

Ask Django

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

EP 1. 개발환경 세팅

Django Python Full Stack Web Framework

- <https://wiki.python.org/moin/WebFrameworks>
- **Django** : The Web framework for perfectionists with deadlines.
- Flask : a micro framework for Python based on Werkzeug.
- Pyramid : a small, fast, down-to-earth,
- Bottle : a fast and simple micro framework for small web-applications

Django Python Full Stack Web Framework

- 백엔드를 담당하는 파이썬 풀스택 웹프레임워크
 - 백엔드 개발에 필요한 모든 기능을 제공
- Lawrence Journal-World 신문사에서 2003년부터 개발하여, 2005년에 세상에 공개
- 2008년에 1.0 릴리즈 (Django Roadmap)
- 기타리스트 Django Reinhardt 이름을 따서, Django (쟁고, 장고)

대개의 웹서비스

- 백엔드 개발
- 프론트엔드 개발 : HTML, CSS, JavaScript 언어를 기본으로 다수의 라이브러리/프레임워크
- 백엔드 서버 운영 : 다양한 클라우드
- 앱 개발 (Android, iOS)

파이썬 설치

- 파이썬 3.4 이상
 - 파이썬 2에게 이제 그만 휴식을 주세요.
- 처음부터 pyenv, venv, virtualenv를 구지 쓰지 않으셔도 됩니다.
- 윈도우에서 데이터분석이나 머신러닝을 같이 하실려면, Anaconda Python 64비트 추천.

파이썬 버전 확인

셸> python --version

Python 3.5.2 :: Anaconda custom (x86_64)

윈도우에서는 PowerShell보다는 명령프롬프트가 지원이 좋습니다.

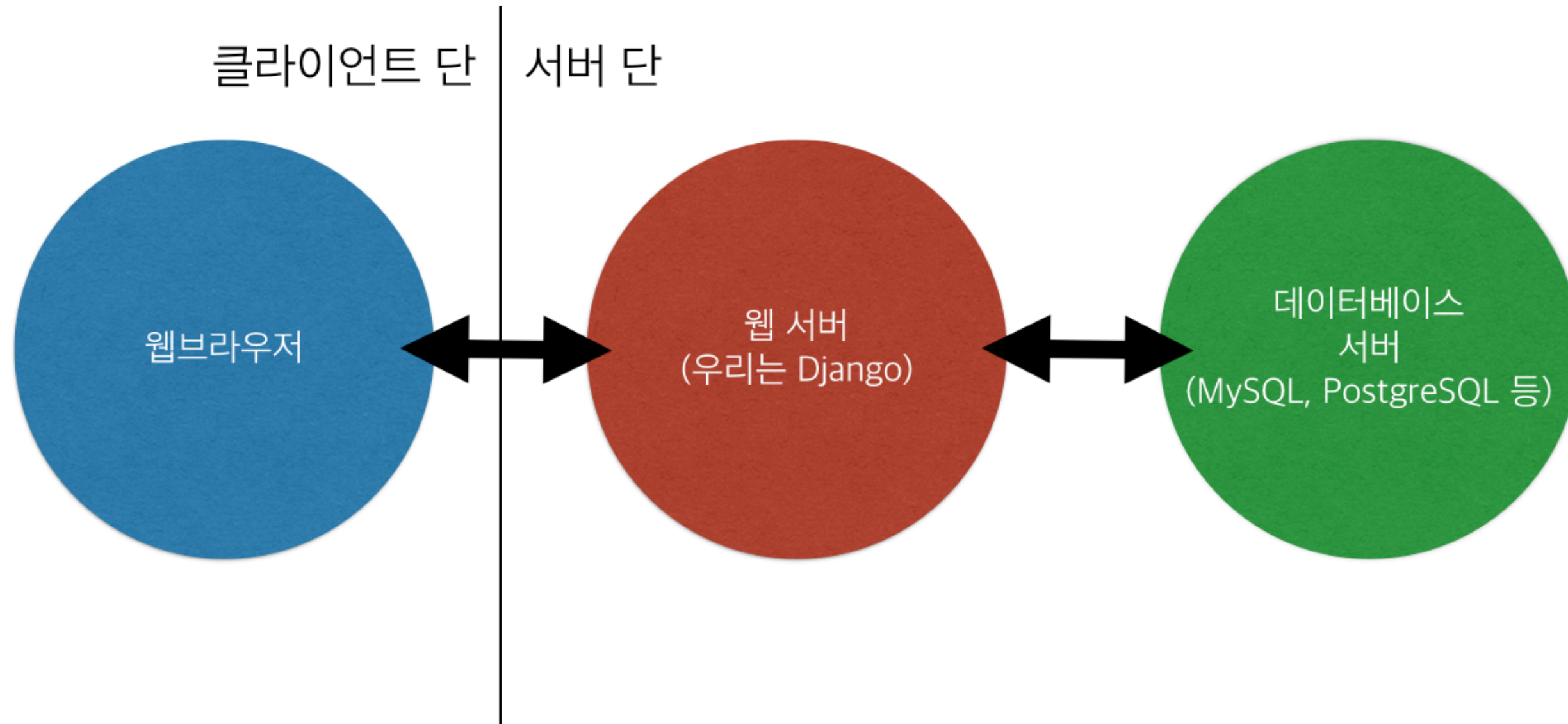
주의 : 맥/리눅스에서는 python 명령으로 파이썬2가 실행될 수 있습니다.
그럴 때에는 python3 명령을 입력해보세요. 환경변수 PATH에 대한 명확한 이해가 필요합니다.

장고 설치

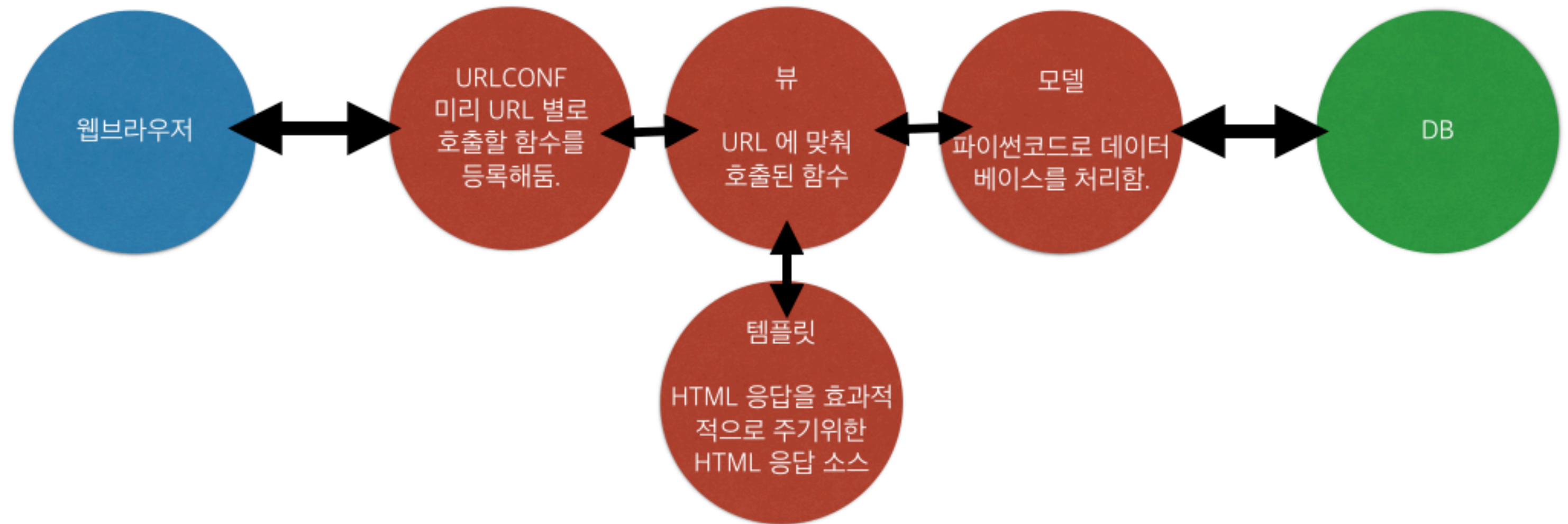
- 장고 1.10 이상
- Django 에서의 LTS (Long Term Support) 버전
 - 대개 3년동안 업데이트를 지원 ([Django Roadmap](#))
- 공식 소스코드 저장소 : <http://github.com/django/django>
- 설치 : 파이썬 패키지 매니저인 pip를 이용
 - 쉘> pip install "django>=1.10"

웹 애플리케이션 및 장고 큰 그림

웹 애플리케이션 기본 구조



장고 기본 구조



서버를 http://localhost:8000/ 에서 서비스할 경우

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

urlpatterns = [
    # http://localhost:8000/blog/ 주소로 접근하면, views.post_list 함수를 호출
    url(r'^blog/$', views.post_list),

    # http://localhost:8000/blog/new/ 주소로 접근하면, views.post_new 함수를 호출
    url(r'^blog/new/$', views.post_new),

    # http://localhost:8000/blog/10/ 주소로 접근하면, views.post_detail 함수를 호출
    url(r'^blog/10/$', views.post_detail),

    # http://localhost:8000/blog/100/ 주소로 접근하면, views.post_detail 함수를 호출
    url(r'^blog/100/$', views.post_detail),

    # http://localhost:8000/blog/숫자/ 주소로 접근하면, views.post_detail 함수를 호출
    url(r'^blog/(?P<id>\d+)/$', views.post_detail),

    # http://localhost:8000/blog/숫자/edit/ 주소로 접근하면, views.post_edit 함수를 호출
    url(r'^blog/(?P<id>\d+)/edit/$', views.post_edit),

    # http://localhost:8000/blog/숫자/edit/ 주소로 접근하면, views.post_delete 함수를 호출
    url(r'^blog/(?P<id>\d+)/edit/$', views.post_delete),
]
```

blog/views.py 샘플

```
from django.shortcuts import get_object_or_404, render
from blog.models import Post

def post_list(request):
    qs = Post.objects.all() # DB로부터 Post목록을 Fetch할 예정
    return render(request, 'blog/post_list.html', {'post_list': post_list})

def post_detail(request, id):
    post = get_object_or_404(Post, id=id) # DB로부터 지정Post를 Fetch
    return render(request, 'blog/post_detail.html', {'post': post})

def post_edit(request, id):
    # TODO: 구현 예정
    pass

def post_delete(request, id):
    # TODO: 구현 예정
    pass
```

```
# blog/models.py
from django.db import models

class Post(models.Model):
    title = models.CharField(max_length=100)
    content = models.TextField()
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
```

```
<!-- blog/templates/blog/post_list.html -->
<table>
  <tbody>
    {% for post in post_list %}
      <tr>
        <td>{{ post.id }}</td>
        <td>
          <a href="{% url 'blog:post_detail' post.id %}">
            {{ post.title }}
          </a>
        </td>
        <td>{{ post.updated_at }}</td>
      </tr>
    {% endfor %}
  </tbody>
</table>
```

```
<!-- blog/templates/blog/post_detail.html -->
<h1>{{ post.title }}</h1>
<p>{{ post.updated_at }}</p>
<hr/>
{{ post.content|linebreaks }}
<hr/>
<a href="{% url 'blog:post_list' %}">글 목록</a>
```

장고 프로젝트 생성

window> mkdir c:\dev; cd c:\dev # drive root에 가까운 경로에 개발디렉토리

mac/linux> mkdir ~/dev; cd ~/dev

dev> django-admin startproject askdjango # Django 프로젝트 생성

- django/conf/project_template 구성으로 생성 #src

장고 프로젝트 생성 및 개발 서버 구동

```
dev> cd askdjango # 디렉토리 이동 (cd : change directory)
dev/askdjango> python manage.py migrate # Model 내역을 데이터베이스에 반영
dev/askdjango> python manage.py createsuperuser # Superuser 계정 생성
dev/askdjango> python manage.py runserver # 개발서버 구동
```

- 위 **python** 명령은 각 머신의 파이썬3 인터프리터. 머신에 따라 **python**, **python3** 혹은 **python3.5** 등이 될 수도 있습니다.
- 이제 웹 브라우저를 띄워, <http://localhost:8000/admin/> 으로 접속 해봅시다.

여기에서 잠깐 !!!

- 웹서비스 개발 시의 웹브라우저는 가급적 Chrome이나 Firefox를 쓰는 것이 좋습니다. 웹브라우저 개발자도구 지원이 좋습니다.



*Life is short,
use Python3/Django.*