

Flask


▼ 연예인 다편 찾기

Clova Face Recognition API 가이드 - CLOVA A.I. APIs Document

Clova Face Recognition API 가이드 CFR API란? Clova Face Recognition API(이하 CFR API)는 이미지 데이터를 입력받은 후 얼굴 인식 결과를 JSON 형태로 반환합니다. CFR API는 이미지에 있는 얼굴 인식하여

 https://developers.naver.com/docs/clova/api/CFR/API_Guide.md#Examples

코드 가져오기

NAVER Developers Products Documents Application NAVER D2 Support Forum API 상태 Search Here 

Clova Face Recognition

CFR API란?

사전 준비사항

CFR API 사용하기

CFR API 레퍼런스

구현 예제


Java

PHP


Node.js

Python

C#

Python 

```
import os
import sys
import requests
client_id = "YOUR_CLIENT_ID"
client_secret = "YOUR_CLIENT_SECRET"
# url = "https://openapi.naver.com/v1/vision/face" // 얼굴감지
url = "https://openapi.naver.com/v1/vision/celebrity" // 유명인 얼굴인식
files = {'image': open('YOUR_FILE_NAME', 'rb')}
headers = {'X-Naver-Client-Id': client_id, 'X-Naver-Client-Secret': client_secret }
response = requests.post(url, files=files, headers=headers)
rescode = response.status_code
if(rescode==200):
    print (response.text)
else:
    print("Error Code:" + rescode)
```

C# 

```
import os
import sys
import requests
client_id = "API_ID"
client_secret = "API_비밀번호"
url = "https://openapi.naver.com/v1/vision/face" # 얼굴감지
# url = "https://openapi.naver.com/v1/vision/celebrity" # 유명인 얼굴인식
files = {'image': open('이미지 파일', 'rb')}
headers = {'X-Naver-Client-Id': client_id, 'X-Naver-Client-Secret': client_secret }
response = requests.post(url, files=files, headers=headers)
rescode = response.status_code
if(rescode==200):
    print (response.text)
else:
    print("Error Code:" + rescode)
```

애플리케이션 등록하기

내 애플리케이션

애플리케이션 등록

API 재휴 신청

계정 설정

애플리케이션 등록 (API 이용신청)

애플리케이션의 기본 정보를 등록하면, 좌측 **내 애플리케이션** 메뉴의 서브 메뉴에 등록하신 애플리케이션 이름으로 서브 메뉴가 만들어집니다.

애플리케이션 이름	<div>애플리케이션 이름</div> <ul style="list-style-type: none"> 네이버 로그인할 때 사용자에게 표시되는 이름이므로 서비스 브랜드를 대표할 수 있는 이름으로 가급적 10자 이내로 간결하게 설정해주세요. 40자 이내의 영문, 한글, 숫자, 공백문자, 쉼표(,), /, "-", "_", 만 입력 가능합니다.
사용 API	<div>선택하세요.</div> <div>사용할 API를 추가해 주세요.</div>

API ID & 비밀번호 가져오기

내 애플리케이션

다른 꿀 찾기

애플리케이션 등록

API 재휴 신청

계정 설정

다른 꿀 찾기

개요	API 설정	멤버관리	로그인 통계	API 통계
----	--------	------	--------	--------

애플리케이션 정보

Client ID	c5ecgBDG1VIMwk5wrQc0
Client Secret	<div> <div>••••••••••</div> <div>보기</div> </div>

연예인 다른꿀 찾기

```

import os
import sys
import requests
client_id = "c5ecgBDG1vLMwk5wrQc0"
client_secret = "vNi69mAbiu"
# url = "https://openapi.naver.com/v1/vision/face" # 얼굴감지
url = "https://openapi.naver.com/v1/vision/celebrity" # 유명인 얼굴인식
files = {'image': open('수정후.jpg', 'rb')}
headers = {'X-Naver-Client-Id': client_id, 'X-Naver-Client-Secret': client_secret }
response = requests.post(url, files=files, headers=headers)
rescode = response.status_code
if(rescode==200):
    print (response.text)
else:
    print("Error Code:" + rescode)

```

```

import matplotlib.pyplot as plt
import matplotlib.image as mpimg

img = mpimg.imread('수정후.jpg')

plt.imshow(img)

```

```

import json

parsed = json.loads(response.text)

parsed

```

얼굴 분석

```

import os
import sys
import requests
client_id = "c5ecgBDG1vLMwk5wrQc0"
client_secret = "vNi69mAbiu"
url = "https://openapi.naver.com/v1/vision/face" # 얼굴감지
# url = "https://openapi.naver.com/v1/vision/celebrity" # 유명인 얼굴인식
files = {'image': open('수정후.jpg', 'rb')}
headers = {'X-Naver-Client-Id': client_id, 'X-Naver-Client-Secret': client_secret }
response = requests.post(url, files=files, headers=headers)
rescode = response.status_code
if(rescode==200):
    print (response.text)
else:
    print("Error Code:" + rescode)

```

```

import json

result = json.loads(response.text)

result

result.keys() # 딕셔너리에서 키값 추출하기

result['faces']

```

```

result['faces'][0]

result['faces'][0]['roi'] # 얼굴 위치 정보

# x, y : 얼굴의 x좌표 & y좌표
# width, height : 너비 & 높이

result['faces'][0]['gender'] # 성별 정보와 정확도

result['faces'][0]['age'] # 나이 정보와 정확도

x, y, w, h = result['faces'][0]['roi'].values()
gender, g_conf = result['faces'][0]['gender'].values()
age, a_conf = result['faces'][0]['age'].values()

annotation = gender + " : " + str(g_conf)+ "\n" + age + " : " + str(a_conf)

annotation

import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import matplotlib.patches as patches

img = mpimg.imread('수정후.jpg')

fig, ax = plt.subplots(figsize=(10,10))
ax.imshow(img)
rect_face = patches.Rectangle((x,y),w,h,linewidth = 5, edgecolor='r',facecolor='none')

ax.add_patch(rect_face)
plt.text(10, 400, annotation, wrap=True, fontsize=18, color='white')

plt.imshow(img)

```

단체 사진 얼굴 분석

```

import matplotlib.pyplot as plt
import matplotlib.image as mpimg

img = mpimg.imread('family.jpeg')

plt.imshow(img)

import os
import sys
import requests
import json
client_id = "c5ecgBDG1VlMwk5wrQc0"
client_secret = "vNi69mAbiu"
url = "https://openapi.naver.com/v1/vision/face" # 얼굴감지
# url = "https://openapi.naver.com/v1/vision/celebrity" # 유명인 얼굴인식
files = {'image': open('family.jpeg', 'rb')}
headers = {'X-Naver-Client-Id': client_id, 'X-Naver-Client-Secret': client_secret }
response = requests.post(url, files=files, headers=headers)
# rescode = response.status_code
# if(rescode==200):
#     print (response.text)
# else:
#     print("Error Code:" + rescode)

result = json.loads(response.text)
result

```

```

import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import matplotlib.patches as patches

img = mpimg.imread('family.jpeg')
fig, ax = plt.subplots(figsize=(10,10))
ax.imshow(img)

for each in result['faces']:

    x, y, w, h = each['roi'].values()
    gender, g_conf = each['gender'].values()
    age, a_conf = each['age'].values()

    rect_face = patches.Rectangle((x,y),w,h,linewidth = 3, edgecolor='r',facecolor='none')

    annotation = gender + " : " + str(g_conf)+ "\n" + age + " : " + str(a_conf)

    plt.text(x, y+h+40, annotation, wrap=True, fontsize=10, color='black')

    ax.add_patch(rect_face)

# plt.show(img)

```

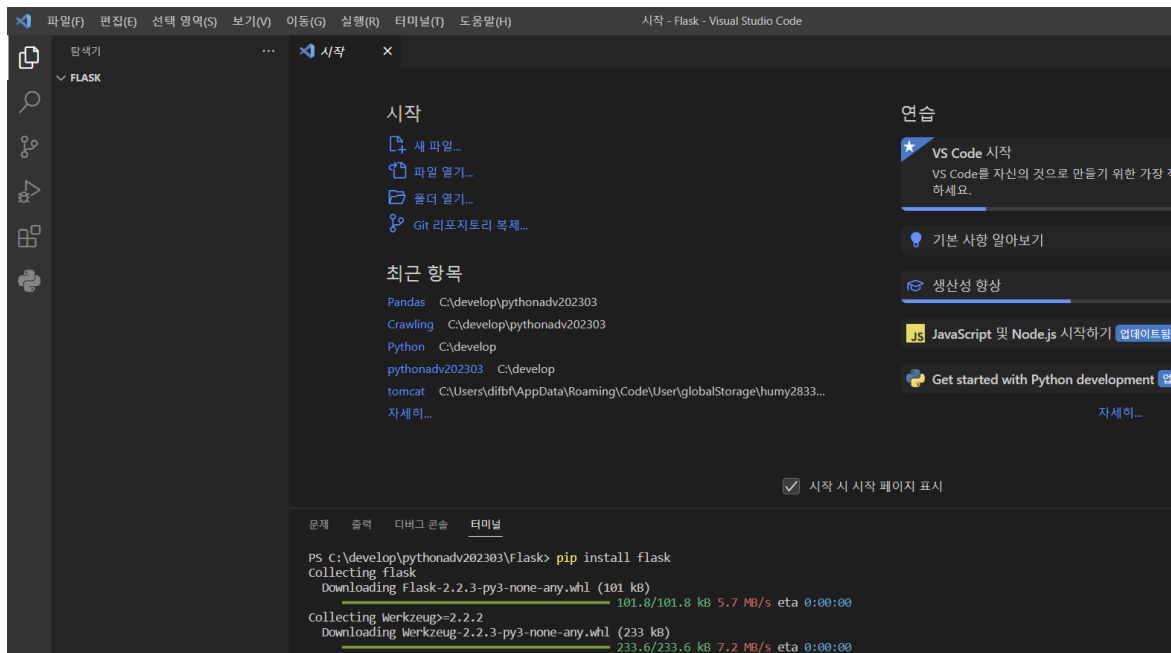
▼ Flask



2004년 오스트리아의 개발자[아르민 로나허(Armin Ronacher)]가 만든 Python Web Framework이다.

Python에서 장고[Django]와 더불어 웹 개발의 양대 산맥이다.

Download Flask



실행하기

```

hello_flask.py X
hello_flask.py > ...
1  from flask import Flask
2
3  app = Flask(__name__)
4
5  @app.route('/')
6  def hello() -> str:
7      return 'Hello World from Flask!'
8
9  app.run()
```

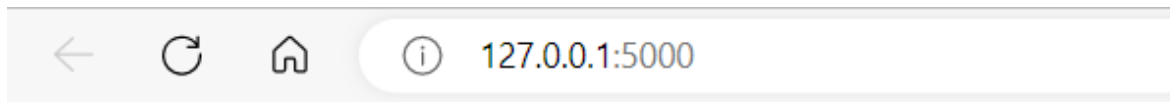
```

from flask import Flask

app = Flask(__name__)

@app.route('/')
def hello() -> str:
    return 'Hello World from Flask!'

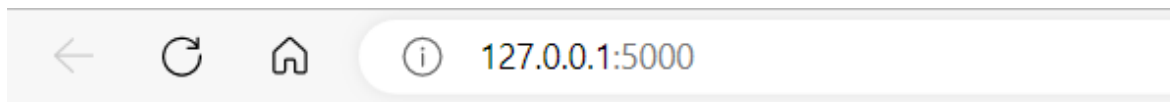
app.run()
```



Hello World from Flask!

실행하기(터미널)

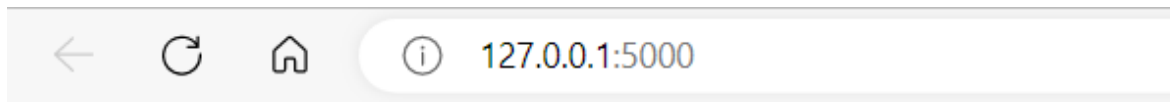
```
PS C:\develop\pythonadv202303\Flask> py -3 hello_flask.py
* Serving Flask app 'hello_flask'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```



Hello World from Flask!

실행하기2(터미널)

```
PS C:\develop\pythonadv202303\Flask> python hello_flask.py
* Serving Flask app 'hello_flask'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```



Hello World from Flask!

▼ Flask로 웹페이지 만들기(PythonAnywhere)



해당 내용은 [Head First Python] 교재를 참조하였습니다.

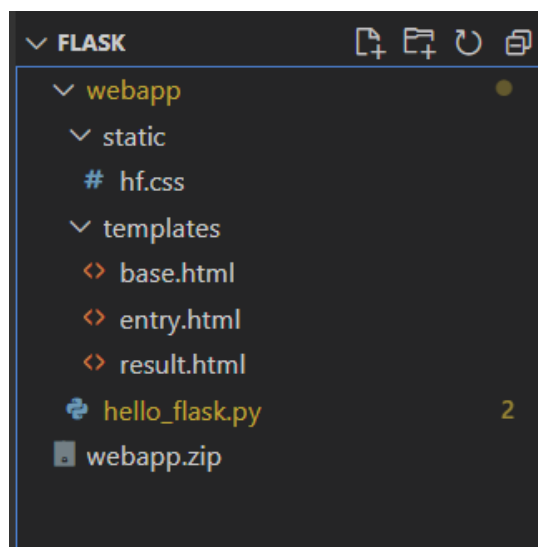
Head First Python - YES24

최신 인지과학이론을 바탕으로 반복학습/재미요소를 모두 갖춘 파이썬 학습서 이 책은 『Head First』 시리즈 학습 원리에 기초하여 모든 페이지가 그림과 대화체를 바탕으로 구성되어 있습니다. 딱딱한 강의식 말투보다는 개인적으로 대화를 나

 <http://www.yes24.com/Product/Goods/57435424>



파일 경로



hello_flask.py

```
from flask import Flask, render_template
# from vsearch import search4letters

app = Flask(__name__)

@app.route('/')
def hello() -> str:
    return 'Hello World from Flask!'

# @app.route('/search4', method=['POST'])
# def do_search() -> str:
#     return str(search4letters('life, the universe, and everything', 'eiru,!'))

@app.route('/entry')
def entry_page() -> 'html':
    return render_template('entry.html', the_title='Welcome to search4letters on the web!')

# app.run()
```

base.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>{{ the_title }}</title>
    <link rel="stylesheet" href="static/hf.css" />

  </head>
  <body>
    {% block body %}

    {% endblock %}
  </body>
</html>
```

entry.html

```
{% extends 'base.html' %}

{% block body %}

<h2>{{ the_title }}</h2>

<form method='POST' action='/search4'>
<table>
<p>Use this form to submit a search request:</p>
<tr><td>Phrase:</td><td><input name='phrase' type='TEXT' width='60'></td></tr>
<tr><td>Letters:</td><td><input name='letters' type='TEXT' value='aeiou'></td></tr>
</table>
<p>When you're ready, click this button : </p>
<p><input value='Do it!' type='SUBMIT'></p>
</form>
```

```
{% endblock %}
```

result.html

```
{% extends 'base.html' %}

{% block body %}

<h2>{{ the_title }}</h2>

<p>You submitted the following data:</p>
<table>
<tr><td>Phrase:</td><td>{{ the_phrase }}</td></tr>
<tr><td>Letters:</td><td>{{ the_letters }}</td></tr>
</table>

<p>When "{{the_phrase }}" is search for "{{ the_letters }}", the following results are returned:</p>
<h3>{{ the_results }}</h3>

{% endblock %}
```

hf.css

```
body {
    font-family:    Verdana, Geneva, Arial, sans-serif;
    font-size:      medium;
    background-color: tan;
    margin-top:      5%;
    margin-bottom:   5%;
    margin-left:     10%;
    margin-right:    10%;
    border:          1px dotted gray;
    padding:         10px 10px 10px 10px;
}
a {
    text-decoration: none;
    font-weight:     600;
}
a:hover {
    text-decoration: underline;
}
a img {
    border:          0;
}
h2 {
    font-size:       150%;
}
table {
    margin-left:     20px;
    margin-right:    20px;
    caption-side:    bottom;
    border-collapse: collapse;
}
td, th {
    padding:         5px;
    text-align:      left;
}
.copyright {
    font-size:       75%;
}
```

```

    font-style:      italic;
}
.slogan {
    font-size:       75%;
    font-style:      italic;
}
.confirmentry {
    font-weight:     600;
}

/** Tables **/

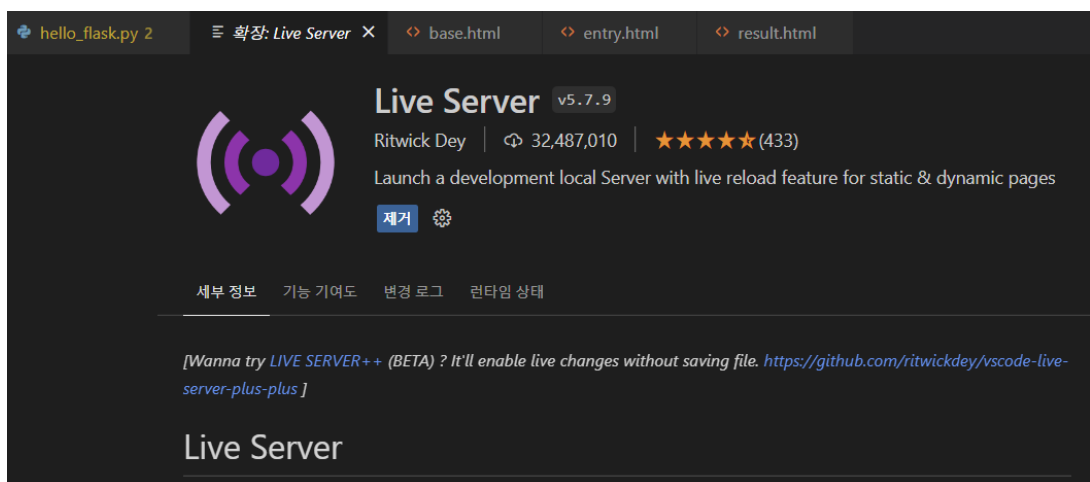
table {
    font-size:       1em;
    background-color: #fafcff;
    border:          1px solid #909090;
    color:           #2a2a2a;
    padding:         5px 5px 2px;
    border-collapse: collapse;
}

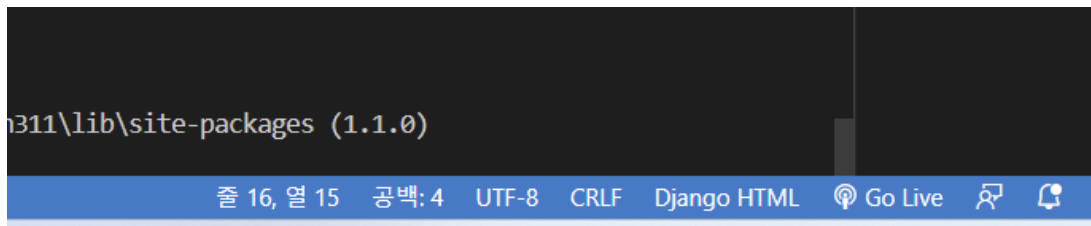
td, th {
    border:          thin dotted gray;
}

/** Inputs **/
input[type=text] {
    font-size:       115%;
    width:           30em;
}
input[type=submit] {
    font-size:       125%;
}
select {
    font-size:       125%;
}

```

▼ 웹페이지 실행 결과 바로 확인하기



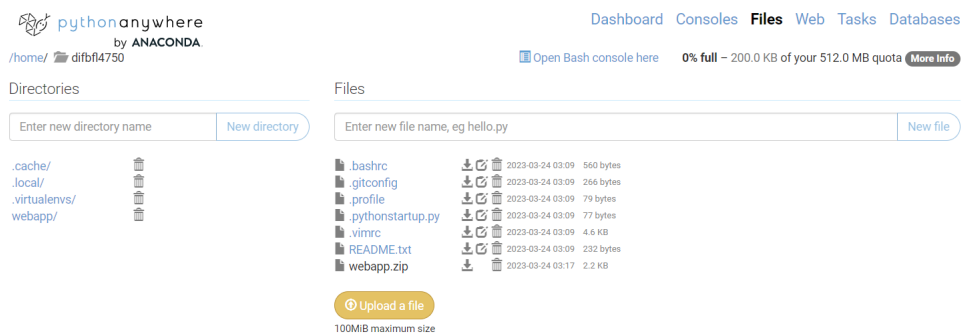


PythonAnywhere 회원가입

Host, run, and code Python in the cloud: PythonAnywhere

<https://www.pythonanywhere.com/>

파일 업로드



압축풀기 & 파일 이동

```
03:21 ~ $
03:21 ~ $
03:21 ~ $
03:21 ~ $
03:21 ~ $
03:21 ~ $
03:21 ~ $
03:21 ~ $
03:21 ~ $
03:21 ~ $
03:21 ~ $ unzip webapp.zip mv webapp/* mysite
Archive: webapp.zip
  inflating: webapp/hello_flask.py
   creating: webapp/static/
  inflating: webapp/static/hf.css
   creating: webapp/templates/
  inflating: webapp/templates/base.html
  inflating: webapp/templates/entry.html
  inflating: webapp/templates/result.html
caution: filename not matched: mv
caution: filename not matched: mysite
03:22 ~ $ mv webapp/* mysite
03:33 ~ $
```

새로운 Web App 만들기

difbf14750.pythonanywhere.com

[Add a new web app](#)

Configuration for difbf14750.pythonanywhere.com

Reload:

[Reload difbf14750.pythonanywhere.com](#)

Best before date:

We're happy to host your free website – and keep it free – for as long as you want to keep it running, but you'll need to log in at least once every three months and click the "Run until 3 months from today" button below. We'll send you an email a week before the site is disabled so that you don't forget to do that. [See here for more details.](#)

This site will be disabled on **Saturday 24 June 2023**

[Run until 3 months from today](#)

[Paying users'](#) sites stay up forever without any need to log in to keep them running.

실행

← ↻ 🏠 ⚠️ 안전하지 않음 | difbfl4750.pythonanywhere.com/entry

Welcome to search4letters on the web!

Use this form to submit a search request:

Phrase:

Letters:

When you're ready, click this button :

Do it!