

# 장고 차근차근 시작하기 2/E

당신의 파이썬/장고 페이스메이커가 되겠습니다. ☺

EP-02. 장고 주요 구성요소

# 장고 주요 기능들 (1)

본 코스에서 다룰 내용들

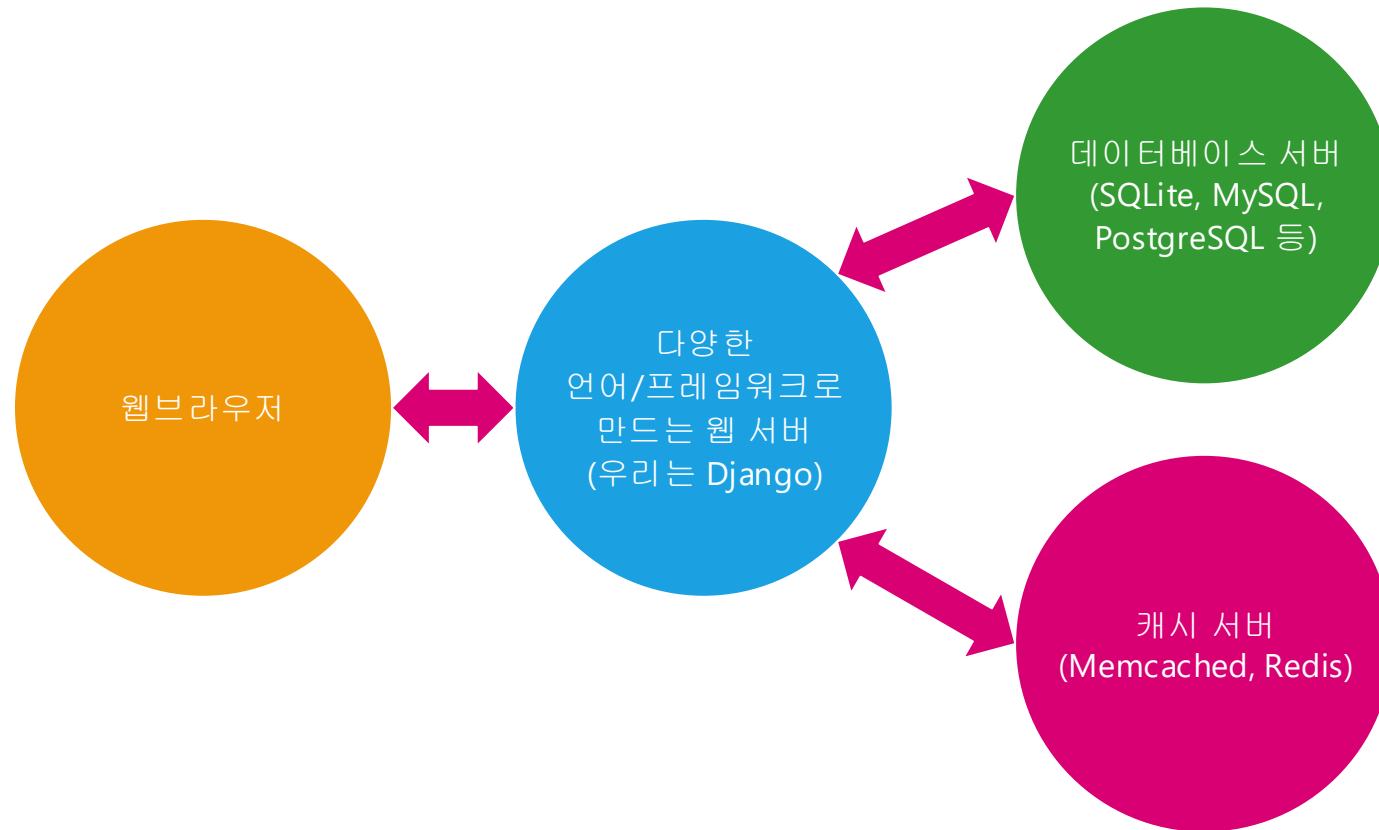
1. Function Based Views : 함수로 HTTP 요청 처리
2. Models : 데이터베이스와의 인터페이스
3. Templates : 복잡한 문자열 조합을 보다 용이하게. 주로 HTML 문자열 조합 목적으로 사용하지만, 푸쉬 메시지나 이메일 내용을 만들 때에도 쓰면 편리.
4. Admin 기초 : 심플한 데이터베이스 레코드 관리 UI
5. Logging : 다양한 경로로 메시지 로깅
6. Static files : 개발 목적으로의 정적인 파일 관리
7. Messages framework : 유저에게 1회성 메시지 노출 목적

# 장고 주요 기능들 (2)

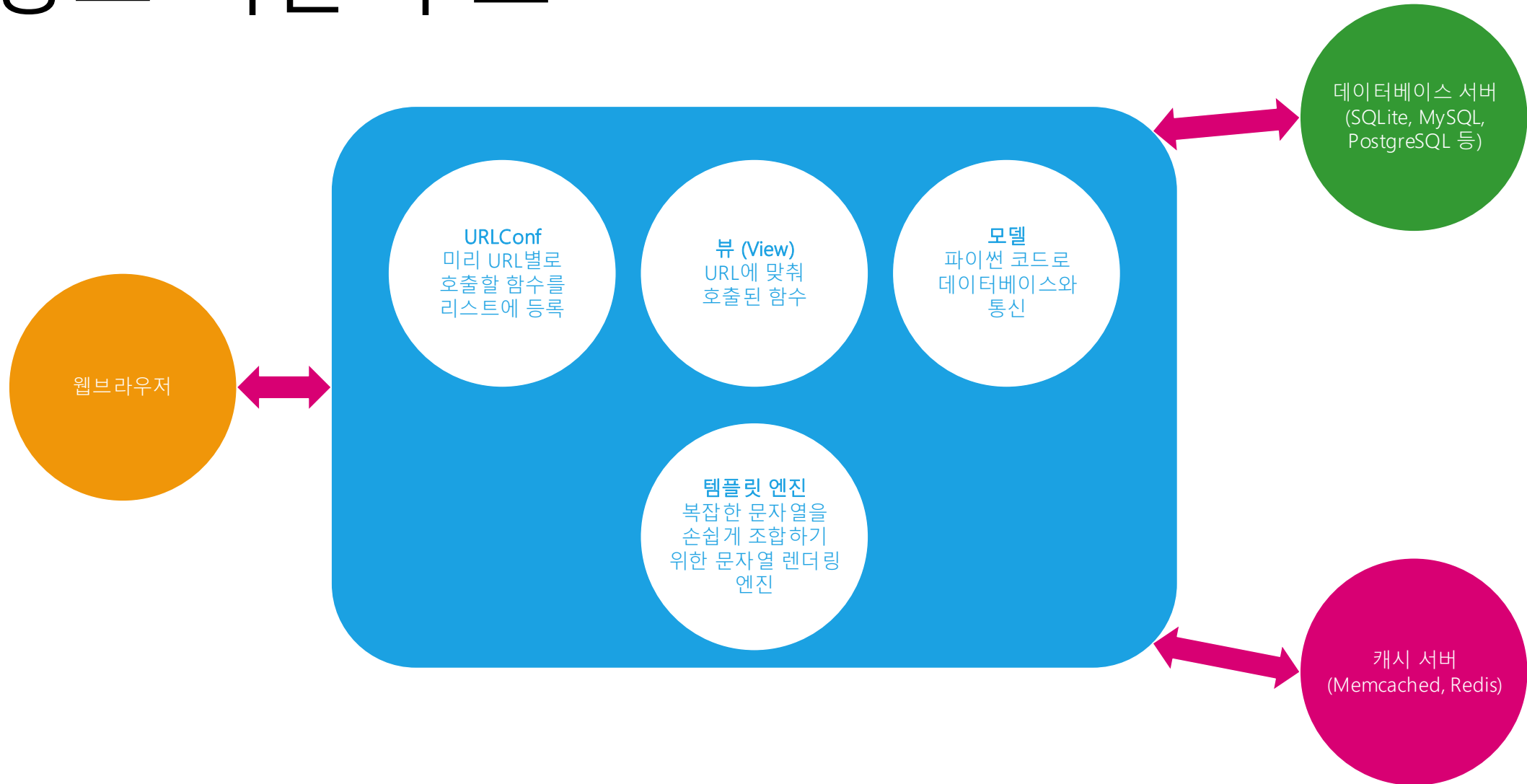
별도 코스에서 다룰 내용들

1. Class Based Views : 클래스로 함수 기반 뷰 만들기
2. Forms : 입력폼 생성, 입력값 유효성 검사 및 DB로의 저장
  - Validators & Fields & Widgets
3. 테스트
4. 국제화 & 지역화
5. 캐싱
6. Geographic : DB의 Geo 기능 활용 (PostgreSQL 중심)
7. Sending Emails
8. Syndication Feeds (RSS/Atom)
9. Sitemaps

# 웹 애플리케이션 기본 구조



# 장고 기본 구조



# 하나의 URL처리를 위한 URLConf, View, Template 그리고 Model

맛보기 코드

# 모델 예시

shop/models.py

```
from django.db import models
```

```
class Item(models.Model):  
    name = models.CharField(max_length=100)  
    desc = models.TextField(blank=True)  
    price = models.PositiveIntegerField()  
    created_at = models.DateTimeField(auto_now_add=True)  
    updated_at = models.DateTimeField(auto_now=True)
```

# URLConf 예시

프로젝트/urls.py

```
from django.urls import path
from shop import views

app_name = 'shop'

urlpatterns = [
    path('shop/', views.item_list),
]
```



# 뷰 예시

장고앱1/views.py

```
from django.shortcuts import render
from shop.models import Item

def item_list(request):
    qs = Item.objects.all()    # DB로부터 Item목록을 Fetch할 예정
    return render(request, 'shop/item_list.html', {
        'item_list': qs,
    })
```

# 템플릿 예시

shop/templates/shop/item\_list.html

```
<table>
  <tbody>
    {% for item in item_list %}
      <tr>
        <td>{{ item.id }}</td>
        <td>
          {{ item.name }}<br/>
          {{ item.price }}<br/>
          {{ item.desc }}
        </td>
        <td>{{ item.updated_at }}</td>
      </tr>
    {% endfor %}
  </tbody>
</table>
```

인생은 짧습니다.  
파이썬/장고를 쓰세요.