

# Ask Django

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

## EP 2. First Django App/URLConf/View/Template

# One Project, Multi App 구조

- 장고 프로젝트 : <장고 프로젝트 Rule>에 따라, 파일/디렉토리 구성이 된 디렉토리
- <장고 앱 Rule>에 따라, 파일/디렉토리 구성이 된 디렉토리
  - 하나의 장고 프로젝트의 다수의 장고 앱

# 장고 프로젝트

manage.py

myproj/\_\_init\_\_.py

myproj/settings.py

myproj/urls.py

myproj/wsgi.py

myproj/templates/layout.html

# 장고 앱

myapp1/\_\_init\_\_.py

myapp1/admin.py

myapp1/forms.py

myapp1/models.py

myapp1/urls.py

myapp1/static/myapp1/logo.png

myapp1/templates/myapp1/

myapp1/templates/myapp1/layout.html

myapp1/templates/myapp1/post\_list.html

myapp1/templates/myapp1/post\_detail.html

# 첫 App/URLConf/ View/Template

# 첫번째 앱 <blog> 앱 생성

- 앱 디렉토리 생성 : shell> python manage.py startapp blog
  - django/conf/app\_template 구성으로 생성 #src
- 앱을 프로젝트에 등록 : 아래와 같이 myproj/settings.py 을 편집하여, INSTALLED\_APPS 항목 끝에 blog 앱이름을 추가

# myproj/settings.py 파일

```
INSTALLED_APPS = [  
    # 생략  
    'blog',  
]
```

## 주의 : 콤마에 유의

다음과 같이 쓰시면 'accountblog' 앱으로 등록을 시도합니다.

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'taggit',  
    'bootstrap3',  
    'accounts'  
    'blog',  
]
```

# 첫 뷰 blog.views.post\_list 구현

```
# blog/views.py
from django.shortcuts import render

def post_list(request):
    return render(request, 'blog/post_list.html')

# blog/urls.py
from django.conf.urls import url
from . import views

urlpatterns = [
    url(r'^$', views.post_list),
]
```

```
# myproj/urls.py
from django.conf.urls import include, url
from django.contrib import admin

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^blog/', include('blog.urls')),
]
```

```
<!-- blog/templates/blog/post_list.html -->
<h1>AskDjango</h1>
<p>안녕하세요. 여러분의 파이썬/장고 페이스메이커가 되겠습니다. ;) </p>
```

- <http://localhost:8000/blog/> 주소로 접속해보세요.
- 템플릿 경로를 `blog/templates/post_list.html` 경로가 아닌 `blog/templates/blog/post_list.html` 경로를 쓰는 지에 대한 이유는, 후에 Template Loader에서 설명합니다.

# 휴대폰망을 통해 접속해보기

# 개발서버 옵션

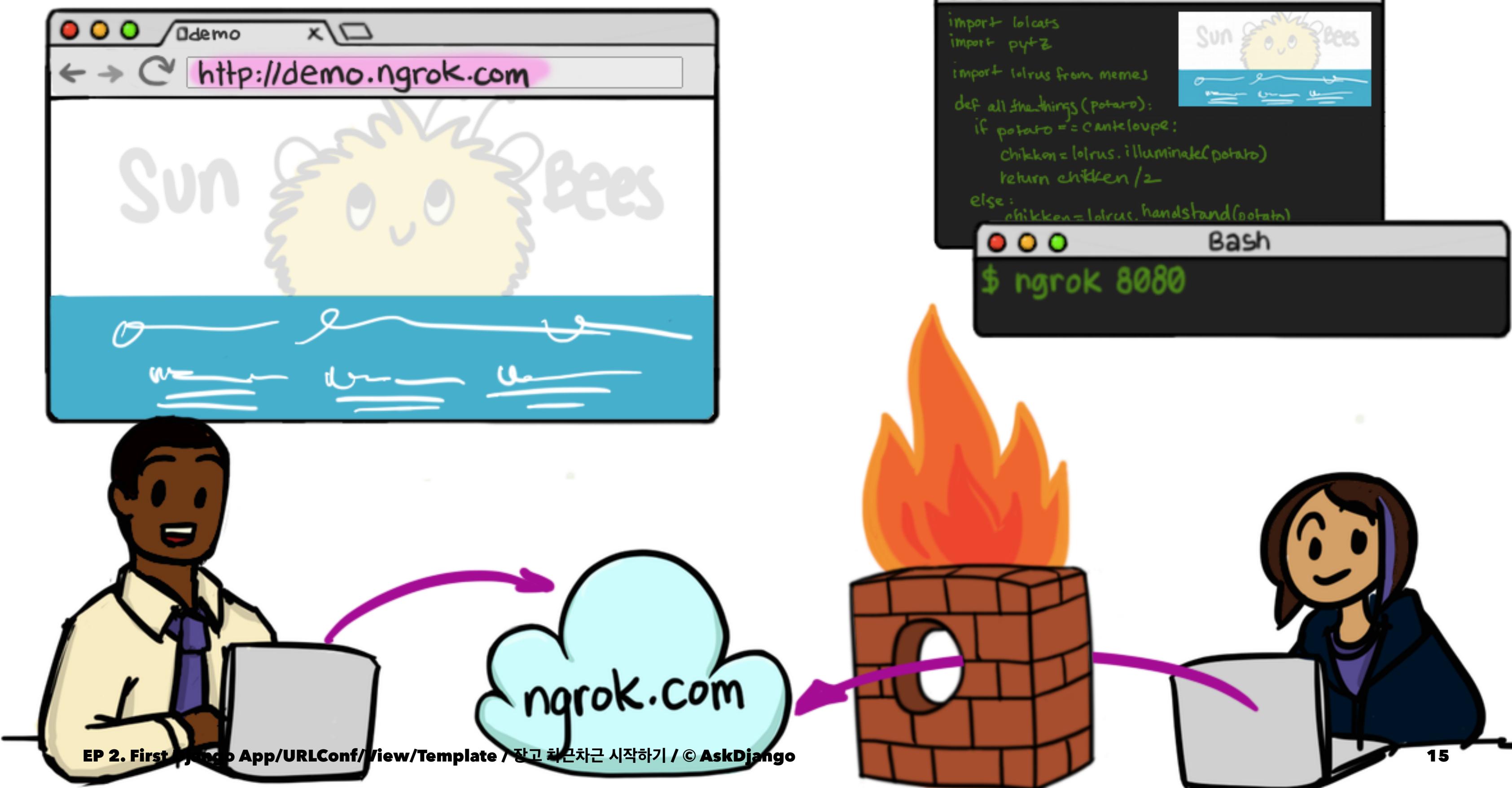
- `python manage.py runserver`
  - 장고서버가 돌고 있는 머신에서만 서버 접속 가능
  - 네트워크를 통해 다른 컴퓨터에서 접속 불가능
- `python manage.py runserver 0:8000`
  - 네트워크를 통해, 같은 네트워크의 다른 컴퓨터에서 접속 가능
  - 외부망에서 접속하려면, 외부 네트워크 설정이 추가로 필요

# 휴대폰 브라우저로 개발서버에 접속할려면

- 방법1) 같은 네트워크를 쓰는 WIFI망에 접속을 하든지
- 방법2) 외부 네트워크 설정
  - 회사에서는 거의 불가능
  - 집에서는 뭐가 뭔지 잘 모르겠고, 귀찮 ...
  - ngrok 를 쓰자.

# ngrok **Secure tunnels to localhost**

- Demo without deploying
- **Simplify mobile device testing**



## 설치 및 실행

- 설치 : [ngrok.com](https://ngrok.com)에서 ngrok를 다운/압축을 풀어, ngrok 실행파일을 myproj 프로젝트 경로로 복사
- 실행
  - 개발서버를 먼저 8000 포트 (디폴트 포트번호)로 구동
  - 윈도우에서 실행> ngrok http 8000
  - 맥에서 실행> ./ngrok http 8000
- 접속 : 우측 스크린샷에서처럼 http 주소로 휴대폰 브라우저로 접속

```
ngrok by @inconshreveable

Tunnel Status          online
Update                 update available (version 2.0.24, Ctrl-U to update)
Version                2.0.19/2.1.1
Web Interface          http://127.0.0.1:4040
Forwarding             http://4b6b19d8.ngrok.io -> localhost:8000
                        https://4b6b19d8.ngrok.io -> localhost:8000

Connections            ttl     opn      rt1     rt5     p50     p90
                        0       0        0.00    0.00    0.00    0.00
```

# 휴대폰으로 접속해보면 ... :-(

- 사이트가 잘 뜨긴 합니다.
- 그렇지만, 글자가 너무 작고
- Zoom 을 하면 되긴 하지만 ...
- 모바일에 전혀 대응되지 않은 페이지같음.

# viewport meta 태그 적용하면, OK

```
<meta name="viewport" content="width=device-width, initial-scale=1.0,  
minimum-scale=1.0, maximum-scale=1.0,  
user-scalable=no" />
```

- 배율 x1.0 고정으로, 유저의 배율 조정을 막겠다는 선언
- 본 태그는 네이버 모바일 페이지에서도 확인 가능

# ngrok 프로그램 주의사항

- ngrok.io 주소는 변경될 수 있습니다. 주소를 고정시키려면, 유료서비스를 써야합니다.
- 실제 서비스 배포가 아닙니다. NEVER !!!
- 이는 사설 네트워크 안의 개발서버를 포트포워딩 등의 설정없이도, 외부 네트워크에서 접속가능도록 해주는 것일 뿐입니다.

A wooden deck chair is positioned on a sandy beach, facing the ocean. The sun is setting, casting a warm orange glow over the scene. A vibrant rainbow arches across the sky above the horizon. In the background, there are some low hills or mountains under a clear sky.

Life is short,  
use Python3/Django.