

## 0. 목차

1. 앱 2개 만들기 (articles, pages) + base.html 만들기
2. 각자 앱에 index page 만들기
3. variable routing - hello page 만들기
4. variable routing - number page 만들기
5. throw page 만들기
6. catch page 만들기

## 1. 앱 2개 만들기 (articles, pages) + base.html 만들기

1. articles, pages startapp으로 만들어주고
2. base.html은 이번에는 practice 안에 넣어줄거임
3. DIRS = [BASE\_DIR/'practice'/'templates'], 로 적어주기

## 2. 각자 앱에 index page 만들기

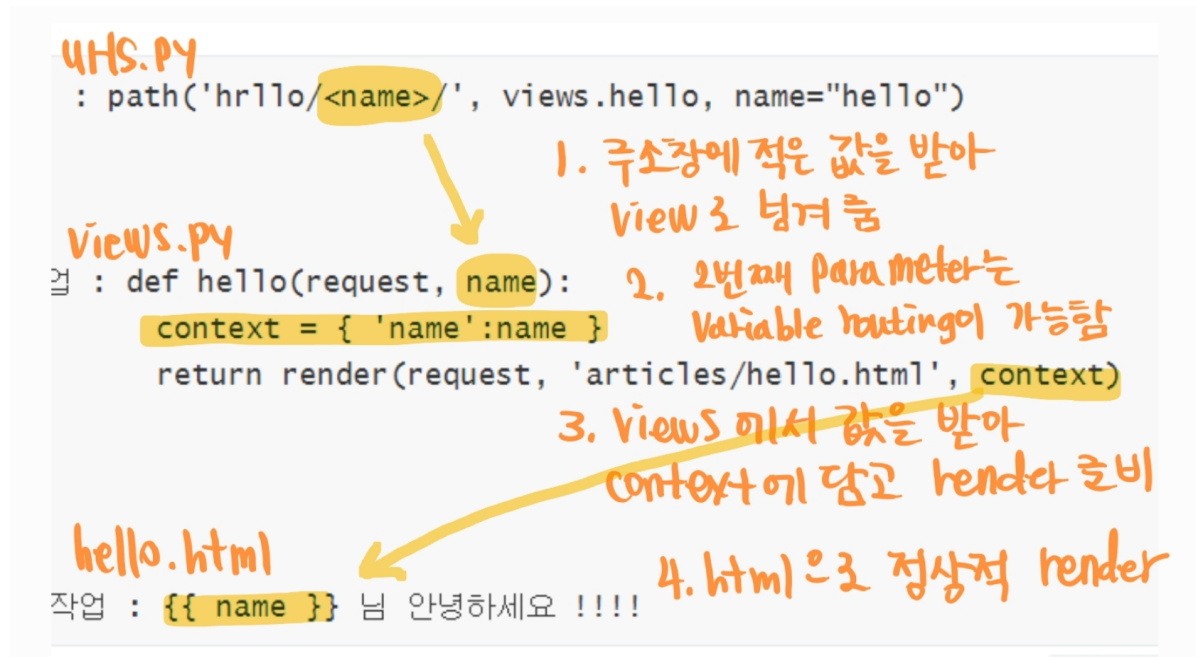
1. practice/urls.py 파일 내 urlspattern에다가 각 앱을 연결해주는 path를 적어줘야됨  
-> include 모듈 임포트  
-> 각 앱의 path를 include를 이용해서 적어주기
2. 각 앱 dir에도 urls.py를 만들어주고  
from django.urls import path  
from . import views  
두가지 모듈 부르기
3. 이제 각자의 urls.py에서 작업을 시작  
templates/앱이름/ 으로 nested된 구조의 템플릿 디렉토리 생성하고 index.html 파일 넣어주기
4. 각 앱의 urls.py 파일 내에 urlpatterns 윗줄에 app\_name = '앱 이름' 기재해주기
5. 각 앱의 urls.py에 path 적어주기  
-> appname 써줬으니까 path 메소드 3번째 요소에 name = 'index' 써줘야됨
6. 각 앱의 views.py에 함수를 만드는데 이때 app 2개고 둘다 index가 있으니, html을 render 할 때 디렉토리도 같이 명시해줘야되는거임  
=> 이런식으로 : return render(request, 'articles/index.html')
7. 각 앱의 templates/앱이름/index.html 파일에 내용 작성해주기

### 3. variable routing - hello page 만들기

1. urls.py 작업 : `path('hrlllo/<name>/', views.hello, name="hello")`
2. views.py 작업 : 

```
def hello(request, name):  
    context = { 'name':name }  
    return render(request, 'articles/hello.html', context)
```
3. hello.html 작업 : `{{ name }}` 님 안녕하세요 !!!!

요청한 값의 흐름이 어떻게 되고 어떻게 반응하는지를 그림으로 보여주겠습니다:



### 4. variable routing - number page 만들기

1. urls.py 작업 : `path('number/<int:pk>', views.number, name="number")`
2. views.py 작업 : 

```
def number(request, number):  
    context = { 'pk':pk }  
    return render(reuquest, 'articles/number.html', context)
```
3. number.html 작업 : 이 페이지의 번호는 `{{ pk }}` 입니다.

## 5. throw page 만들기

1. `urls.py` 작업 : `path('throw/', views.throw, name="throw")`
2. `views.py` 작업 : 데이터 보내는 작업은 `views` 함수에서 따로 다루는게 없다 그냥 `index` 함수와 동일한 구성으로 만들면 됨 !!
3. `throw.html` 작업 : `form tag` 적용해주기  

```
<form action="받는주소"> -> 주소는 url 태그로 : {% url  
'articles:catch'%}  
    <label for="message"></label>  
    <input type="text" id="message" name="message">  
    <button>보내기</button>  
</form>
```

## 6. catch page 만들기

1. `urls.py` 작업 : `path('catch/', views.catch, name="catch")`
2. `views.py` 작업 : 데이터 받아오는 메서드 `GET` 사용하기  

```
def catch(request):  
    message = request.GET.get('message')  
    context = { 'message':message }  
    return render(request, 'articles/catch.html', context)
```
3. `catch.html` 작업 : `{{'message'}}` 받았어 !!!