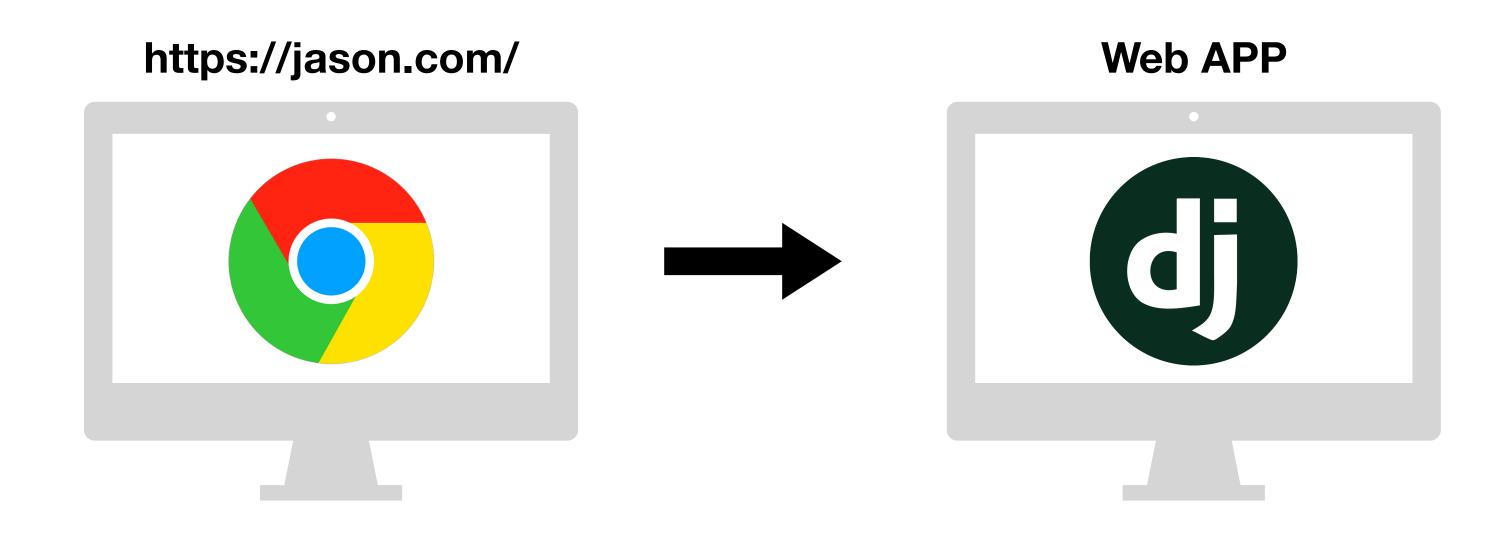
# django



클라이언트가 요청을 보냈을 때

응답을 보낸다

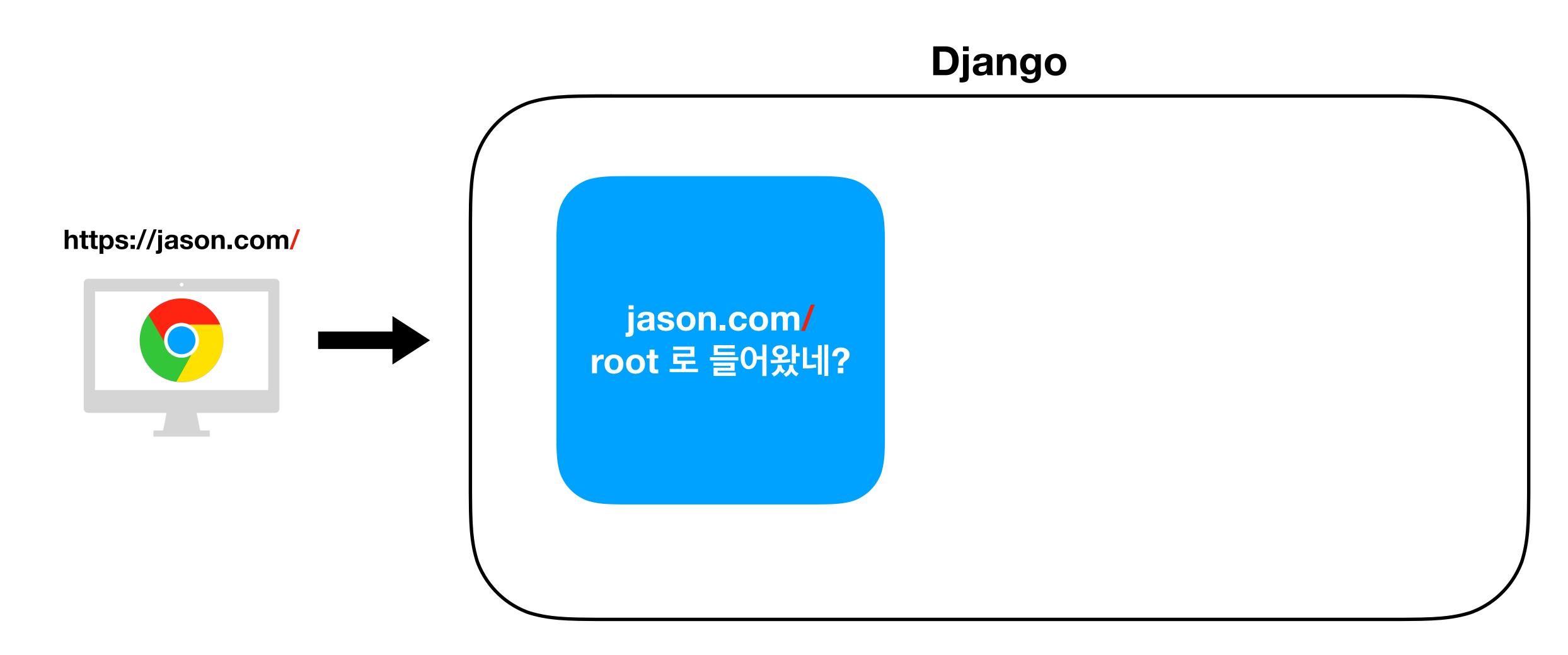
Django 로 페이지 만들기 어떻게 했었죠?

1. 클라이언트의 URL 을 확인한다.

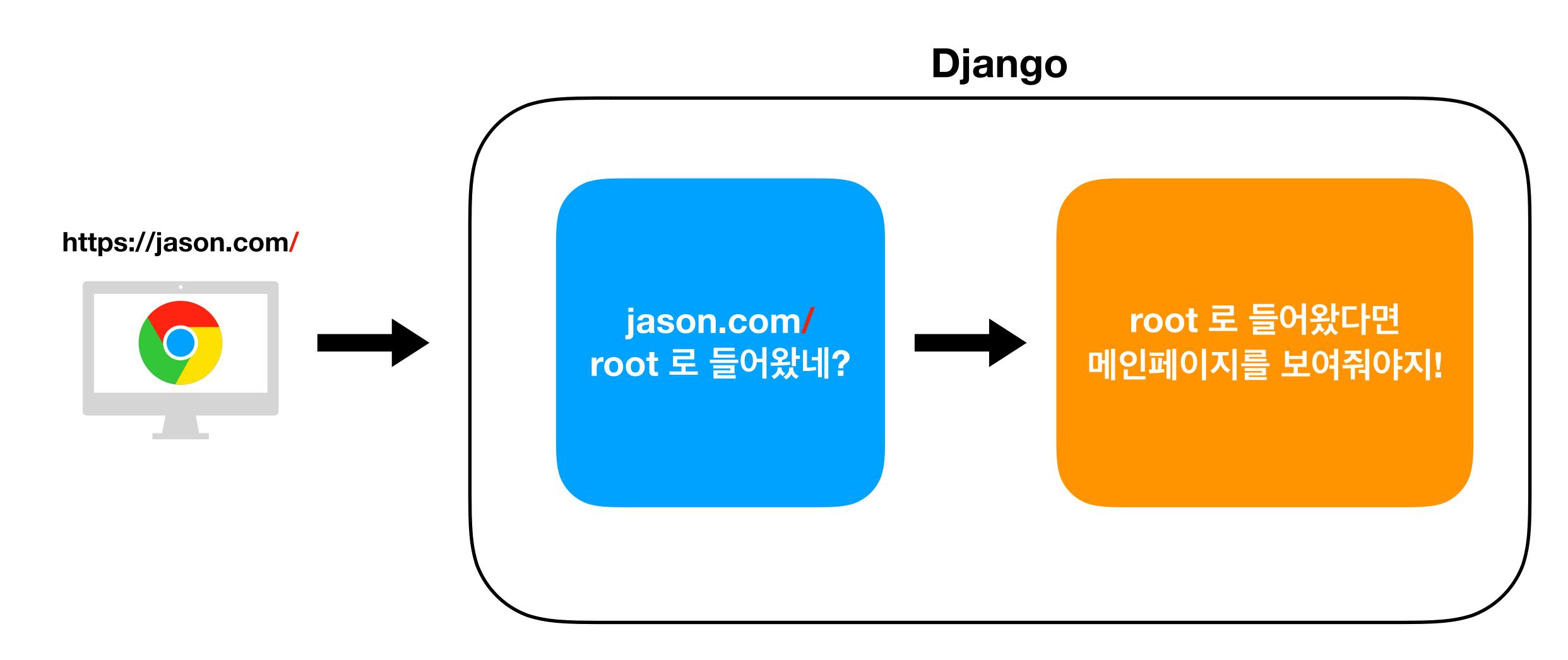
2. VIEW 함수를 실행한다.

3. TEMPLATE 을 렌더한다.

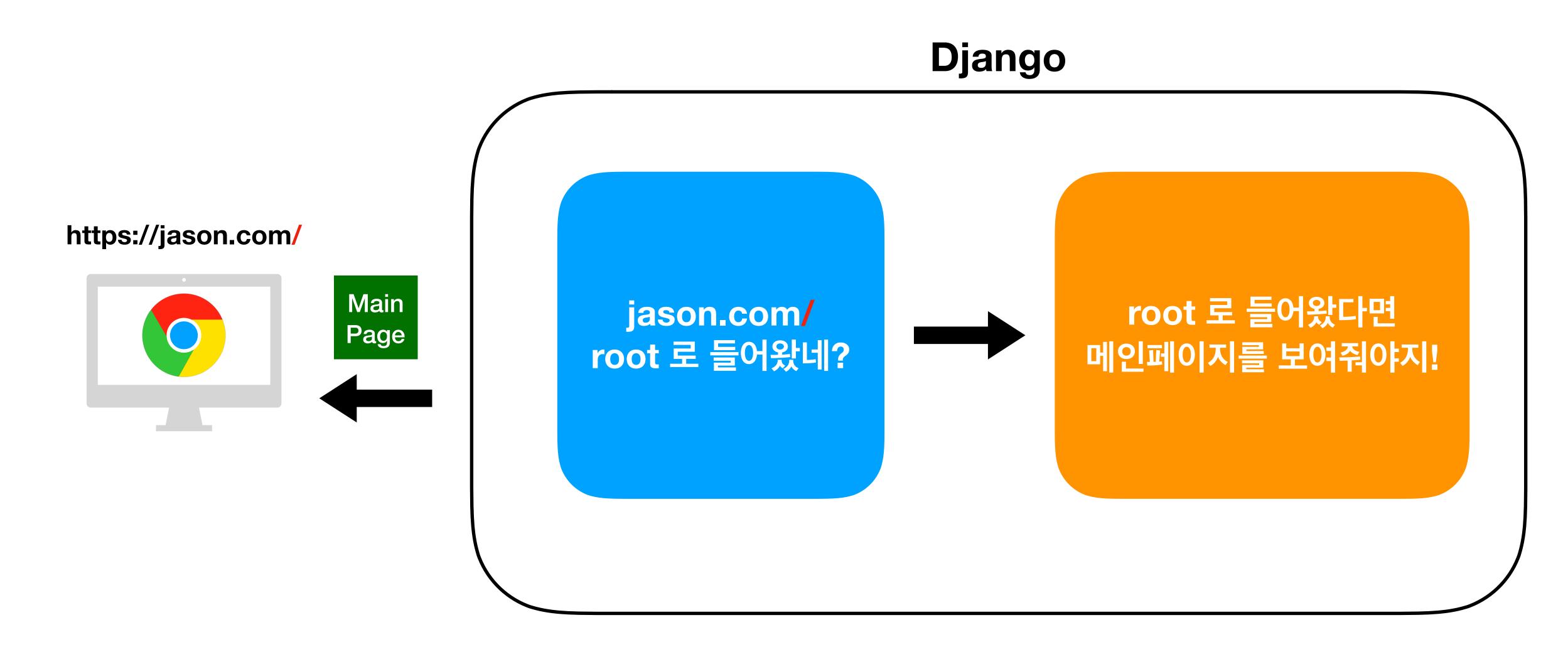
- 1. 클라이언트의 URL 을 확인한다.
- 2. VIEW 함수를 실행한다.
- 3. TEMPLATE 을 렌더한다.



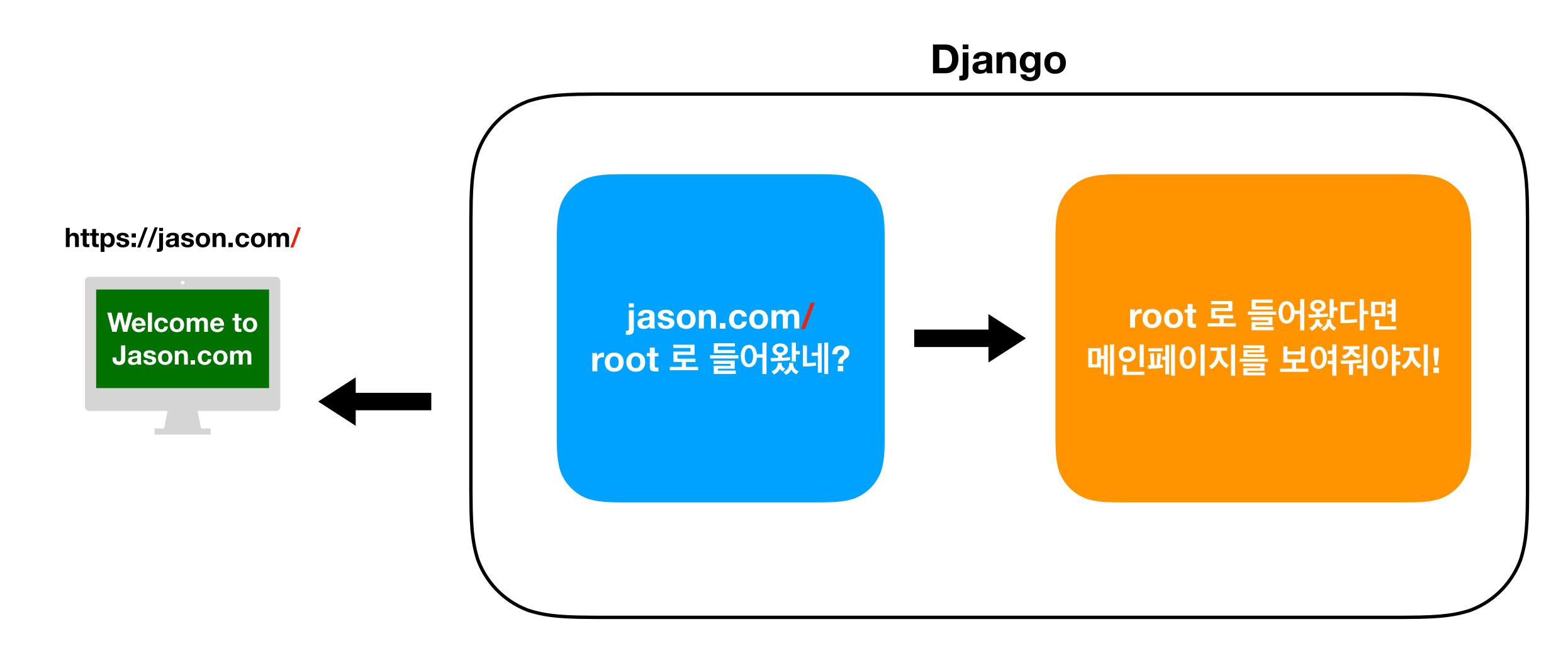
- 1. 클라이언트의 URL 을 확인한다.
- 2. VIEW 함수를 실행한다.
- 3. TEMPLATE 을 렌더한다.



- 1. 클라이언트의 URL 을 확인한다.
- 2. VIEW 함수를 실행한다.
- 3. TEMPLATE 을 렌더한다.



- 1. 클라이언트의 URL 을 확인한다.
- 2. VIEW 함수를 실행한다.
- 3. TEMPLATE 을 렌더한다.



코드로 옮겨볼까요?

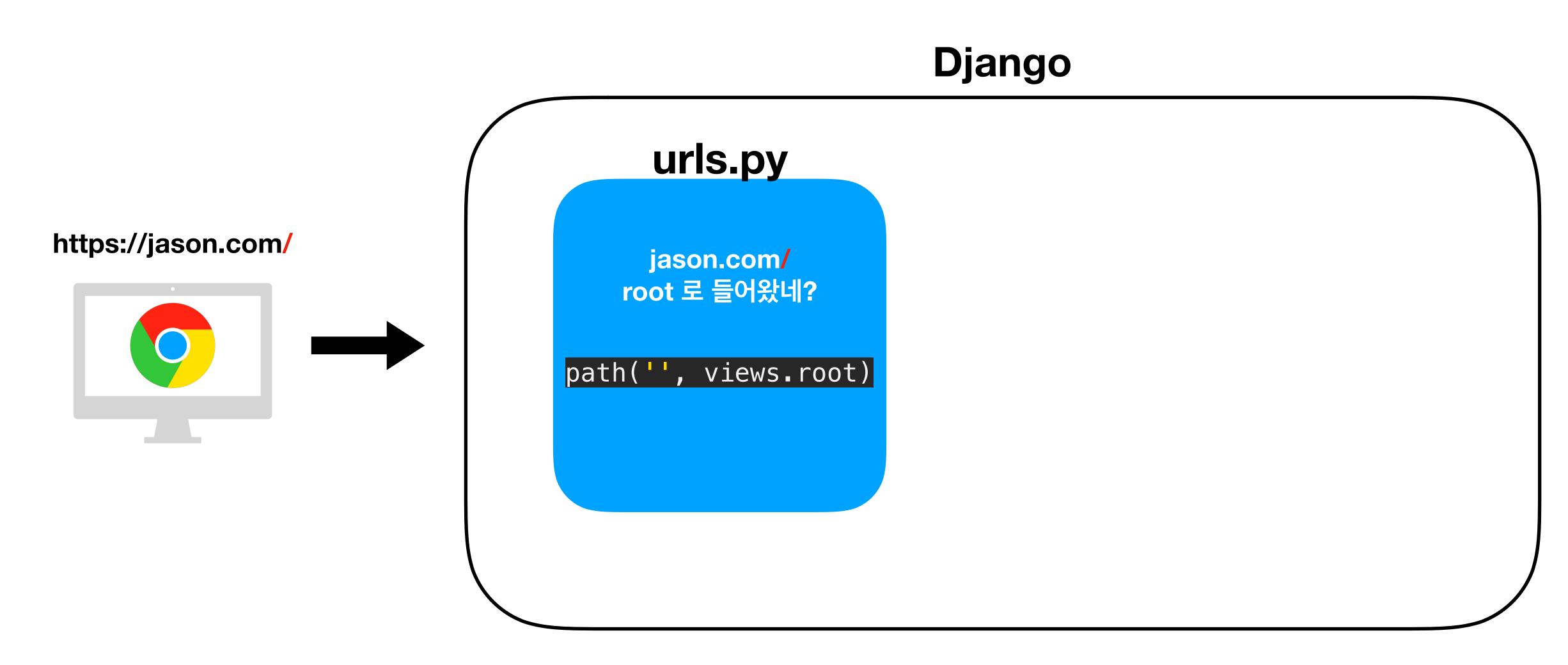
## Django 프로젝트에서는?

1. 클라이언트의 URL 을 확인한다. → urls.py

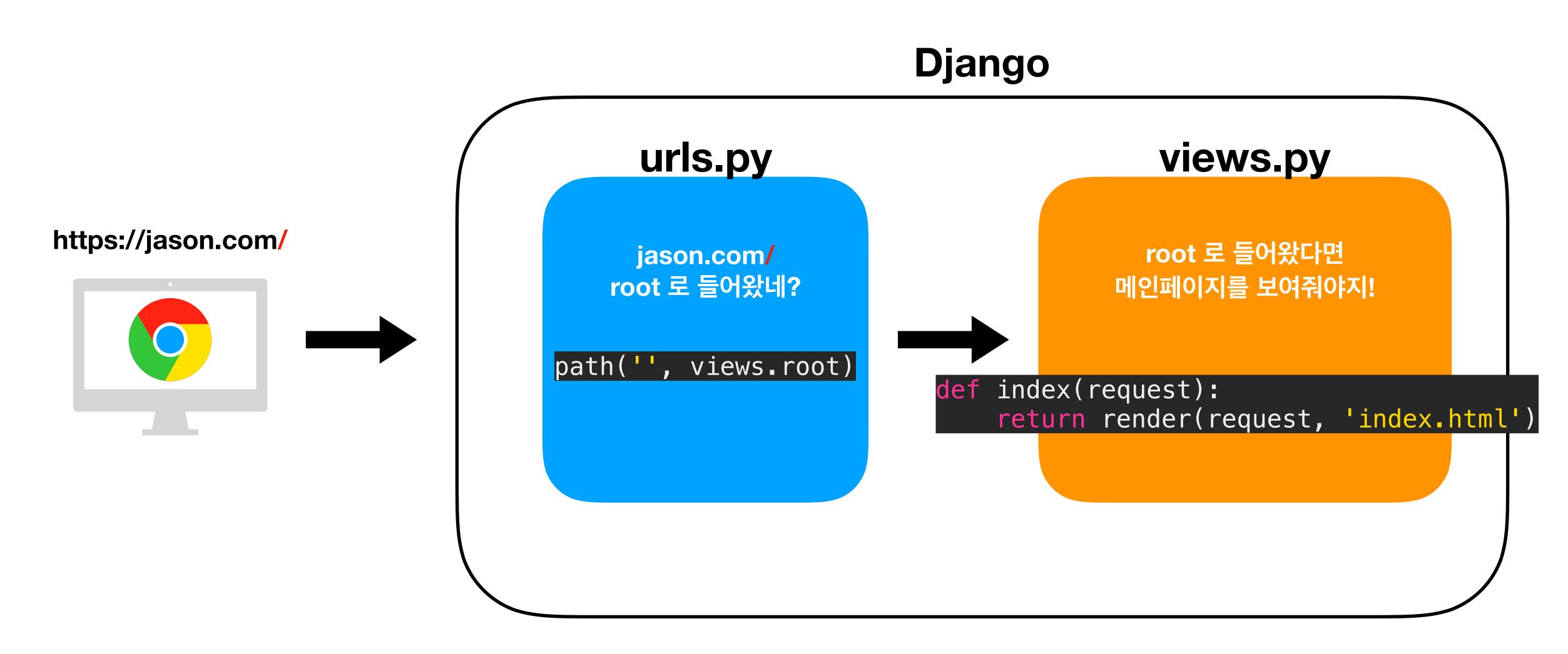
2. VIEW 함수를 실행한다. → views.py

3. TEMPLATE 을 렌더한다. → templates/

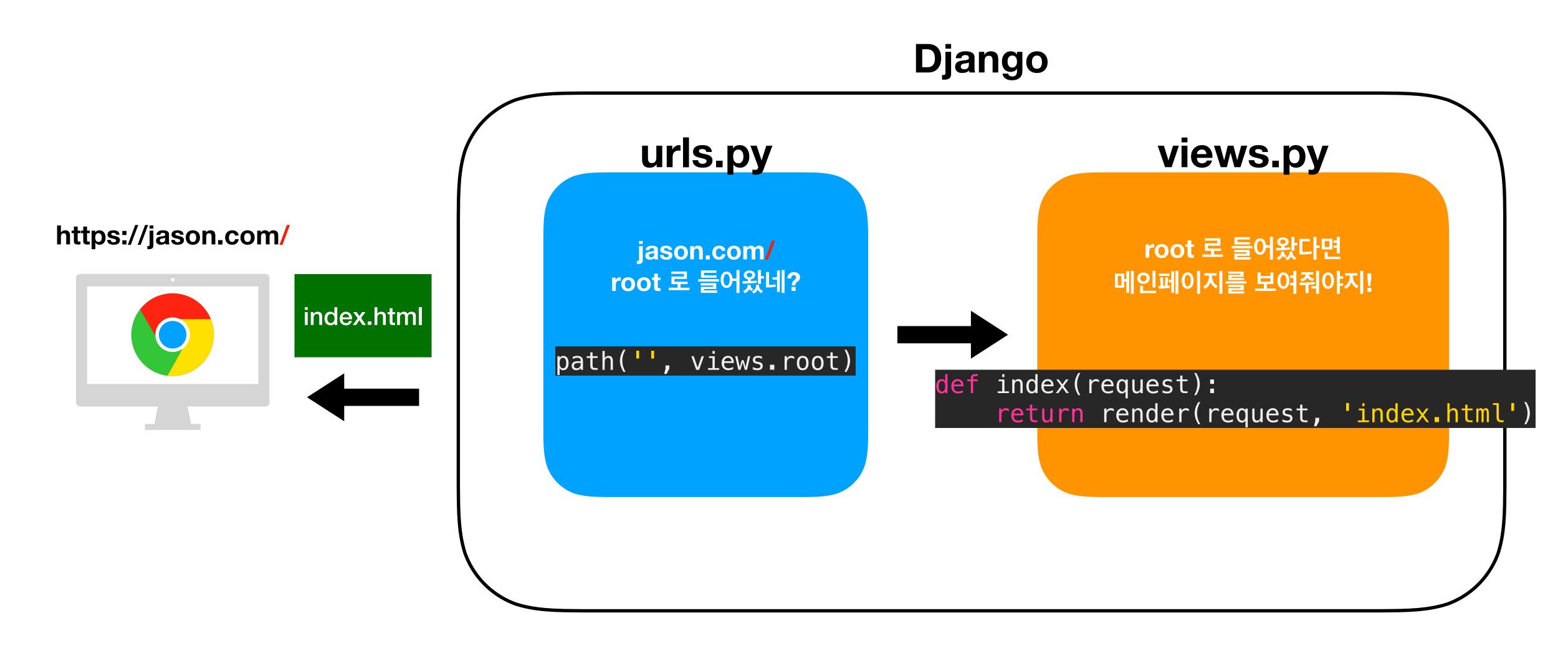
- 1. 클라이언트의 URL 을 확인한다. → urls.py
- 2. VIEW 함수를 실행한다. → views.py
- 3. TEMPLATE 을 렌더한다. → templates/



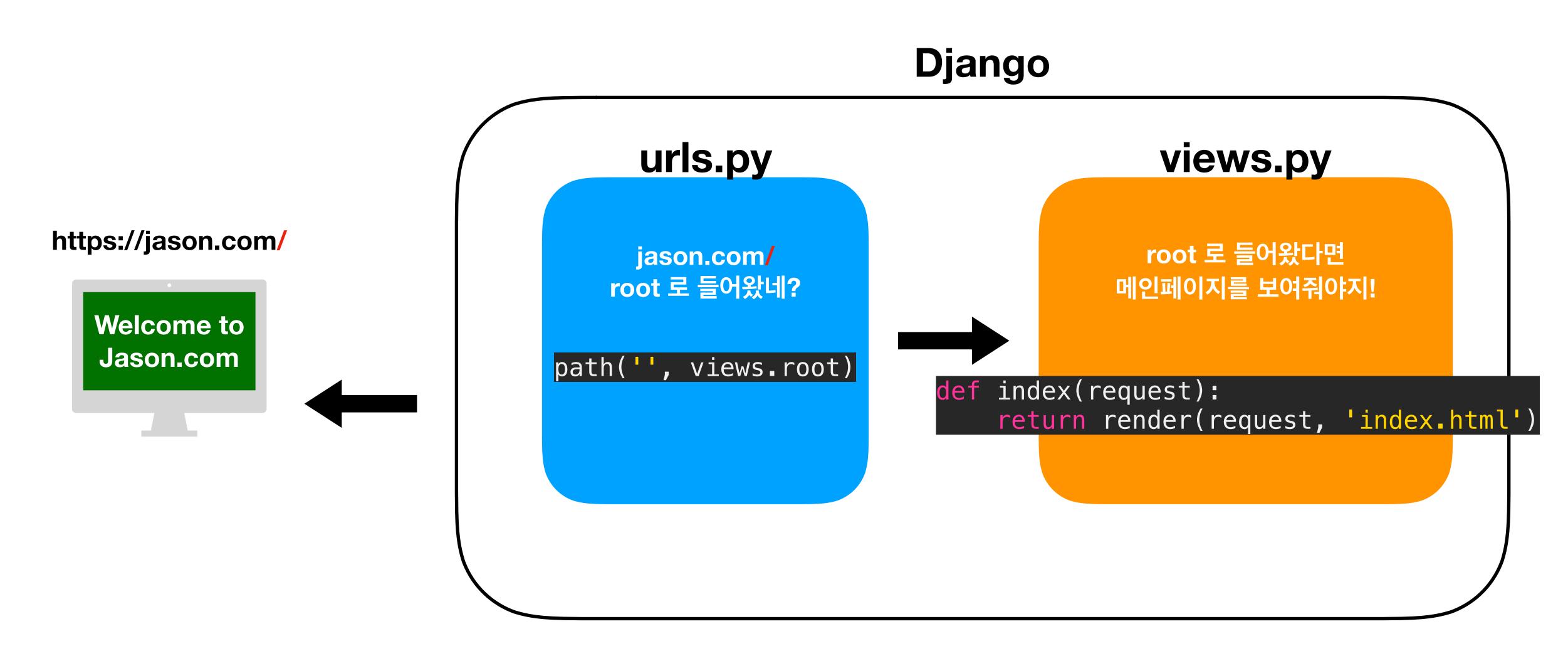
- 1. 클라이언트의 URL 을 확인한다. → urls.py
- 2. VIEW 함수를 실행한다. → views.py
- 3. TEMPLATE 을 렌더한다. → templates/



- 1. 클라이언트의 URL 을 확인한다. → urls.py
- 2. VIEW 함수를 실행한다. → views.py
- 3. TEMPLATE 을 렌더한다. → templates/

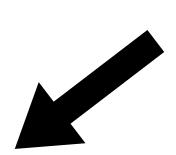


- 1. 클라이언트의 URL 을 확인한다. → urls.py
- 2. VIEW 함수를 실행한다. → views.py
- 3. TEMPLATE 을 렌더한다. → templates/



Variable Renging 件試致죠?

- 1. 클라이언트의 URL 을 확인한다. → urls.py
- 2. VIEW 함수를 실행한다. → views.py
- 3. TEMPLATE 을 렌더한다. → templates/



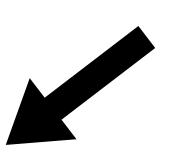


#### urls.py

jason.com/greeting/IU/name 이라는 변수에 IU 값을 가진 그리팅 페이지로 들어왔네?

path('greeting/<:name>/', views.greeting)

- 1. 클라이언트의 URL 을 확인한다. → urls.py
- 2. VIEW 함수를 실행한다. → views.py
- 3. TEMPLATE 을 렌더한다. → templates/





#### urls.py

jason.com/greeting/IU/
name 이라는 변수에 IU 값을 가진
그리팅 페이지로 들어왔네?

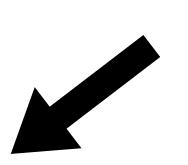
path('greeting/<:name>/', views.greeting)

#### views.py

greeting 은 name 이라는 변수를 URL 에서 받으니까 request 뒤 두번째 인자에 name 인자를 넣어줘야지! 그리고 greeting html 에서 사용할 수 있도록 context 로 넘겨줘야지!

```
def greeting(request, name):
    context = { 'name': name }
    return render(request, 'greeting.html', context)
```

- 1. 클라이언트의 URL 을 확인한다. → urls.py
- 2. VIEW 함수를 실행한다. → views.py
- 3. TEMPLATE 을 렌더한다. → templates/







#### urls.py

jason.com/greeting/IU/
name 이라는 변수에 IU 값을 가진
그리팅 페이지로 들어왔네?

path('greeting/<:name>/', views.greeting)

#### views.py

greeting 은 name 이라는 변수를 URL 에서 받으니까 request 뒤 두번째 인자에 name 인자를 넣어줘야지! 그리고 greeting html 에서 사용할 수 있도록 context 로 넘겨줘야지!

```
def greeting(request, name):
    context = { 'name': name }
    return render(request, 'greeting.html', context)
```

- 1. 클라이언트의 URL 을 확인한다. → urls.py
- 2. VIEW 함수를 실행한다. → views.py
- 3. TEMPLATE 을 렌더한다. → templates/







jason.com/greeting/IU/
name 이라는 변수에 IU 값을 가진
그리팅 페이지로 들어왔네?

path('greeting/<:name>/', views.greeting)

views.py

greeting 은 name 이라는 변수를 URL 에서 받으니까 request 뒤 두번째 인자에 name 인자를 넣어줘야지! 그리고 greeting html 에서 사용할 수 있도록 context 로 넘겨줘야지!

def greeting(request, name):
 context = { 'name': name }
 return render(request, 'greeting.html', context)

## 데이터를 주고 받는 Throw and Catch 페이지도 만들어볼까요?

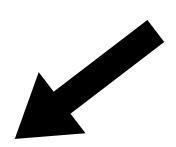
어떻게 한다?

1. 클라이언트의 URL 을 확인한다. → urls.py

2. VIEW 함수를 실행한다. → views.py

3. TEMPLATE 을 렌더한다. → templates/

- 1. 클라이언트의 URL 을 확인한다. → urls.py
- 2. VIEW 함수를 실행한다. → views.py
- 3. TEMPLATE 을 렌더한다. → templates/



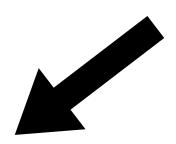


#### urls.py

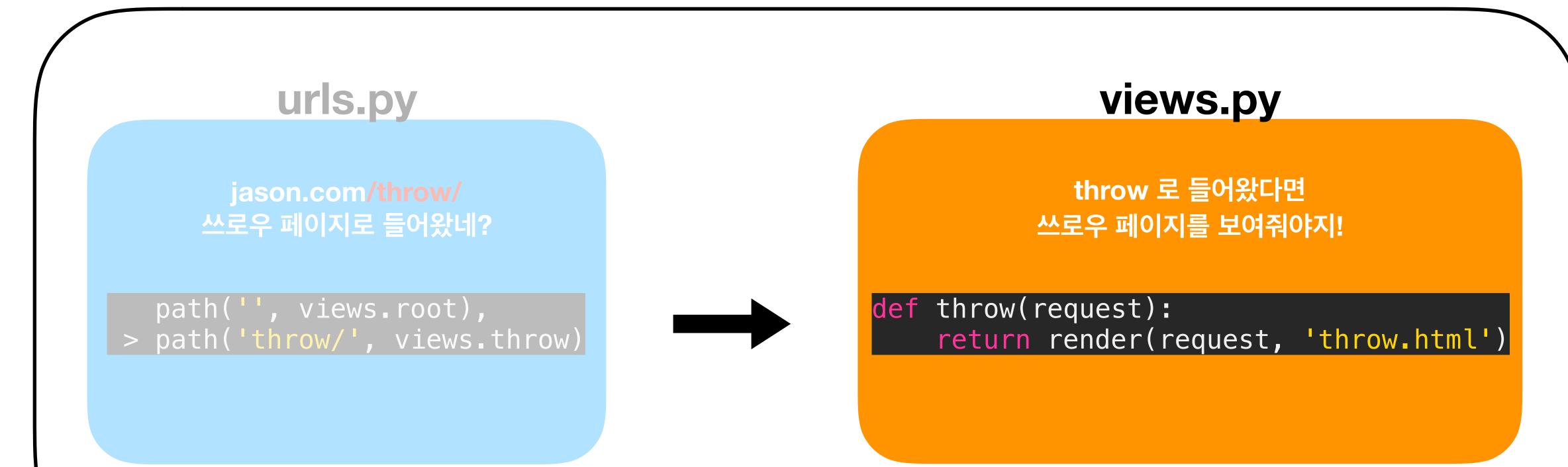
jason.com/throw/ 쓰로우 페이지로 들어왔네?

path('', views.root),
> path('throw/', views.throw)

- 1. 클라이언트의 URL 을 확인한다. → urls.py
- 2. VIEW 함수를 실행한다. → views.py
- 3. TEMPLATE 을 렌더한다. → templates/



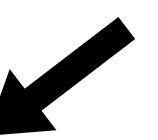




- 1. 클라이언트의 URL 을 확인한다. → urls.py
- 2. VIEW 함수를 실행한다. → views.py
- 3. TEMPLATE 을 렌더한다. → templates/









## urls.py

jason.com/throw/ 쓰로우 페이지로 들어왔네?

path('', views.root), > path('throw/', views.throw) views.py

Django

throw 로 들어왔다면

def throw(request): return render(request, 'throw.html')

- 1. 클라이언트의 URL 을 확인한다. → urls.py
- 2. VIEW 함수를 실행한다. → views.py
- 3. TEMPLATE 을 렌더한다. → templates/







#### urls.py

jason.com/throw/ 쓰로우 페이지로 들어왔네?

path('', views.root),
> path('throw/', views.throw)

#### views.py

throw 로 들어왔다면 쓰로우 페이지를 보여줘야지!

def throw(request):
 return render(request, 'throw.html')

## Throw 페이지는 data 를 입력받아서 Catch 페이지로 data 와 함께 요청을 보내는 페이지에요

### 아래 예시처럼 form tag 를 사용하죠

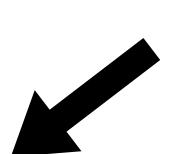
어디로 요청을 보낼건지

### Throw 페이지에 적용시키면 이런 모습이 되겠네요

#### 어디로 요청을 보낼건지

## Throw 페이지에서 전달한 data 를 Catch 페이지에서 출력시켜 볼까요?

- 1. 클라이언트의 URL 을 확인한다. → urls.py
- 2. VIEW 함수를 실행한다. → views.py
- 3. TEMPLATE 을 렌더한다. → templates/





#### urls.py

jason.com/catch/ 캐치 페이지로 들어왔네?

```
path('', views.root),
  path('throw/', views.throw),
> path('catch/', views.catch),
```

- 1. 클라이언트의 URL 을 확인한다. → urls.py
- 2. VIEW 함수를 실행한다. → views.py
- 3. TEMPLATE 을 렌더한다. → templates/





#### urls.py

jason.com/catch/ 캐치 페이지로 들어왔네?

path('', views.root), path('throw/', views.throw), > path('catch/', views.catch),

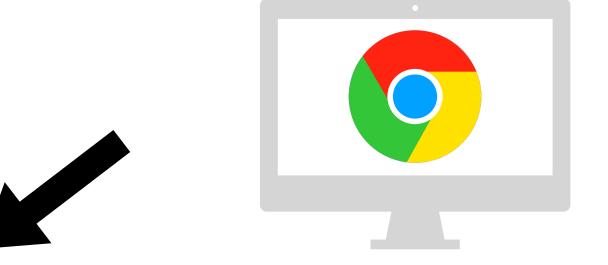


#### views.py

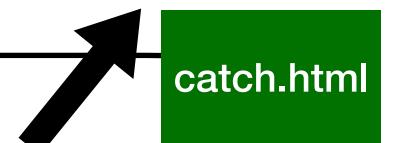
catch 로 들어왔다면 throw 에서 전달한 데이터를 캐치 페이지에서 보여줘야지!

```
catch(request):
thing = request.GET.get('thing')
context = { 'thing': thing }
return render(request, 'catch.html', context)
```

- 1. 클라이언트의 URL 을 확인한다. → urls.py
- 2. VIEW 함수를 실행한다. → views.py
- 3. TEMPLATE 을 렌더한다. → templates/







#### urls.py

jason.com/catch/ 캐치 페이지로 들어왔네?

path('', views.root),
path('throw/', views.throw),
> path('catch/', views.catch),



catch 로 들어왔다면 throw 에서 전달한 데이터를 캐치 페이지에서 보여줘야지!

```
def catch(request):
    thing = request.GET.get('thing')
    context = { 'thing': thing }
    return render(request, 'catch.html', context)
```

- 1. 클라이언트의 URL 을 확인한다. → urls.py
- 2. VIEW 함수를 실행한다. → views.py
- 3. TEMPLATE 을 렌더한다. → templates/





jason.com/catch/ 캐치 페이지로 들어왔네?

path('', views.root),
path('throw/', views.throw),
> path('catch/', views.catch),

views.py

catch 로 들어왔다면 throw 에서 전달한 데이터를 캐치 페이지에서 보여줘야지!

def catch(request):
 thing = request.GET.get('thing')
 context = { 'thing': thing }
 return render(request, 'catch.html', context)