



리액트와 함께 장고 시작하기 / 장고

# 장고 admin을 통한 데이터 관리 (기초)

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

# django admin

## django.contrib.admin 앱을 통해 제공

디폴트 경로 : /admin/ ➔ 실제 서비스에서는 다른 주소로 변경 권장

혹은 [django-admin-honeypot](#) 앱을 통해, 가짜 admin 페이지 노출

## 모델 클래스 등록을 통해, 조회/추가/수정/삭제 웹UI를 제공

서비스 초기에, 관리도구로서 사용하기에 제격

관리도구 만들 시간을 줄이고, End-User 서비스에 집중 !

## 내부적으로 Django Form을 적극적으로 사용

<https://docs.djangoproject.com/en/3.0/ref/contrib/admin/>

# 모델 클래스를 admin에 등록하기

앱/admin.py

```
from django.contrib import admin
from .models import Item
```

# 등록법 #1

```
admin.site.register(Item)
```

# 기본 ModelAdmin으로 동작

---

# 등록법 #2

```
class ItemAdmin(admin.ModelAdmin):
    pass
```

```
admin.site.register(Item, ItemAdmin) # 지정한 ModelAdmin으로 동작
```

---

# 등록법 #3

```
@admin.register(Item)
class ItemAdmin(admin.ModelAdmin):
    pass
```

# 모델 클래스에 `__str__` 구현

admin 모델 리스트에서 "모델명 object"를 원하는 대로 변경하기 위해

객체를 출력할 때, 객체.`__str__`()의 리턴값을 활용

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

```
class Item(models.Model):  
    name = models.CharField(max_length=100)  
    desc = models.TextField(blank=True)  
    price = models.PositiveIntegerField()  
    is_publish = models.BooleanField(default=False)
```

```
def __str__(self):  
    return f'<{self.pk}> {self.name}'
```

`str(...)`은 자바의 `toString()`과 유사

# list\_display 속성 정의

모델 리스트에 출력할 컬럼 지정

```
from django.contrib import admin
from .models import Item
```

```
@admin.register(Item)
```

```
class ItemAdmin(admin.ModelAdmin):
```

```
    list_display = ['pk', 'name', 'short_desc', 'price', 'is_publish']
```

```
    def short_desc(self, item):
        return item.desc[:20]
```

# list\_display\_links 속성 정의

list\_display 지정된 이름 중에, detail 링크를 걸 속성 리스트

```
from django.contrib import admin
from .models import Item
```

```
@admin.register(Item)
```

```
class ItemAdmin(admin.ModelAdmin):
```

```
    list_display = ['pk', 'name', 'short_desc', 'price', 'is_publish']
```

```
    list_display_links = ['name']
```

```
    def short_desc(self, item):
```

```
        return item.desc[:20]
```

# search\_fields 속성 정의

admin내 검색UI를 통해, DB를 통한 where 쿼리 대상 필드 리스트

```
from django.contrib import admin
from .models import Item
```

```
@admin.register(Item)
class ItemAdmin(admin.ModelAdmin):
    list_display = ['pk', 'name', 'short_desc', 'price', 'is_publish']
    list_display_links = ['name']
    search_fields = ['name']

    def short_desc(self, item):
        return item.desc[:20]
```

# list\_filter 속성 정의

지정 필드값으로 필터링 옵션 제공

```
from django.contrib import admin
from .models import Item
```

```
@admin.register(Item)
```

```
class ItemAdmin(admin.ModelAdmin):
```

```
    list_display = ['pk', 'name', 'short_desc', 'price', 'is_publish']
```

```
    list_display_links = ['name']
```

```
    list_filter = ['is_publish']
```

```
    search_fields = ['name']
```

```
    def short_desc(self, item):
```

```
        return item.desc[:20]
```



Life is short.  
You need Python and Django.

I will be your pacemaker.

