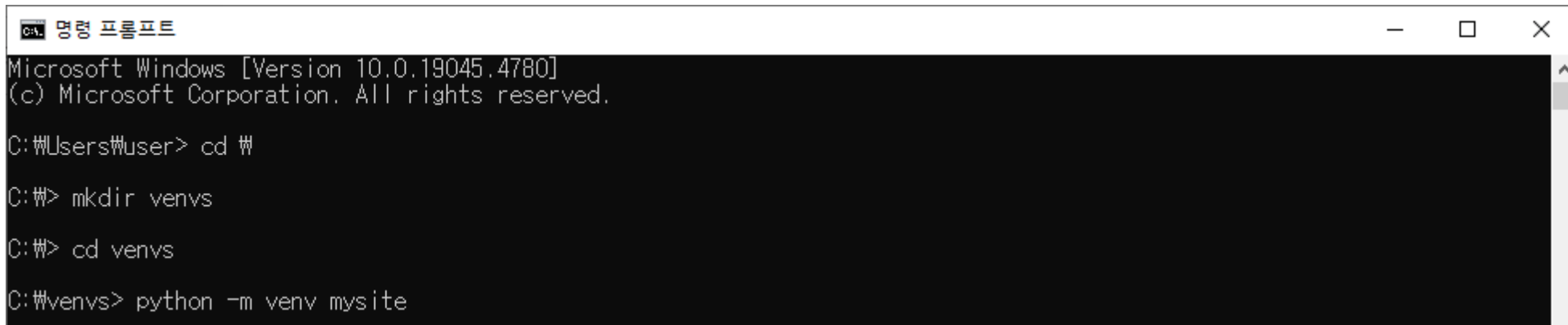
# 파이썬 가상환경 설정

## 파이썬 가상환경이란?

파이썬 프로젝트를 진행할 때, 독립된 환경을 만들어주는 도구  
하나의 PC에서 서로 다른 버전의 파이썬, 프레임워크, 라이브러리를 쉽게  
설치해 사용할 수 있도록 해줌

# 파이썬 가상환경 설정

명령 프롬프트를 실행하여, 해당 명령어를 타이핑 합니다



```
C:\> 명령 프롬프트
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User> cd \

C:\> mkdir venvs

C:\> cd venvs

C:\venvs> python -m venv mysite
```

`mysite`는 생성할 가상 환경의 이름, 다른 이름으로 지어도 됨  
(하지만 세미나 실습의 편의성을 위해 `mysite`로 통일해줍니다)

MacOS 버전 명령어

`cd ~`

`mkdir venvs`

`cd venvs`

`python3 -m venv mysite`

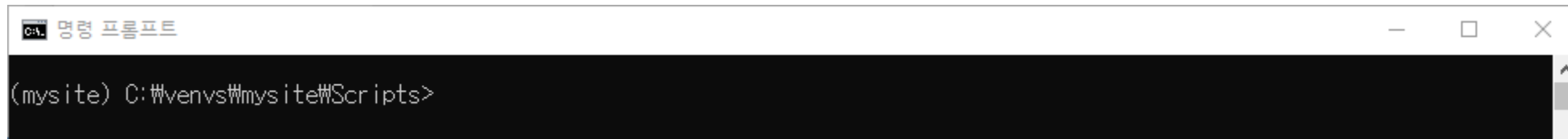


# 파이썬 가상환경 설정

```
C:\Users\User>cd C:\venvs\mysite\Scripts  
C:\venvs\mysite\Scripts> activate
```

해당 명령어를 통해 가상 환경에 진입

MacOS 버전은  
`cd ~/venvs/mysite/bin`  
`source activate`



```
C:\venvs\mysite\Scripts> activate  
(mysite) C:\venvs\mysite\Scripts>
```

이렇게 나오면 진입에 성공한 것

MacOS는 (mysite) user@MacBook mysite % 이런식으로 뜸  
(deactivate를 입력하면 가상 환경에서 탈출할 수 있습니다)

# 장고 설치하기

```
(mysite) C:\#venvs\mysite\Scripts>pip install django==4.0.3
Collecting django==4.0.3
  Using cached Django-4.0.3-py3-none-any.whl.metadata (4.0 kB)
Collecting asgiref<4,>=3.4.1 (from django==4.0.3)
  Using cached asgiref-3.8.1-py3-none-any.whl.metadata (9.3 kB)
Collecting sqlparse>=0.2.2 (from django==4.0.3)
  Using cached sqlparse-0.5.1-py3-none-any.whl.metadata (3.9 kB)
Collecting tzdata (from django==4.0.3)
  Using cached tzdata-2024.1-py2.py3-none-any.whl.metadata (1.4 kB)
Using cached Django-4.0.3-py3-none-any.whl (8.0 MB)
Using cached asgiref-3.8.1-py3-none-any.whl (23 kB)
Using cached sqlparse-0.5.1-py3-none-any.whl (44 kB)
Using cached tzdata-2024.1-py2.py3-none-any.whl (345 kB)
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.8.1 django-4.0.3 sqlparse-0.5.1 tzdata-2024.1

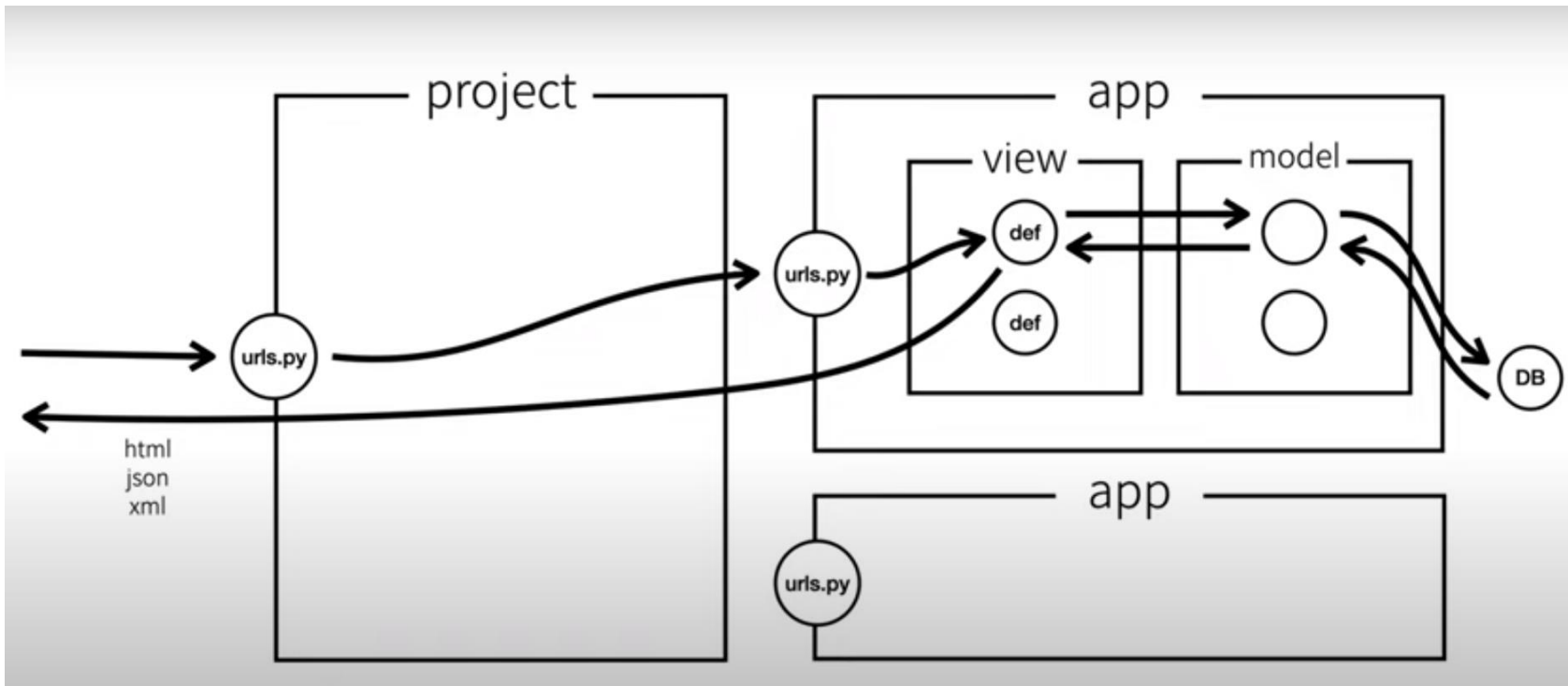
(mysite) C:\#venvs\mysite\Scripts>
```

해당 명령어를 통해 장고 설치

You should consider upgrading via the 'python -m pip install --upgrade pip' command. 같은 메시지가 뜨면 **python -m pip install --upgrade pip** 명령어를 통해 pip을 최신 버전으로 설치

# 장고 프로젝트 설치하기

장고에는 프로젝트라는 개념이 있는데, 프로젝트 하나는 하나의 웹사이트를 의미한다고 보면 됨  
그리고 그 프로젝트를 다루는 여러 개의 앱이 존재함  
(자세한 내용은 실제 강의에서 다룹니다)



# 장고 프로젝트 설치하기



```
ca 명령 프롬프트
Microsoft Windows [Version 10.0.19045.5608]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>cd ~
C:\>mkdir hiccseminar
C:\>cd hiccseminar
C:\hiccseminar>.
```

```
cd ~
mkdir hiccseminar
cd hiccseminar
```

다시 홈 디렉토리로 돌아와서, 해당 명령어를 실행합니다

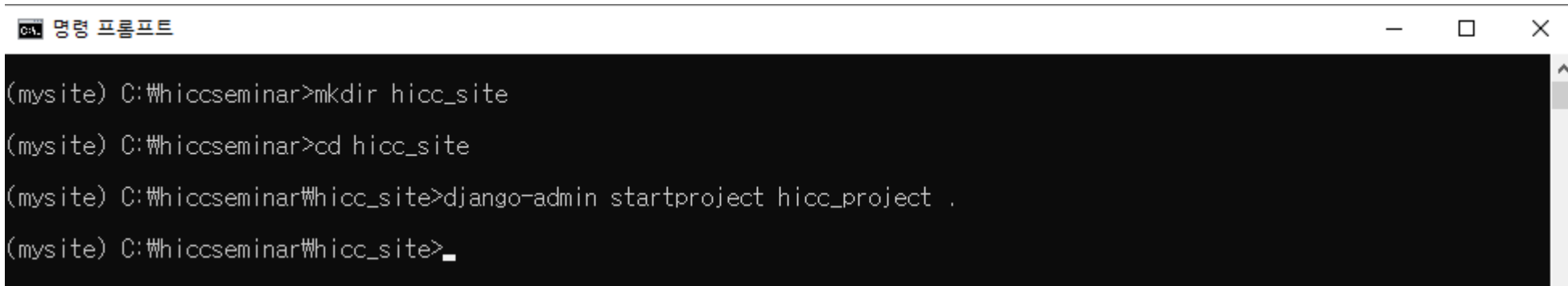
해당 세미나에서는 **C:\hiccseminar** 디렉토리를 모든 프로젝트들의 루트 디렉토리로 정하고 진행합니다

# 장고 프로젝트 설치하기

```
source ~/venvs/mysite/bin/activate
```

```
C:\hiccseminar>C:\venvs\mysite\Scripts\activate_
```

해당 명령어를 통해 projects 폴더 안에서 가상 환경에 진입



```
(mysite) C:\hiccseminar>mkdir hicc_site
(mysite) C:\hiccseminar>cd hicc_site
(mysite) C:\hiccseminar\hicc_site>django-admin startproject hicc_project .
(mysite) C:\hiccseminar\hicc_site>
```

해당 명령어를 통해 실습에서 진행할 프로젝트를 의미하는 hicc\_site 디렉토리를 생성

`django-admin startproject hicc_project .` 명령으로 장고 프로젝트를 생성

hicc\_project는 환경 설정을 의미하는 폴더 이름이고, 뒤의 .은 현재 디렉토리인 hicc\_site를 기준으로 생성한다는 의미

# 서버 구동해보기

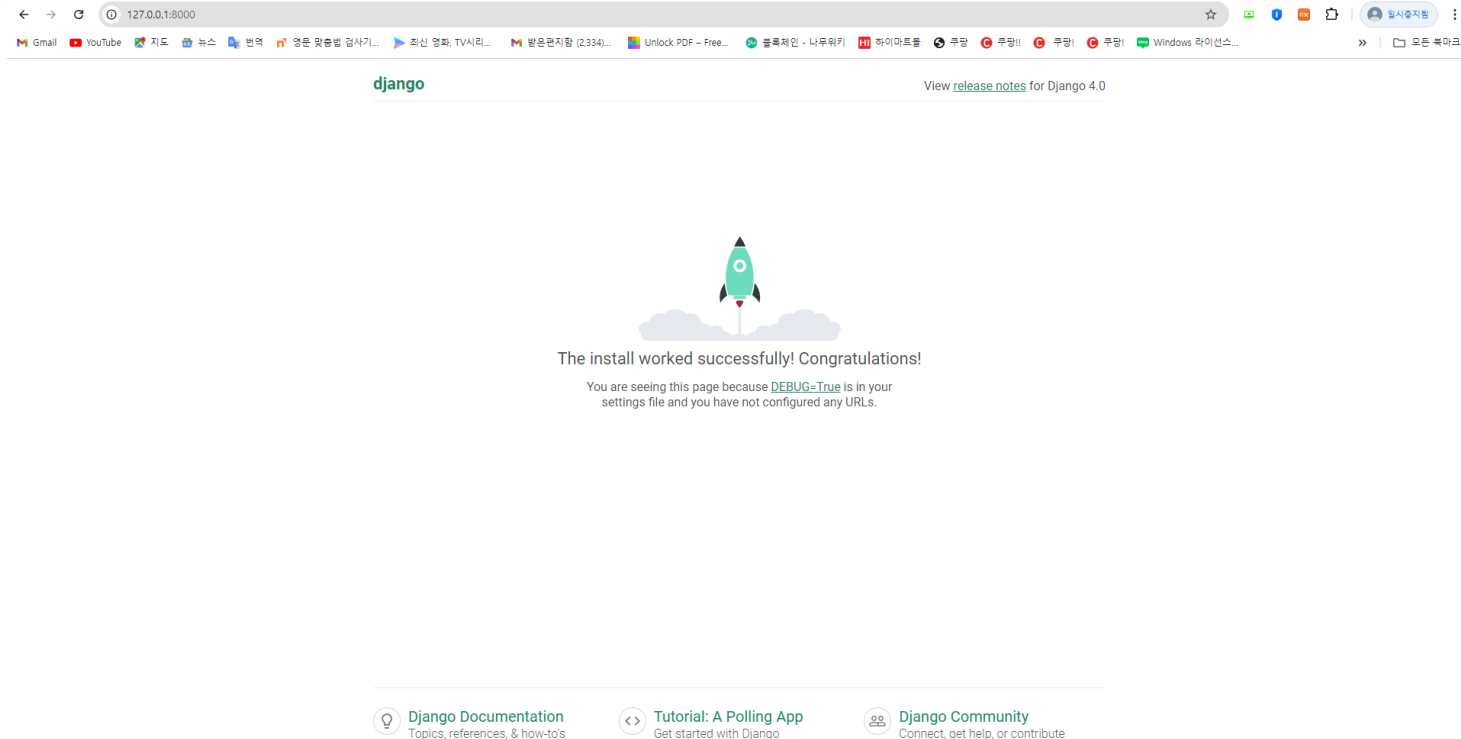
```
(mysite) C:\hiccseminar\hicc_site>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin,
auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
March 25, 2025 - 02:52:32
Django version 5.1.7, using settings 'hicc_project.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

`python manage.py runserver` 명령어를 통해 서버를 구동

# 서버 구동해보기

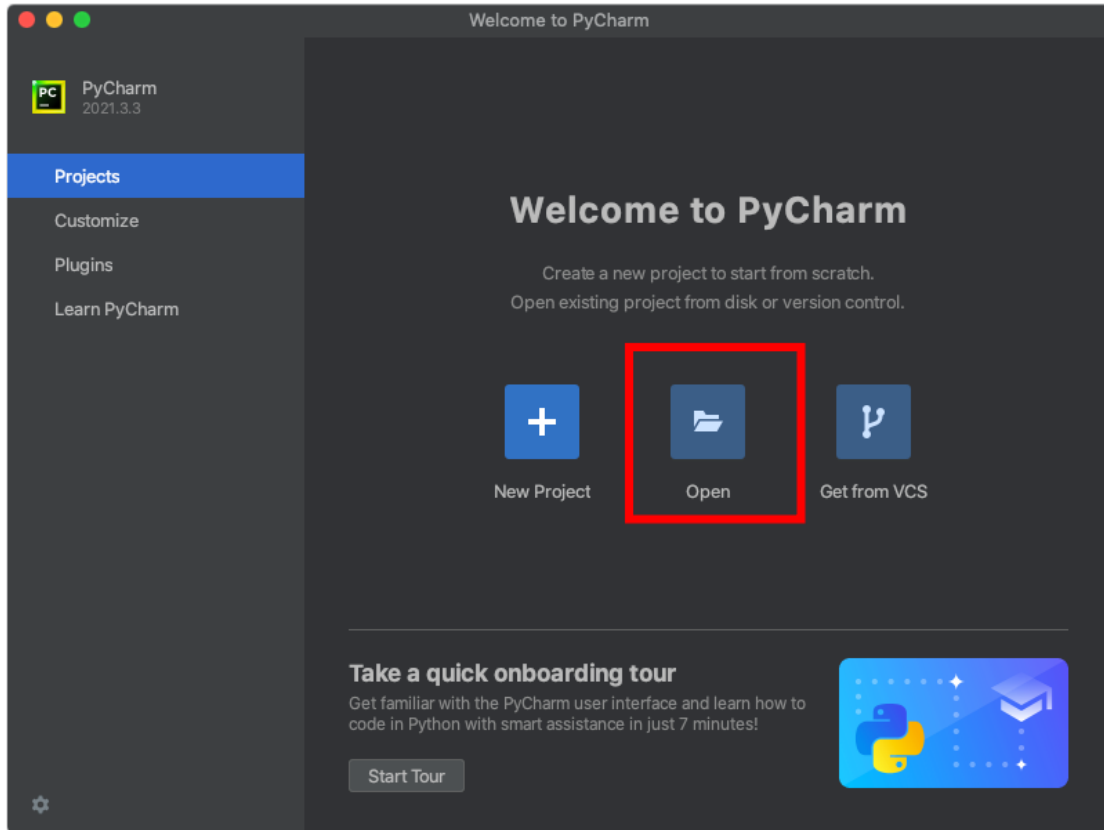


<http://127.0.0.1:8000/>

링크에 접속했을 때 이런 화면이 나오면 서버 구동 성공

이런식으로 내 컴퓨터 위에 개발용으로 켜는 서버를 "로컬 서버"라고 부름, 다른 사람들은 해당 url에 접근 불가

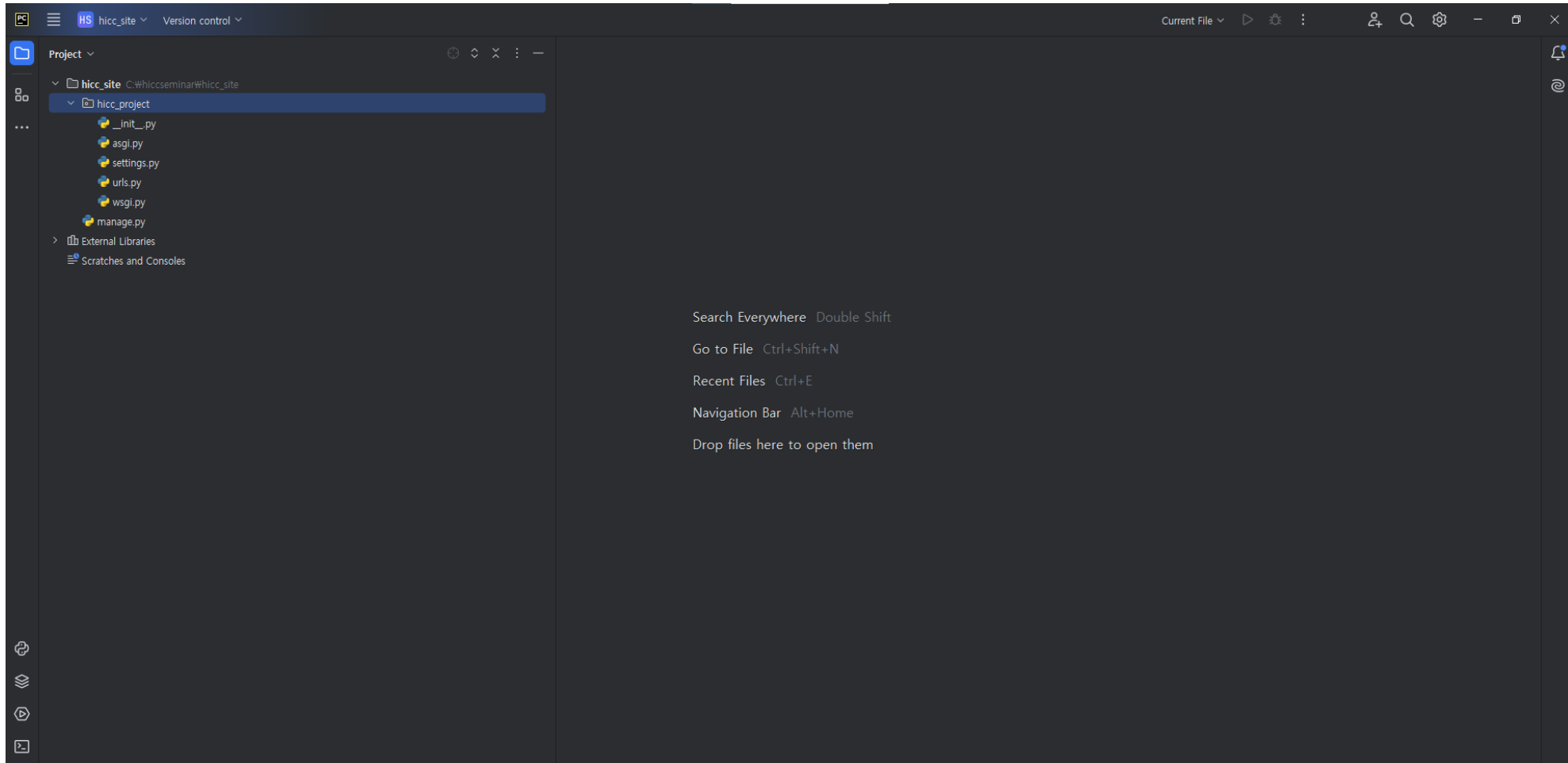
# 파이참으로 프로젝트 열기



여기서 open을 선택하여 앞에서 우리가 생성한 장고 프로젝트인 `C:/projects/hicc_site` 를 선택

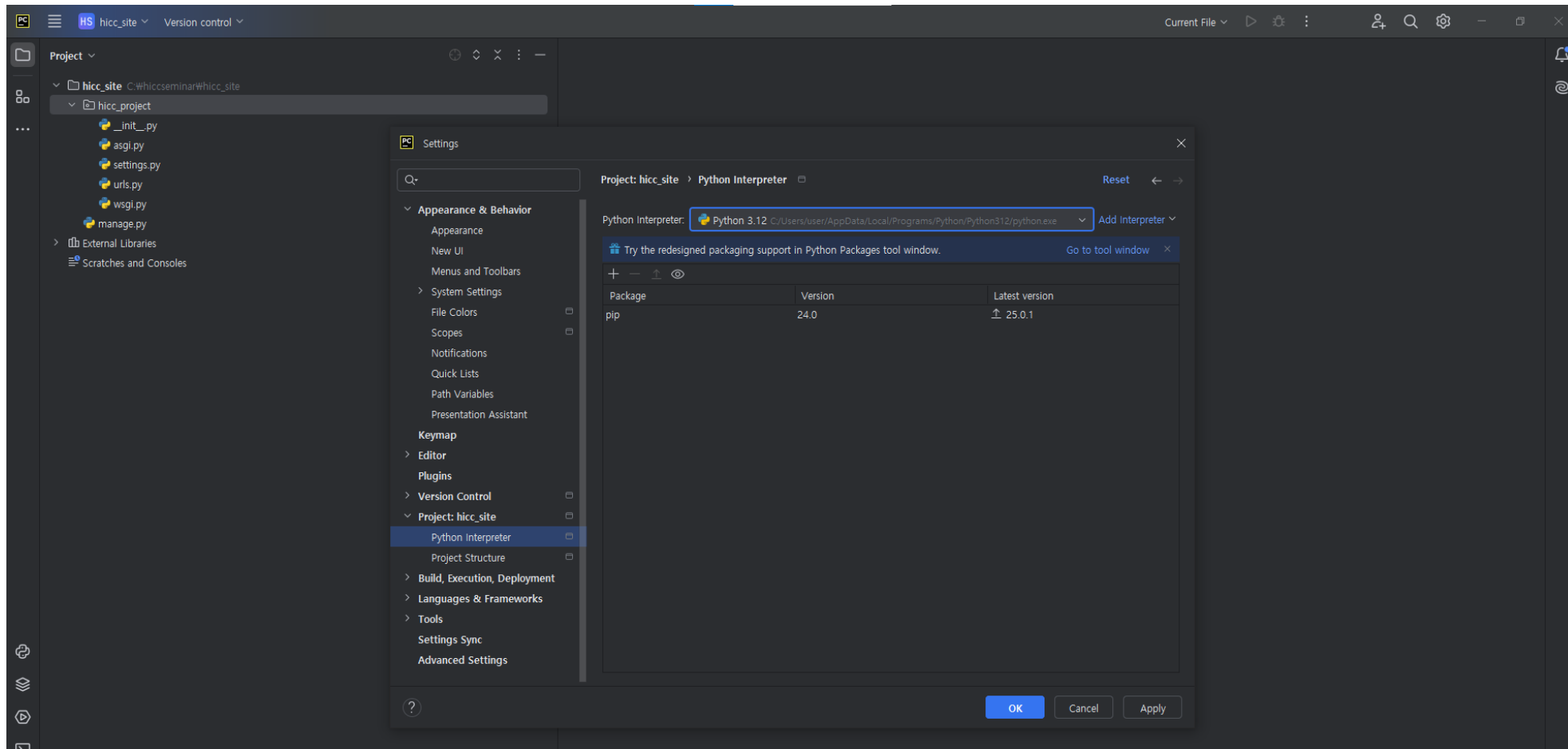


# 파이참으로 프로젝트 열기



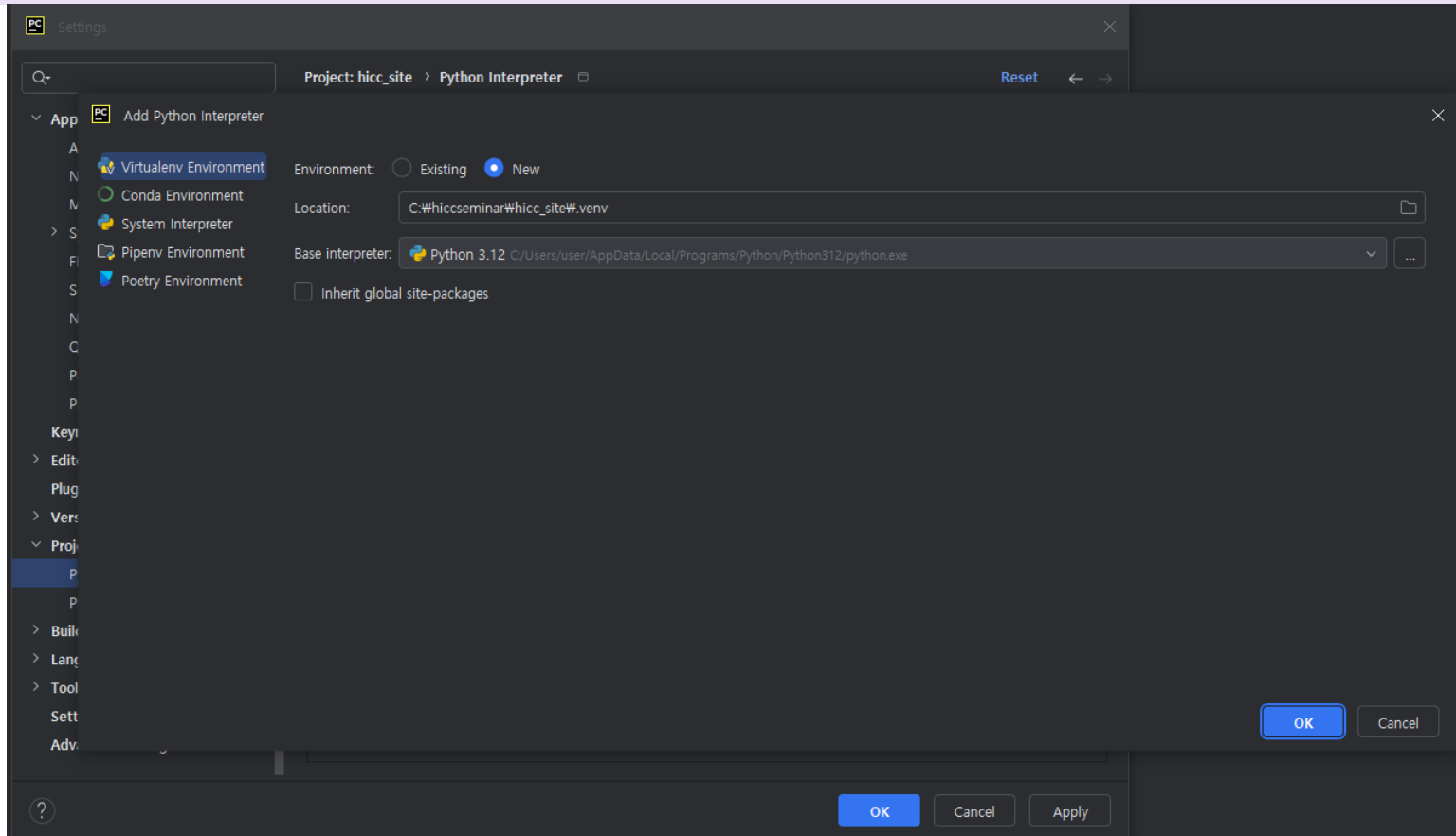
화면이 이렇게 나오면 성공

# 파이썬 인터프리터 설정



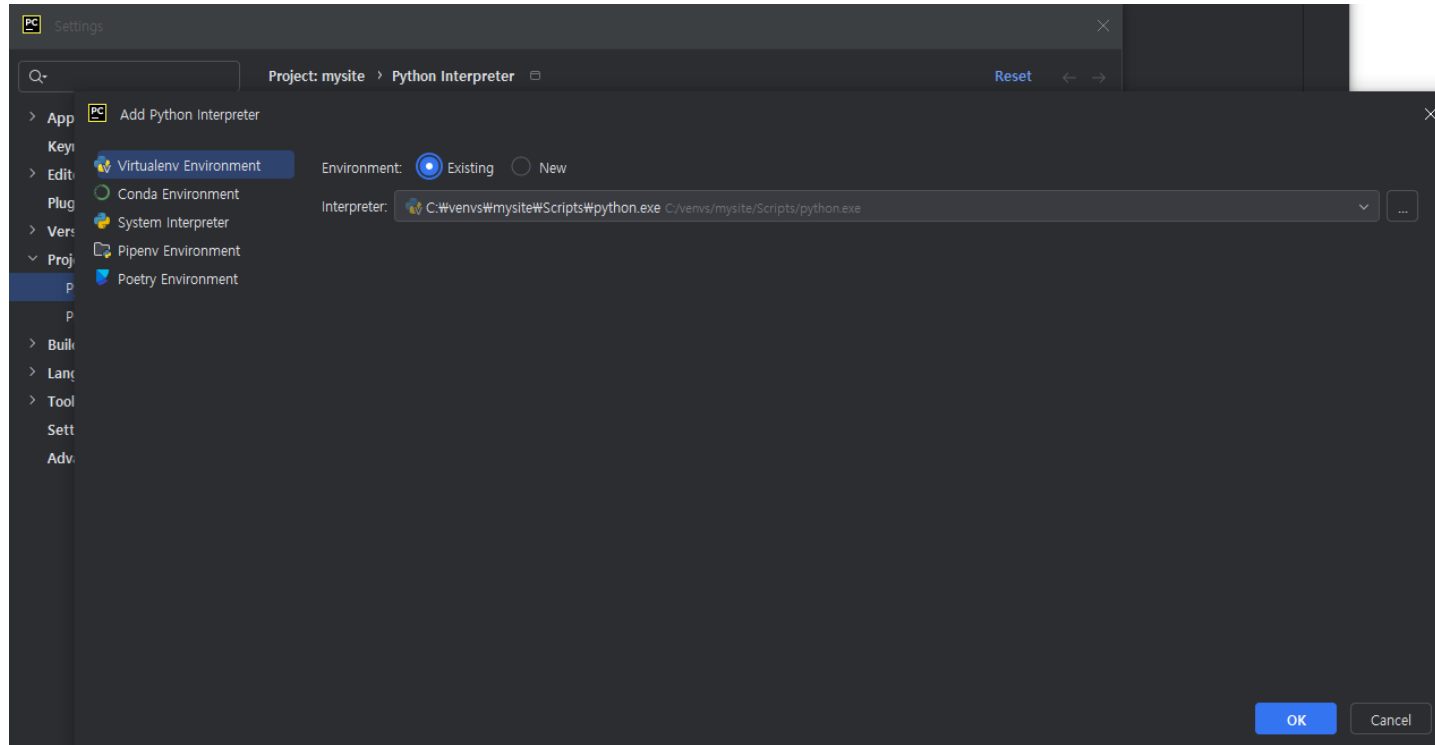
우측 상단의 톱니바퀴 버튼을 클릭 후 setting 클릭 후  
Python Interpreter 클릭 후 Add Interpreter – Add Local Interpreter 클릭

# 파이썬 인터프리터 설정



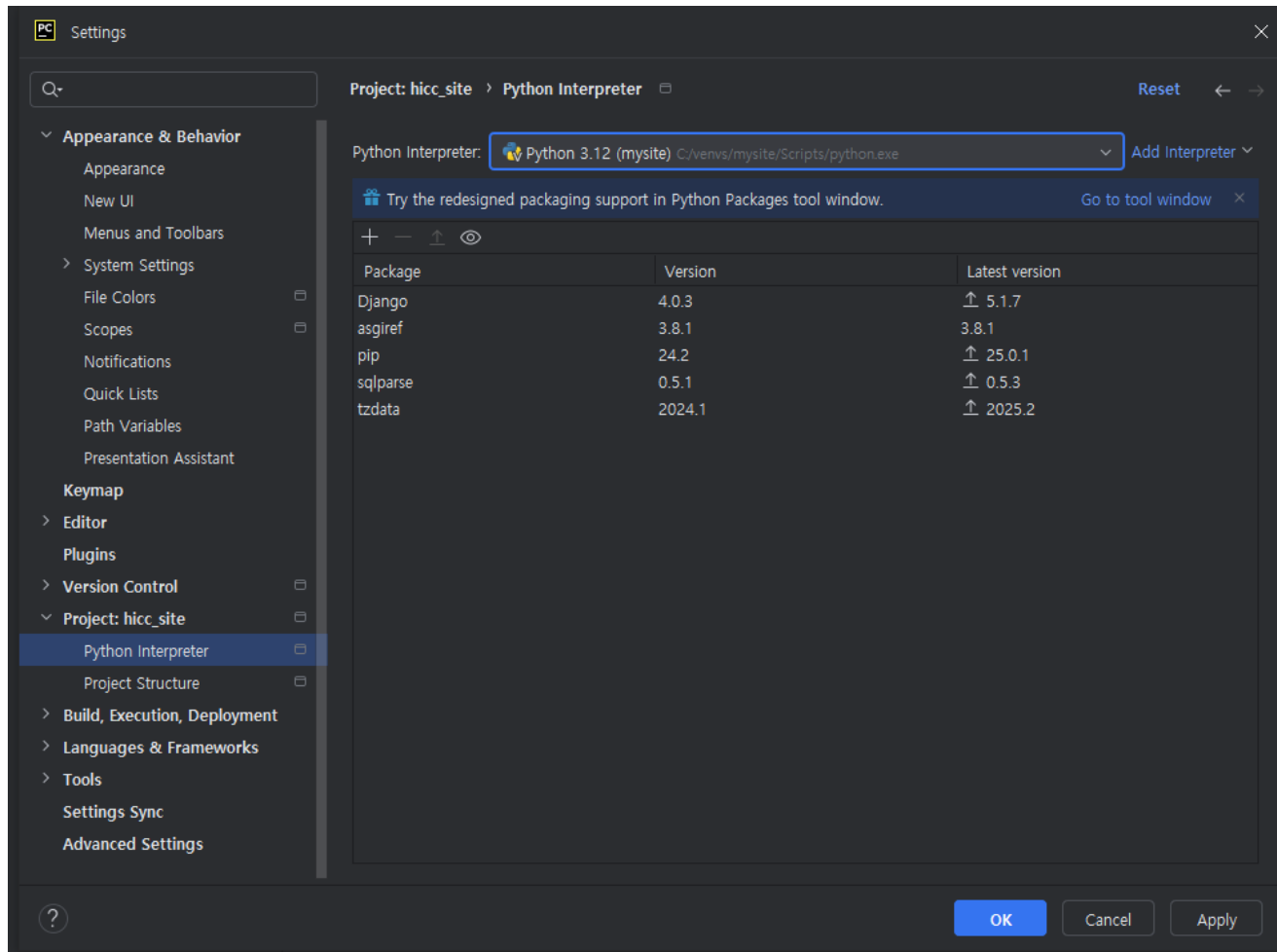
Existing 클릭 후 ... 버튼 클릭

# 파이썬 인터프리터 설정



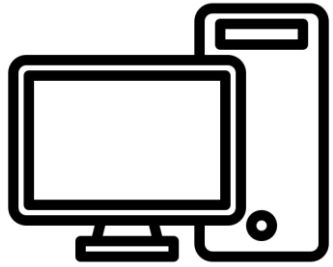
아까 설치했던 가상환경의 파이썬 파일 선택 후 OK 누르기

# 파이썬 인터프리터 설정



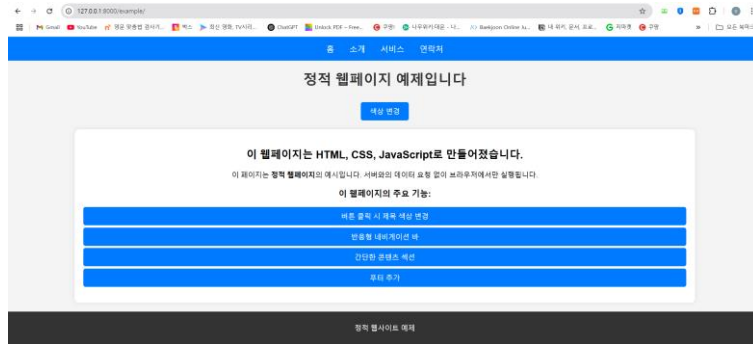
화면이 이처럼 나오면 성공! Apply 버튼 누르고 OK 누르기

# 웹 세계에서 일어나는 일 (정적)



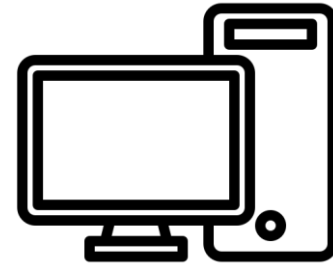
내 컴퓨터

3. 브라우저에서 코드 실행



1. URL 입력

2. 프론트엔드 코드 전송

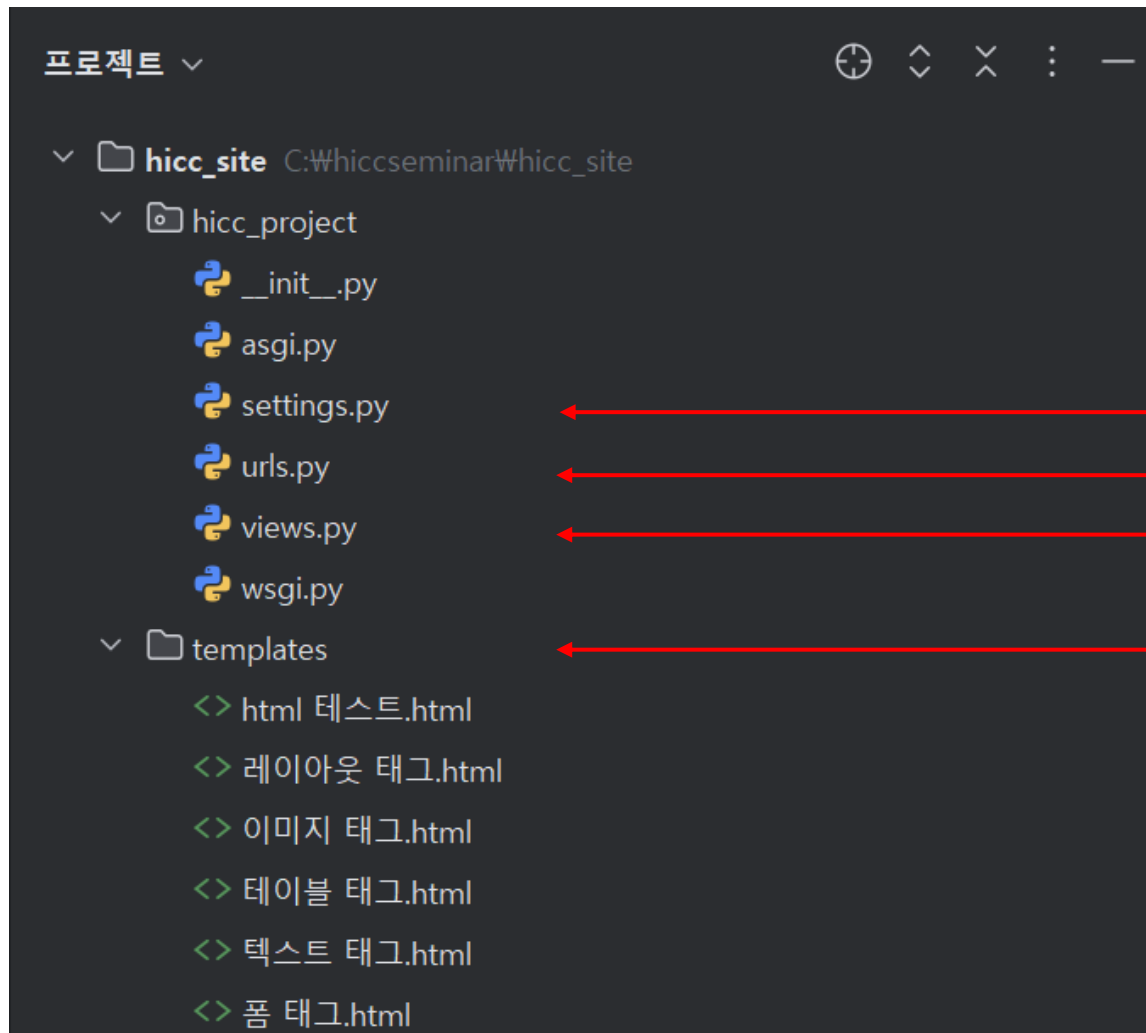


서버

렌더링(rendering)은 컴퓨터 프로그램을 사용하여 이미지나 영상을 만드는 과정을 말합니다.

프로그래밍에선 위와 같은 과정을 '렌더링'이라고 합니다.

# Django 프로젝트 구조



← 프로그램 설정을 담아둔 파일

← url 경로를 설정하는 파일

← 특정 url을 입력했을 때 브라우저로 코드를 전송하는 파일

← 브라우저 쪽으로 보내줄 html 코드들을 모아둔 폴더

# 설정 코드 수정

```
54  TEMPLATES = [  
55      {  
56          'BACKEND': 'django.template.backends.django.DjangoTemplates',  
57          'DIRS': [BASE_DIR / 'templates'],  
58          'APP_DIRS': True,  
59          'OPTIONS': {  
60              'context_processors': [  
61                  'django.template.context_processors.debug',  
62                  'django.template.context_processors.request',  
63                  'django.contrib.auth.context_processors.auth',  
64                  'django.contrib.messages.context_processors.messages',  
65              ],  
66          },  
67      },  
68  ]  
69
```

제일 기본이 되는 폴더의 이름을 'templates'로 설정한다는 의미



# views.py 코드 수정



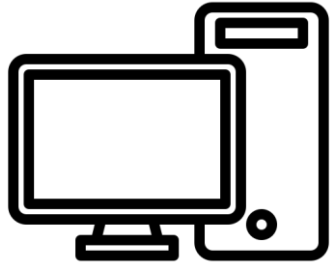
views.py

views.py는 유저의 요청을 받았을 때, html 코드 혹은 가공한 데이터 등을 보내주는 등 요청에 맞는 어떠한 결과를 응답하는 중심적인 역할을 하는 파일입니다.

웹 프로그램이 작동하는데 있어서 대부분의 기능을 해당 파일에 코드로 정의한다고 보면 됩니다.

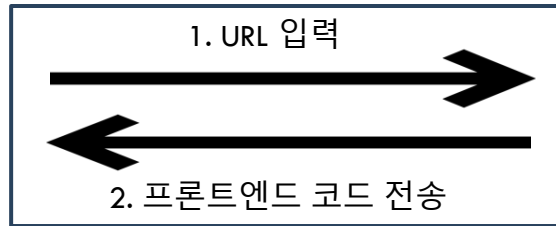
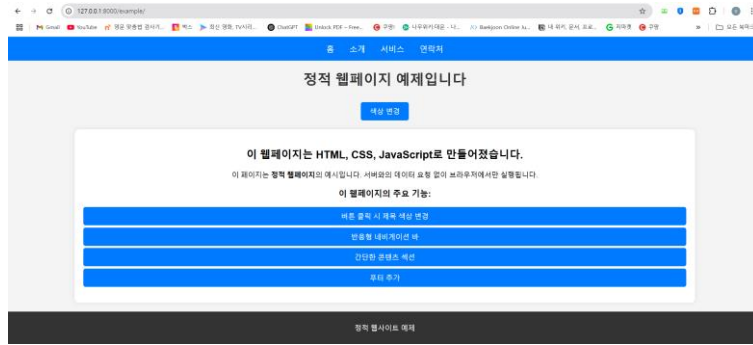
이번 챕터에서는 기능 중 하나인 정적 페이지 렌더링 부분에 대해서만 다루도록 하겠습니다.

# 웹 세계에서 일어나는 일 (정적)

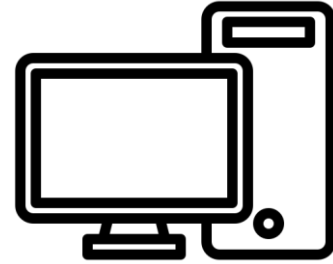


내 컴퓨터

3. 브라우저에서 코드 실행



내 컴퓨터가 서버에 요청(request)을 보내고,  
서버는 내 컴퓨터로 응답(response)을 보냅니다.



서버

# views.py 코드 수정

```
from django.shortcuts import render

def html_test_render(request) :
    return render(request, 'html 테스트.html')

def image_tag_render(request) :
    return render(request, '이미지 태그.html')

def text_tag_render(request) :
    return render(request, '텍스트 태그.html')

def table_tag_render(request) :
    return render(request, '테이블 태그.html')

def form_tag_render(request) :
    return render(request, '폼 태그.html')

def layout_tag_render(request) :
    return render(request, '레이아웃 태그.html')
```

# urls.py 코드 수정



urls.py

urls.py는 웹 페이지를 개발할 때, 특정 url과 views.py 파일의 함수들을 연결하는 역할을 합니다.

예시)

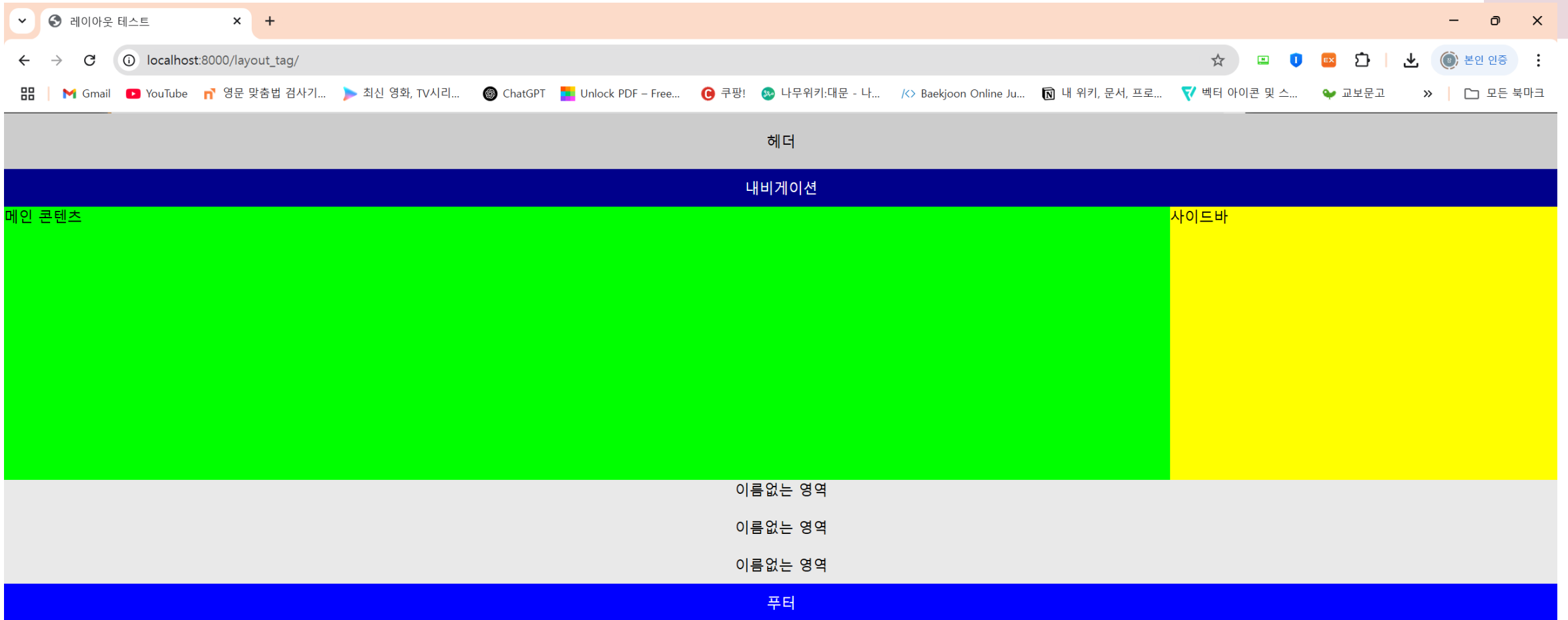
views.py 파일에 로그인 페이지를 렌더링 하는 login 함수를 정의했다고 가정합니다. 해당 상황에서 urls.py 파일의 역할은 [www.naver.com/login](http://www.naver.com/login) 페이지로 이동했을 때 login 함수를 실행시켜서 로그인 페이지를 렌더링하게 하도록 설정할 수 있습니다.

# urls.py 코드 수정

```
from django.contrib import admin
from django.urls import path
from hicc_project.views import text_tag_render, html_test_render, image_tag_render,
form_tag_render, table_tag_render, layout_tag_render

urlpatterns = [
    path('admin/', admin.site.urls),
    path('html_test/', html_test_render),
    path('text_tag/', text_tag_render),
    path('image_tag/', image_tag_render),
    path('form_tag/', form_tag_render),
    path('table_tag/', table_tag_render),
    path('layout_tag/', layout_tag_render), ]
```

# 정적 페이지 렌더링



# 텍스트 태그 파일 수정해서 연결하기

</ol>

<a href="/form\_tag/">등록 하기</a>

<script>

    alert("텍스트 태그입니다.");

    var name = prompt("이름: ");

    document.write(name + "님, 어서오세요!");

</script>

</body>

-- url에 localhost:8000 부분은 빼줘야 됨

# 텍스트 태그 파일 수정해서 연결하기

</ol>

<a href="/form\_tag/">등록 하기</a>

<script>

    alert("텍스트 태그입니다.");

    var name = prompt("이름: ");

    document.write(name + "님, 어서오세요!");

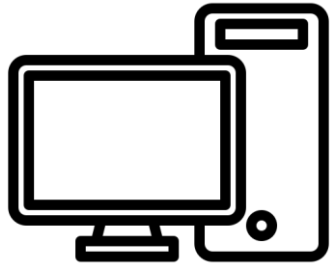
</script>

</body>

-- url에 localhost:8000 부분은 빼줘야 됨

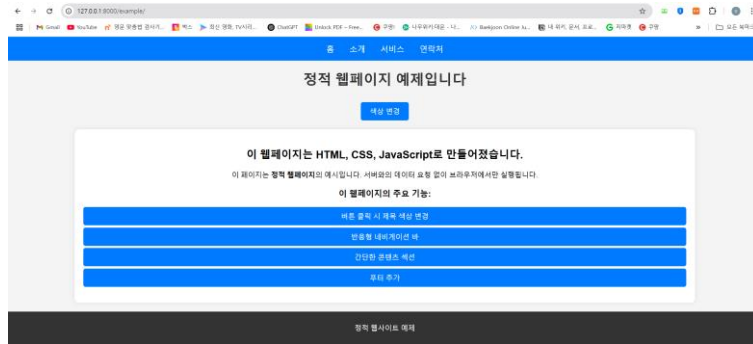


# 웹 세계에서 일어나는 일 (정적)



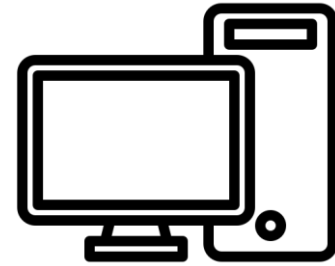
내 컴퓨터

3. 브라우저에서 코드 실행



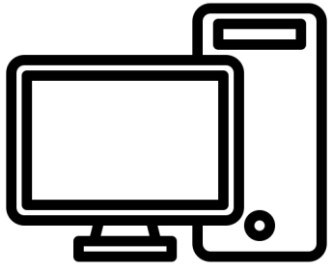
1. URL 입력

2. 프론트엔드 코드 전송



서버

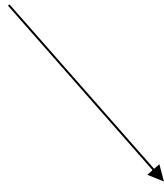
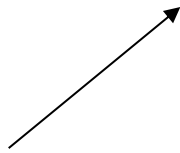
# 웹 세계에서 일어나는 일 (동적)



내 컴퓨터

3. 브라우저에서 코드 실행

7. 실행된 결과 띄우기

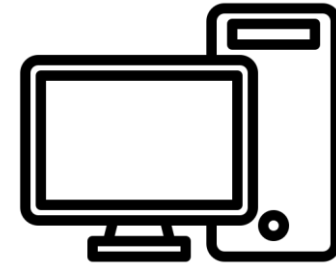
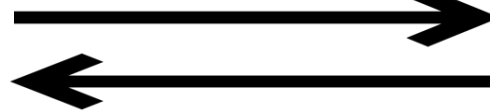


질문 게시판		
번호	제목	작성일시
<a href="#">질문 등록하기</a>		

4. 백엔드 서버에 api 요청

5. 서버에서 코드 실행

6. api 응답 데이터 전송



서버

질문 게시판		
번호	제목	작성일시
3	<a href="#">HICC는 무엇의 약자인가요?</a>	2025. 3. 11. 오전 3:53:20
4	<a href="#">HICC 동방은 몇호인가요?</a>	2025. 3. 11. 오전 4:25:32
<a href="#">질문 등록하기</a>		

# 다음주 예고

1주차. 웹 통신과 개발의 이해 (OT)

2주차. HTML, CSS, JS 개념 기초

3주차. 웹 프레임워크 이해 (Django를 활용한 페이지 랜더링) -- 정적 웹 서비스 구현

4주차. GIT 사용 실습 -- 필수 개발자 도구 익히기

5주차. MVC 패턴 (MTV) + 데이터베이스 기초

6주차. 네트워크 기초 + REST API (1) (Django ORM 하드코딩) -- 동적 웹 서비스 구현

7주차. REST API (2) (템플릿 프론트엔드 연결)

8주차. 최신 웹 개발 스택 맛보기 (DRF + React를 통한 -- 메인스트림 개발 스택을 활용하여 간편하게 개발하기  
웹페이지 개발)

# 다음주 예고

github.com/hicc-seminar-2025/hyeonhakjang/tree/main/hiccseminar/hicc\_site/templates

hicc-seminar-2025 / hyeonhakjang

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Files

main

Go to file

- hiccseminar/hicc\_site
  - hicc\_project
    - \_\_pycache\_\_
      - \_\_init\_\_.py
      - asgi.py
      - settings.py
      - urls.py
      - views.py
      - wsgi.py
    - templates
      - images
        - html 테스트.html
        - 동방.png
        - 레이아웃 태그.html
        - 이미지 태그.html
        - 테이블 태그.html
        - 텍스트 태그.html
        - 폼 태그.html
      - db.sqlite3
      - manage.py

hyeonhakjang / hiccseminar / hicc\_site / templates /

hyeonhakjang Add files via upload 31a272b · now History

Name	Last commit message	Last commit date
..		
images	Add files via upload	now
html 테스트.html	Add files via upload	now
동방.png	Add files via upload	now
레이아웃 태그.html	Add files via upload	now
이미지 태그.html	Add files via upload	now
테이블 태그.html	Add files via upload	now
텍스트 태그.html	Add files via upload	now
폼 태그.html	Add files via upload	now