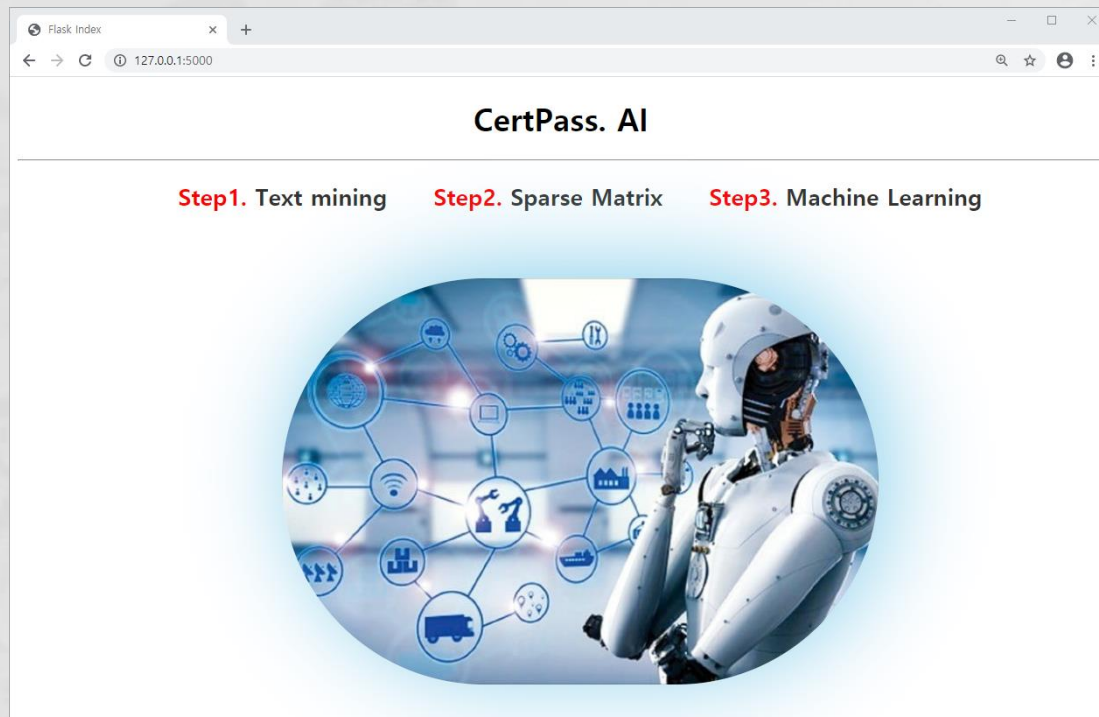


Flask Web

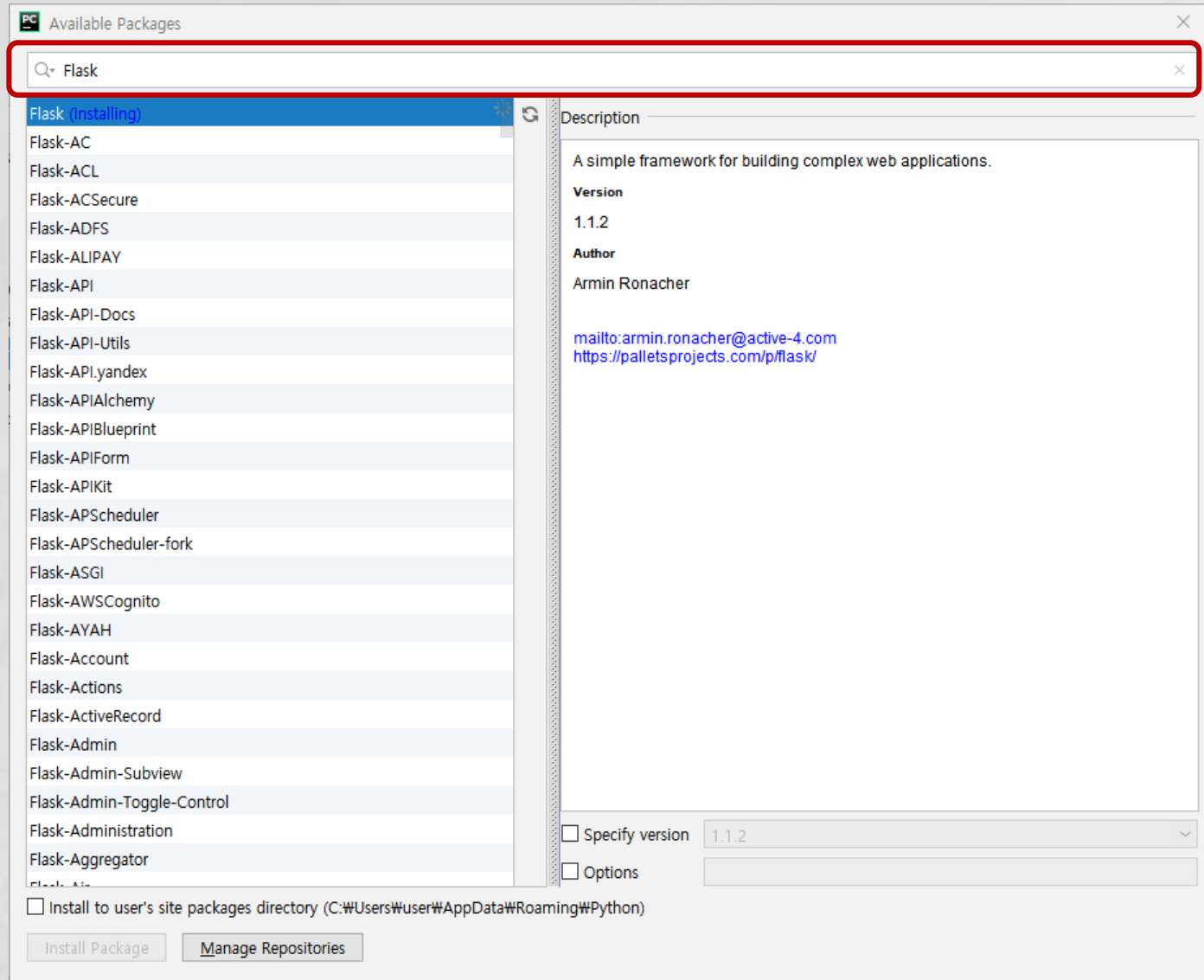
작성자 : 김진성

1. Flask 개요

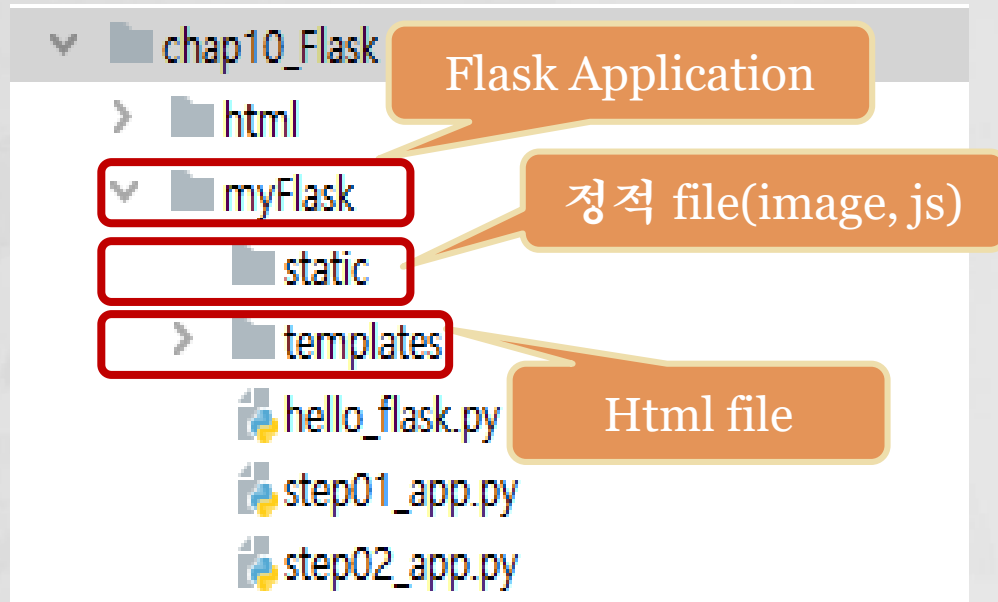
- Python 웹 Application 개발 프레임워크
- 마이크로 프레임워크
 - 핵심 기능으로 구현, 확장 가능한 프레임워크



Flask 설치



폴더 만들기



2. Flask 어플리케이션 만들기


app.py : myFlask 폴더에서 작성

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def index():
    return 'Hello Flask'

@app.route('/info')
def info():
    return 'Info'
```

Flask Application 실행



The screenshot shows the PyCharm IDE with a Flask application named 'hello_flask.py' open. The code in the editor is as follows:

```
4 from flask import Flask
5 app = Flask(__name__)
6
7
8 @app.route('/')
9 def index():
10     return 'Hello Flask'
11
12
13
14
15 @app.route('/info')
16 def info():
17     return 'Info'
18
```

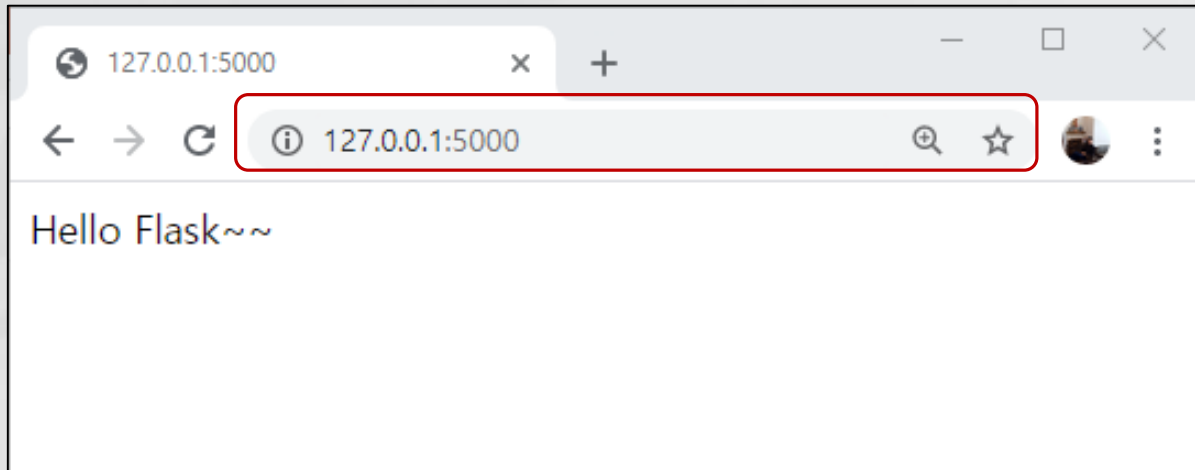
The Run console shows the output of the command:

```
C:\Python37\python.exe D:/ITWILL/3_Python-I/workspace/chap10_Flask/myFlask/hello_flask.py
* Serving Flask app "hello_flask" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

An orange callout bubble with the Korean word '클릭' (Click) points to the 'Run' button in the Run console. A red box highlights the output of the command.

Windows 정품 인증
[설정]으로 이동하여 Windows를 정품 인증합니다.

브라우저에서 확인



3. 템플릿 추가하기

templates 파일 : 사용자 요청과 서버의 응답을 작성하는 html 파일

app.py : 템플릿 코드 추가

```
from flask import Flask, render_template
app = Flask(__name__)
```

```
@app.route('/')
def index():
    return render_template('index.html')
```

```
@app.route('/info')
def info():
    return render_template('info.html')
```


3. 템플릿 추가하기

index.html: 템플릿 파일

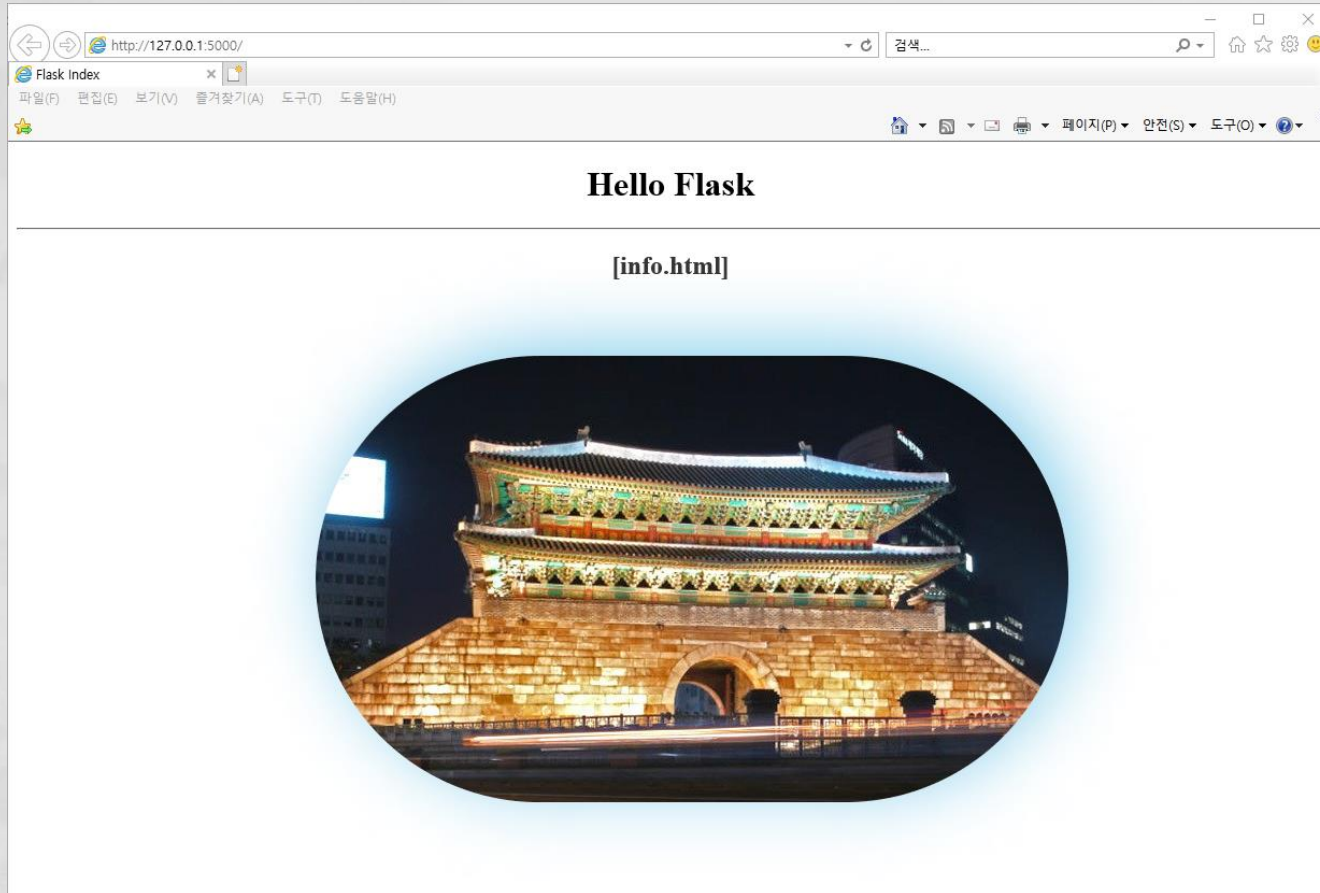
```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Flask Index</title>
</head>
<body>

<h1>Hello Flask</h1>
<p>This page is for Flask tutorial.</p>

</body>
</html>
```

4. Static

static 파일 : 정적 파일(이미지, js, css 등)



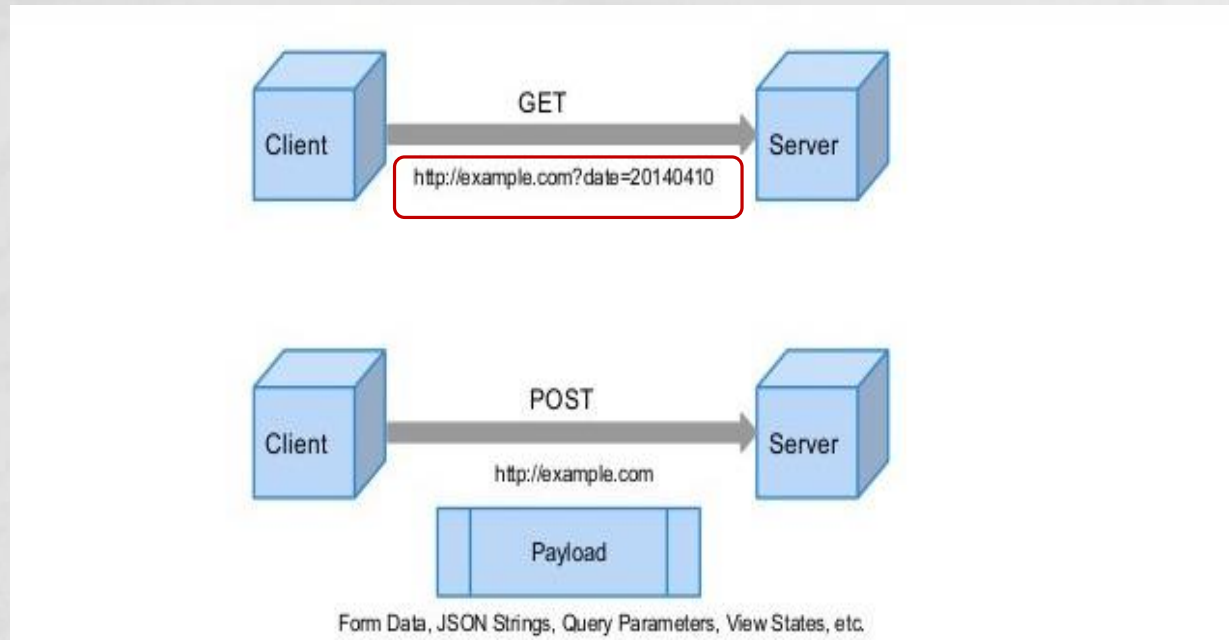
index.html: 이미지 파일 추가

```
<style type="text/css">
  img{
    margin: 50px 0 0 40px;
    border-radius : 500px;
    box-shadow : 10px -10px 100px skyblue;
  }
</style>

<body>
  <div align="center">
    <h1>Hello Flask</h1>
    <hr/>
    <h2> <a href='/info'> [info.html] </a> </h2>
    
  </div>
</body>
```

5. GET/POST

- 템플릿(template)에서 전송하는 파라미터 받는 방식
 - GET : 파라미터가 url에 노출되어 전송(눈에 보임)
 - POST : 파라미터가 HTTP body에 포함되어 전송(눈에 보이지 않음)



Post 방식으로 아이디와 비밀번호 서버 전송

```
<form name="login" method="post" action= "/login">
<table border="1">
  <tr>
    <th> 아이디 </th>
    <td> <input type="text" name="id" size="15"/> </td>
    <!-- type="text" : 키보드로 데이터 입력 상자 -->
  </tr>
  <tr>
    <th> 비밀번호 </th>
    <td> <input type="password" name="pwd" size="15"/> </td>
    <!-- type="password" : 암호 입력 상자 -->
  </tr>
  <tr>
    <th colspan="2">
      <input type="submit" value="전송"/> &nbsp; &nbsp;
      <input type="reset" value="취소"/>
    </th>
  </tr>
</table>
</form>
```

app.py : methods 추가

```
from flask import Flask, render_template
app = Flask(__name__)
```

```
@app.route('/')
def index():
    return render_template('index.html')
```

```
@app.route('/login', methods=['GET', 'POST'])
def login() :
    if request.method=='GET' : # get 방식
        id = request.args.get('id')
        pwd = request.args.get('pwd')

    if request.method=='POST' : # post 방식
        id = request.form['id']
        pwd = request.form['pwd']
```