

파이썬 실행파일 만들기

[(실습) projects 폴더 말고 별도의 폴더(pyinstaller)를 만들어 그 안에 빌드할 .py 파일별로 해당폴더를 만들어 실행 pyinstaller/hello_cli & pyinstaller/ex17_gui_tkinter.py]

1. PyInstaller 설치

우선 명령프롬프트에서 아래의 명령어를 통해 PyInstaller 패키지 설치

```
> pip3 install pyinstaller
```

2. 실행파일 만들기

python 파일이 있는 폴더로 이동한 다음, 아래 명령어를 입력하면 해당 폴더에 실행파일이 만들어짐.

```
> pyinstaller 파일명.py
```

3. 만들어진 폴더(build, dist)에서 dist로 이동

- exe 실행파일 더블클릭으로 실행.

- 콘솔창이 뜨면서 GUI 프로그램이 실행됨

4. 콘솔창 출력되지 않도록 하기

콘솔창이 출력되지 않게 하려면 아래와 같이 명령어에 '-w' 또는 '--windowed'를 추가해줍니다.

```
pyinstaller -w 파일명.py
```

5. 실행파일 하나만 생성하기

실행파일 하나만 생성하기 위해서는 아래와 같이 명령어에 '-F' 또는 '--onefile'을 추가합니다.

```
pyinstaller -w -F 파일명.py
```

다시 dist 폴더로 이동하면 실행파일 하나만 생성된 것을 볼 수 있음.

프로그램에 이미지, 동영상 등 미디어 파일이나 폰트파일 등의 별도 파일경로가 존재하는 경우

pyinstaller가 exe를 만들때 임시 폴더에 파일들을 이동시켜 위치하기에 경로 문제로 앱이 실행되지 않음.
해결하려면..

1. 임시경로와 현재파일의 상대경로를 연결해주는 resource_path()함수를 만들어서 사용해야 함.

2. 파일경로가 쓰여지는 코드의 경로에 resource_path(상대경로)를 지정해 줘야 함.

3. pyinstaller 빌드할때. --add-data 옵션을 통해 resources 폴더를 지정해 줘야 함.

[(실습) pyinstaller/tkinter_image 폴더 만들고 새로 만들어.. 이미지만 로드하여 보여주는 .py 만들고 .. 실행파일까지 만들기.....]

```
-----  
import sys  
import os  
  
def resource_path(relative_path):  
    """ PyInstaller로 생성된 실행 파일에서 리소스 경로를 찾는 함수 """  
    try:  
        # PyInstaller에 의해 임시 폴더에 압축 해제된 경로 --- 애플리케이션에 포함된 리소스들이 임시로  
        # 압축 해제되는 임시 폴더의 경로를 가리키는 내부적인 변수  
        base_path = sys._MEIPASS # MEI는 PyInstaller의 전신인 McMillan Enterprise Installer의 약자 -  
        McMillan Enterprise는 PyInstaller 프로젝트의 초기 개발과 관련된 이름  
    except Exception:  
        # 원본 .py 파일 실행 시의 기준 경로  
        base_path = os.path.abspath(".")  
    return os.path.join(base_path, relative_path)  
  
# resources 폴더안에 하위 폴더로 image, audio, video 같은 폴더를 생성하여 해당 리소스를 위치..  
# pyinstaller 빌드 시 --add-data 옵션 사용  
# 이미지 파일과 같은 앱의 리소스가 있는 폴더(예: resouces/image 폴더)를 PyInstaller 명령어에 포함  
  
# Windows 사용자  
# pyinstaller --add-data "원본_경로;대상_경로" your_script.py  
  
# macOS 및 Linux 사용자  
# pyinstaller --add-data "원본_경로:대상_경로" your_script.py  
  
# > pyinstaller --windowed --onefile --add-data "resources;resources" tkinter_image.py  
# -----
```