

웹페이지 만들기



Streamlit : 파이썬으로 빠르고 쉽게 웹페이지를 만들수 있는 라이브러리

ngrok : 로컬 웹사이트를 빠르고 쉽게 온라인으로 접속할 수 있게 해주는 서비스

```
!pip install streamlit -q  
!pip install pyngrok
```

• Streamlit이란?

: 2019년 하반기에 탄생한, 파이썬 기반 웹 어플리케이션 툴입니다. 데이터사이언스/머신러닝 프로젝트를 웹 어플리케이션에 배포하는 목적으로 아주 편리하고 강력한 기능을 제공하고 있습니다.

• Streamlit의 특징은?

1. 웹 개발에 딥-다이브한 지식이 없어도 됩니다.
2. 파이썬 문법을 사용하기 때문에, 파이썬 언어 사용자라면 누구나 쉽게 사용 가능합니다.
3. 빠르게 시각화 등의 웹 어플리케이션 기능을 구축할 수 있고 클라우드 환경에 배포 하기 쉽습니다.

• Streamlit의 단점은?

1. 특정 html 페이지에서 동작이 발생할 때마다 리로드 되는 형태로 동작하기 때문에, 정교하고, 고성능의 웹 어플리케이션을 구축하는 용도와는 거리가 있습니다.
2. SEO 검색 최적화를 할 수 없습니다.

Python으로 웹페이지 만들기

Python

ngrok 사이트에서 토큰 발급



Python으로 웹페이지 만들기

Python

ngrok 사이트에서 토큰 발급

<https://ngrok.com/>

Your Authtoken - ngrok

<https://dashboard.ngrok.com/get-started/your-authtoken>

남은재

Getting Started
Setup & Installation
Your Authtoken
Cloud Edge
Tunnels
Events
API
Security
Team
Billing
Settings

A verification link has been sent to ddobaki@outlook.kr. Additional restrictions will apply to your account until you confirm your email address.
[Resend Verification](#)

Your Authtoken

This is your personal Authtoken. Use this to authenticate the ngrok agent that you downloaded.

2GmyjJX8DkRTnoIA3RWn04IoCn6_64oRQaYxPndxZX7DWFgF9 [Copy](#)

Command Line

Authenticate your ngrok agent. You only have to do this once. The Authtoken is saved in the default configuration file.

```
$ ngrok config add-authtoken 2GmyjJX8DkRTnoIA3RWn04IoCn6_64oRQaYxPndxZX7DWFgF9
```

Configuration File

Alternatively, you can directly add the Authtoken to your `ngrok.yml` configuration file. Use `ngrok config edit` to open the file.

```
# in ngrok.yml
```

Get access to powerful features like:

- URLs that don't change
- Your own custom domains
- and more!

[Upgrade Now](#)

```
import pyngrok
import ngrok
ngrok.set_auth_token('??????')
```

Ngrok 인증토큰 셋팅

```
import streamlit as st
import tensorflow as tf
from PIL import Image, ImageOps
import numpy as np
from tensorflow.keras.applications.imagenet_utils import
decode_predictions
```

PIL : Python Imaging Library – 파이썬에서 여러 포맷의 이미지를 읽고 조작을 돕는 라이브러리

decode_predictions: 인공지능이 예측한 결과를 어떤 이미지와 매핑이 되는지 해석을 도와주는 함수

```
resnet50_pre = tf.keras.applications.resnet.ResNet50(weights='imagenet', input_shape=(224,224,3))
```

ResNet50: 이미지 분류 인공지능 알고리즘, 2015년 이미지 인식대회(ILSVRC)에서 우승차지

```
st.title('이미지 분류 실습 웹페이지')  
# 파일을 업로드할수 있는 영역 생성  
file = st.file_uploader('이미지를 선택해 주세요.', type=['jpg', 'png'])
```

```
if file is None:
    st.text('이미지를 먼저 선택해 주세요.')
else:
    image = Image.open(file)
    st.image(image, use_column_width=True)
    img_resized = ImageOps.fit(image, (224,224), Image.ANTIALIAS)
    #ImageOps함수를 통해 이미지 사이즈 변경
    img_resized = img_resized.convert('RGB')
    #image load시 4개의 채널로 되어 있음. 인공지능이 인식할수 있는 3개의 채널로 변경
    img_resized = np.asarray(img_resized)
    #인공지능이 원하는형식으로의 변환을 위해 np array로 변환
```

이미지 선택여부에 따른 처리


```
pred = resnet50_pre.predict(img_resized.reshape([1,224,224,3]))

decoded_pred = decode_predictions(pred)    #예측 결과를 이미지와 맵핑

results = ''
for i, instance in enumerate(decoded_pred[0]):
    results += '{}위: {} ({:.2f}%)' .format(i+1, instance[1], instance[2] * 100)
st.success(results)
```

이미지 선택여부에 따른 처리

```
%%writefile image_01.py
```

Image_01.py 로 저장

```
!ls      # 현재 디렉토리 확인

!cat image_01.py      #파일 내용 확인

# 웹페이지 업로드  nohup: 끊기지 않기 위한 명령어
!nohup streamlit run image_01.py --server.port 80 &

#온라인에서 접속가능하기 위한 작업
url = ngrok.connect(port='80')
url
```