

# 실전 개발형 코딩 테스트 문제 풀이 ③

## 서버 환경 구축하기

서버 환경 구축하기 | 문제 풀이를 위해 서버 환경 구축하기

강사 나동빈

# 실전 개발형

## 코딩 테스트 문제 풀이 ③

서버 환경 구축하기

개발형 코딩 테스트  
서버 환경 구축하기

## 자신의 컴퓨터에 파이썬 설치하기

개발형  
코딩 테스트  
서버 환경  
구축하기

- 문제 풀이를 시작하기 위해 파이썬을 설치한다.
- 파이썬(Python) 다운로드 경로: <https://www.python.org/downloads/>
- [Downloads] - [All releases] 페이지로 이동하여 구체적인 버전의 파이썬(Python)을 설치할 수 있다.
- 설치 이후에 정상적으로 설치가 완료되었는지 확인하기 위해 명령 프롬프트(CMD)에서 python 명령어를 사용할 수 있다.

```
python example.py
```

개발형 코딩 테스트  
서버 환경 구축하기

## 문제 풀이 서버 실행하기

개발형  
코딩 테스트  
서버 환경  
구축하기

- 문제 풀이를 위해 기본적인 라이브러리를 설치한다.

```
pip install flask  
pip install uuid  
pip install requests  
pip install opencv-python  
pip install numpy
```

- 다음의 명령어를 이용해 문제 풀이 서버를 구동시킨다. (기본 5000번 포트)
- 5000번 포트가 이미 사용 중이라면, 403 오류가 발생할 수 있다.

```
python server.py
```

개발형 코딩 테스트  
서버 환경 구축하기

## Python을 활용한 REST API 호출 방법

개발형  
코딩 테스트  
서버 환경  
구축하기

- 일반적인 개발형 코딩 테스트에서는 REST API를 활용할 것을 요구한다.
- JSON 형식(format)의 데이터를 다룰 수 있어야 한다.
- Python의 경우 REST API 호출 및 JSON 데이터 처리를 손쉽게 할 수 있다.

개발형 코딩 테스트  
서버 환경 구축하기

### REST API 호출 예제: `client_template.py`

개발형  
코딩 테스트  
서버 환경  
구축하기

- Python에서는 requests 라이브러리를 이용해 HTTP 통신을 진행할 수 있다.
- POST 방식으로 서버(server)에 데이터를 전송할 때는 JSON 형식으로 데이터를 보낼 수 있다.
- 헤더(header)에서 Content-Type 속성의 값으로 application/json를 사용한다.

개발형 코딩 테스트  
서버 환경 구축하기

## REST API 호출 예제: client\_template.py

개발형  
코딩 테스트  
서버 환경  
구축하기

```
import requests, json

API_HOST = '[API_SERVER]'
headers = {
    'Access-Token': '[YOUR_ACCESS_TOKEN]',
    'Content-Type': 'application/json'
}

def request(path, method, data={}):
    url = API_HOST + path
    print(f'Request URL: {url}')
    print(f'HTTP Method: {method}')
    print(f'Headers: {headers}')

    if method == 'GET':
        return requests.get(url, headers=headers)
    elif method == 'POST':
        print(f'Sended data: {data}')
        return requests.post(url, headers=headers, data=json.dumps(data))

response = request('/start', 'GET')
print(f'Response status: {response.status_code}')
print(f'Response headers: {response.headers}')
print(f'Response body: {response.text}')
```