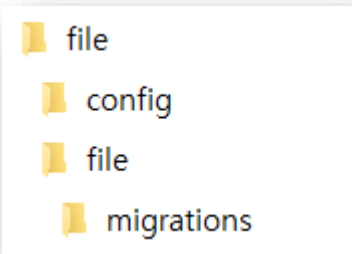
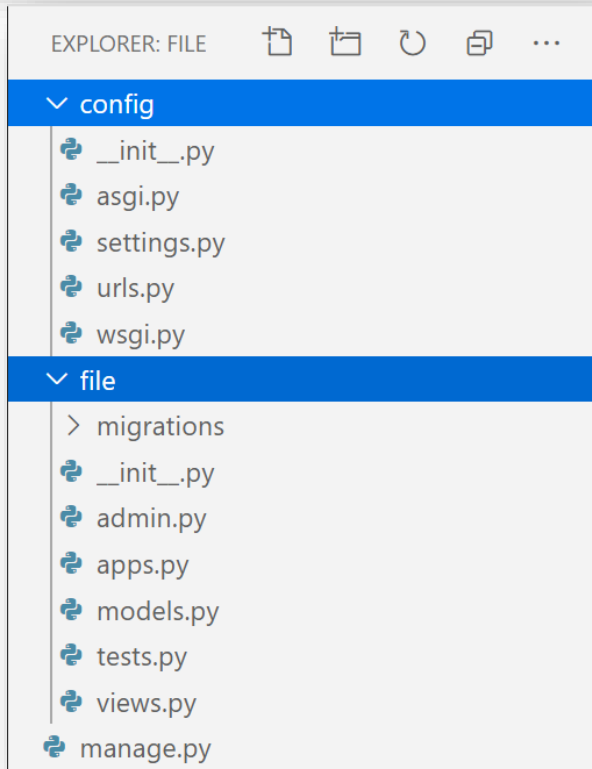


## ■ 파일 업로드

### ● 프로젝트 / 앱 생성

```
E:\dev\python_django>mkdir file  
E:\dev\python_django>cd file  
E:\dev\python_django\file>django-admin startproject config .  
E:\dev\python_django\file>django-admin startapp file  
E:\dev\python_django\file>_
```



## ■ 파일 업로드

### ● 업로드된 파일 정보 확인

```
# my_file 파라미터가 없는 경우 오류 발생  
upload_file = request.FILES['my_file']
```

```
# my_file 파라미터가 없는 경우 None 반환  
upload_file = request.FILES.get('my_file')
```

```
# 파일명  
name = upload_file.name
```

```
# 파일크기  
size = upload_file.size
```

```
['DEFAULT_CHUNK_SIZE', '__bool__', '__class__', '__delattr__', '__dict__', '__dir__  
' , '__doc__', '__enter__', '__eq__', '__exit__', '__format__', '__ge__', '__getattr  
ibute__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le_  
' , '__len__', '__lt__', '__module__', '__ne__', '__new__', '__reduce__', '__reduce  
_ex__', '__repr__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', '__  
weakref__', '_get_name', '_name', '_set_name', 'charset', 'chunks', 'close', 'close  
d', 'content_type', 'content_type_extra', 'encoding', 'field_name', 'file', 'fileno  
' , 'flush', 'isatty', 'multiple_chunks', 'name', 'newlines', 'open', 'read', 'reada  
ble', 'readinto', 'readline', 'readlines', 'seek', 'seekable', 'size', 'tell', 'tru  
ncate', 'writable', 'write', 'writelines']
```

## ■ 파일 업로드

- 업로드된 파일 저장 (임시 저장 -> 영구 저장)

```
with open(name, 'wb') as file:
    # read()
    # 메모리를 많이 사용하고 저장시간이 오래 걸림
    file.write(upload_file.read())

    # chunks()
    # 적은 메모리 사용, 저장시간 짧음
    # 기본 chunk 크기 - 65536Byte (64KB)
    for chunk in upload_file.chunks():
        file.write(chunk)
```

## ■ 파일 업로드

### ● config/urls.py

```
from django.contrib import admin
from django.urls import path
from file import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('upload1/', views.upload1),
]
```

## ■ 파일 업로드

### ● file/views.py

```
from django.shortcuts import render
from django.http import HttpResponse

def upload1(request):
    if request.method == 'POST':
        upload_file = request.FILES.get('my_file')

        name = upload_file.name
        size = upload_file.size

        with open(name, 'wb') as file:
            for chunk in upload_file.chunks():
                file.write(chunk)

        return HttpResponse('%s<br>%s' % (name, size))

    return render(request, 'upload1.html', {})
```

## ■ 파일 업로드

### ● templates/upload1.html

```
<form action="/upload1/" method="post"
      enctype="multipart/form-data">
  <input type="file" name="my_file">
  <input type="submit" value="완료">
  {% csrf_token %}
</form>
```

### ● config/settings.py

```
TEMPLATES = [
    {
        ...

        'DIRS': [os.path.join(BASE_DIR, 'templates')],
        'APP_DIRS': True,

        ...
    },
]
```

## ■ 파일 업로드

- <http://localhost:8000/upload1/>

파일 선택

선택된 파일 없음

완료

← → ↻ ⓘ localhost:8000/upload1/

test.bin  
1000000

- 테스트용 파일 생성

```
E:W>fsutil file createnew test.bin 1000000  
E:Wtest.bin 파일 작성
```

## ■ 파일 업로드 - multiple

### ● config/urls.py

```
from django.contrib import admin
from django.urls import path
from file import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('upload1/', views.upload1),
    path('upload2/', views.upload2),
]
```



## ■ 파일 업로드 - multiple

### ● file/views.py

```
def upload2(request):
    if request.method == 'POST':
        upload_files = request.FILES.getlist('my_file')

        result = ''
        for upload_file in upload_files:
            name = upload_file.name
            size = upload_file.size

            with open(name, 'wb') as file:
                for chunk in upload_file.chunks():
                    file.write(chunk)
            result += '%s<br>%s<br>' % (name, size)

        return HttpResponse(result)

    return render(request, 'upload2.html', {})
```

## ■ 파일 업로드 - multiple

### ● templates/upload2.html

```
<form action="/upload2/" method="post"
      enctype="multipart/form-data">
  <input type="file" name="my_file" multiple>
  <input type="submit" value="완료">
  {% csrf_token %}
</form>
```

## ■ 파일 업로드 - multiple

● <http://localhost:8000/upload2/>

파일 선택

선택된 파일 없음

완료

← → ↻ ⓘ localhost:8000/upload2/

test.bin  
1000000  
test2.bin  
2000000  
test3.bin  
3000000

## ■ 파일 다운로드

### ● file/views.py

```
def download(request):  
    filepath = os.path.join(settings.BASE_DIR, '경로/파일명')  
    filename = os.path.basename(filepath)  
    with open(filepath, 'rb') as f:  
        response = HttpResponse(  
            f, content_type='application/octet-stream')  
        response['Content-Disposition'] =  
            'attachment; filename="{0}"'.format(filename)  
    return response
```