


플라스크 - 파일 리스트 정보

Python

 복사

```
from flask import Flask, render_template, request, send_file import os
from datetime import datetime import zipfile app = Flask(__name__)
@app.route('/', methods=['GET', 'POST']) def list(): UPLOAD_PATH =
"uploads" files = [] for file in os.listdir(UPLOAD_PATH): file_path =
os.path.join(UPLOAD_PATH, file) file_size = os.path.getsize(file_path)
file_ctime =
datetime.fromtimestamp(os.path.getctime(file_path)).strftime('%Y-%m-%d
%H:%M:%S') print(f"경로: {file_path}, 크기: {file_size}, ctime:
{file_ctime}") files.append((file, file_size, file_ctime, file_path))
print(files) #디버깅 return render_template('list.html', files=files)
@app.route('/compress', methods=['GET', 'POST']) def compress():
UPLOAD_PATH = "uploads" files = request.form.getlist("files") zip_path =
os.path.join(UPLOAD_PATH, "compress.zip") with zipfile.ZipFile(zip_path,
"w") as zip_file: for file in files: file_path = os.path.join(UPLOAD_PATH,
file) zip_file.write(file_path, file) return send_file(zip_path,
as_attachment=True) if __name__ == "__main__": app.run(debug=True)
```

list.html

```
<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta
name="viewport" content="width=device-width, initial-scale=1.0"> <link
rel="stylesheet" type="text/css" href="{{url_for('static',
filename='style.css')}}"> <title>Document</title> </head> <body> <form
method="POST" action="{{url_for('compress')}}"> <table border="1"> <tr>
<td>파일 이름</td> <td>파일 크기</td> <td>파일 날짜</td> <td>파일 경로</td>
<td>파일 선택</td> </tr> {% for file in files %} <tr> <td>{{file[0]}}</td>
<td>{{file[1]}}</td> <td>{{file[2]}}</td> <td>{{file[3]}}</td> <td><input
type="checkbox" name="files" value="{{file[0]}}"></td> </tr> {% endfor %}
</table> <button type="submit">파일 압축하기</button> </form> </body>
</html>
```

style.css

```
/* CSS 스타일을 여기에 추가하세요 */ body { font-family: Arial, sans-serif;
} h1 { color: #333; } table { border-collapse: collapse; width: 100%; }
th, td { border: 1px solid #ddd; padding: 8px; text-align: left; } th {
background-color: #f2f2f2; } tr:nth-child(even) { background-color:
#f9f9f9; } tr:hover { background-color: #f5f5f5; }
```

추가 기능 list.html

```
<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta
name="viewport" content="width=device-width, initial-scale=1.0"> <link
rel="stylesheet" type="text/css" href="{{url_for('static',
filename='style.css')}}"> <title>Document</title> </head> <body> <form
method="POST" action="{{url_for('upload')}}" enctype="multipart/form-
data"> <input type="file" name="file"> <button type="submit">업로드
</button> </form> <form method="POST" action="{{url_for('compress')}}">
<table border="1"> <tr> <td>파일 이름</td> <td>파일 크기</td> <td>파일 날짜
</td> <td>파일 경로</td> <td>파일 선택</td> </tr> {% for file in files %}
<tr> <td>{{file[0]}}</td> <td>{{file[1]}}</td> <td>{{file[2]}}</td> <td>
{{file[3]}}</td> <td><input type="checkbox" name="files" value="
{{file[0]}}"></td> </tr> {% endfor %} </table> <button type="submit">파일
압축하기</button> </form> </body> </html>
```

app.py

```

from flask import Flask, render_template, request, send_file import os
from datetime import datetime import zipfile app = Flask(__name__)
@app.route('/', methods=['GET', 'POST']) def list(): UPLOAD_PATH =
"uploads" files = [] for file in os.listdir(UPLOAD_PATH): file_path =
os.path.join(UPLOAD_PATH, file) file_size = os.path.getsize(file_path)
file_ctime =
datetime.fromtimestamp(os.path.getctime(file_path)).strftime('%Y-%m-%d
%H:%M:%S') print(f"경로: {file_path}, 크기: {file_size}, ctime:
{file_ctime}") files.append((file, file_size, file_ctime, file_path))
print(files) #디버깅 return render_template('list.html', files=files)
@app.route('/compress', methods=['GET', 'POST']) def compress():
UPLOAD_PATH = "uploads" files = request.form.getlist("files") zip_path =
os.path.join(UPLOAD_PATH, "compress.zip") with zipfile.ZipFile(zip_path,
"w") as zip_file: for file in files: file_path = os.path.join(UPLOAD_PATH,
file) zip_file.write(file_path, file) return send_file(zip_path,
as_attachment=True) @app.route('/upload', methods=['GET', 'POST']) def
upload(): UPLOAD_PATH = "uploads" file = request.files['file'] file_path =
os.path.join(UPLOAD_PATH, file.filename) file.save(file_path) return "업로
드 완료" if __name__ == "__main__": app.run(debug=True)

```