

Haladó programozás beadandó

Feladat: Alkalmazás készítése python-ban, amivel szerkeszteni lehet képeket.

Szerkesztési lehetőségek: Kép méretének, színének változtatása és forgatása.

Kezdésnek elmondanánk, hogy először két verziót csináltunk. Az első az volt, hogy az összes képet, amik a „kepek” mappában találhatók automatikusan szerkeszti egységesen, ahogy a kódban előre megvan adva. Ez a verzió is hibátlanul működött, de nekünk nem tetszett, mivel így nem lehetett minden képet egyénileg szerkeszteni.

A második verzió pedig az, hogy a „kepek” mappából tudjuk kiválasztani melyik képet, hogyan szeretnénk szerkeszteni a Visual Studio Code terminálon keresztül. Ez azért sokkal praktikusabb, mert külön terminálon keresztül tudunk képeket szerkeszteni és nem kell belenyúlni a kódba. Ettől sokkal inkább olyan lesz, mint egy script.

0. lépés: Python Pillow telepítése, ez számunkra azért fontos, hogy a képeket meg nyitni és szerkeszteni tudjuk.

```
● PS C:\Users\Zsolt\Desktop\Egyetem\Haladó prog\Képeszerkesztő python> pip install pillow
Collecting pillow
  Downloading pillow-12.0.0-cp314-cp314-win_amd64.whl.metadata (9.0 kB)
  Downloading pillow-12.0.0-cp314-cp314-win_amd64.whl (7.1 MB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.1/7.1 MB 49.6 MB/s 0:00:00
Installing collected packages: pillow
Successfully installed pillow-12.0.0
```

A kód egyes részei:

```
import os
from PIL import Image
import argparse
```

← tudunk dolgozni a képekkel, Python Pillow

← parancssori argumentumok

Ezek elengedhetetlenek a program futtatásához.

```
# SZÍNEK
filterek = {
    "none": (0, 0, 0),
    "piros": (255, 0, 0),
    "zold": (0, 255, 0),
    "kek": (0, 0, 255),
```

Következő a filterek definiálása, ezt azért fontos előre megadni, hogy tudjuk használni magát a színt (ékezetek nélkül) a parancssor futtatásánál, hogy ne kelljen fejből tudni például a narancssárga RGB kódját.

A parancssori argumentumok definiálása:

Az első sorral magát az argumentumot, hozzuk létre.

A következő 4 sorral adunk hozzá „kifejezéseket” az argumentumhoz. Például a „--kep” kifejezéssel hivatkozunk az adott argumentumra amikor a parancssorba írunk, a „required=True”, pedig azt csinálja, hogy kötelező legyen ezt megadni.

Az utolsó sor, pedig beolvassa és változóba menti az argumentumot.

```
# PARANCSOSI ARGUMENTUMOK
parser = argparse.ArgumentParser(description="Képszerkesztő")

parser.add_argument("--kep", type=str, required=True,
                    help="A szerkesztendő kép fájlneve a kepek mappából")

parser.add_argument("--meret", type=int, default=1000,
                    help="Új szélesség pixelben")

parser.add_argument("--forgatas", type=int, default=0,
                    help="Forgatás szöge (0-360)")

parser.add_argument("--szin", type=str, default="none",
                    choices=list(filtererek.keys()),
                    help="Filter színe")

args = parser.parse_args()
```

Következőnek a mappák elérését adtuk meg és azt, hogy el legyenek mentve a használható fájl formátumok. Utóbbi a hiba kezelés során fontos.

```
# MAPPÁK ÉS FÁJLTÍPUSOK
bemeneti_mappa = "kepek"
kimeneti_mappa = "szerkesztett_kepek"

elfogadott_formatum = (".jpg", ".jpeg", ".png", ".bmp", ".webp")
```

A kép helyének definiálása. #KÉP BEOLVASÁSA
kep_helye = os.path.join(bemeneti_mappa, args.kep)

A hibák kezelése.

```
# HIBÁK KEZELÉSE
if not os.path.exists(kep_helye):
    print(f"Hiba: A '{args.kep}' fájl nem található a kepek mappában!")
    exit()

if not args.kep.lower().endswith(valid_ext):
    print("Hiba: A megadott fájl nem támogatott képformátum.")
    exit()
```

A kép megnyitása, amin éppen dolgozunk. Nem összekeverendő az „args.kep”-pel.

```
# KÉP MEGNYITÁSA
kep = Image.open(kep_helye)
```

Ezután megadtuk a funkcióinkat:

A program először elmenti az eredeti kép szélességét és magasságát. A méret változtatását úgy oldottuk meg, hogy ha csak a szélességet lehet változtatni és ehhez igazodik a magasság. Ezzel elérve, hogy ne torzuljon a kép.

A forgatás automatikusan átméretezi, hogy ne veszítsen a kép méretéből semmit. („expand=True”) és csak akkor fut le, ha nem nullát adtunk meg.

A filter is csak akkor fut le, ha változtattunk színt és ez egy úgynevezett overlay-t használ, hogy az eredti színeken felül legyen egy plusz szín, pontosan, ahogy egy filter működne. Az „image.blend” illeszti rá a filtert a képre. Az „alpha=0.35” határozza meg, hogy mennyire legyen áttetsző ez a filter.

```
# MÉRET
w, h = kep.size
scale = args.meret / w
kep = kep.resize((args.meret, int(h * scale)))

# FORGATÁS
if args.forgatas != 0:
    kep = kep.rotate(args.forgatas, expand=True)

# FILTER ALKALMAZÁSA
if args.szin != "none":
    r, g, b = filterek[args.szin]
    overlay = Image.new("RGB", kep.size, (r, g, b))
    kep = Image.blend(kep, overlay, alpha=0.35)
```

Végére maradt a mentés és a parancssori visszajelzés.

A mentés funkciót úgy csináltuk meg, hogy az eredi kép nevéhez fűzze hozzá a „_szerkesztett”, és a „v1” kifejezéseket, valamint, ha ugyanazt a képet mentjük többször akkor növelje a verziószámot.

```
# MENTÉS
nev, kiterj = os.path.splitext(args.kep)
alap_nev = f"{nev}_szerkesztett"

verzio = 1
uj_nev = f"{alap_nev}_v{verzio}{kiterj}"
mentes_helye = os.path.join(kimeneti_mappa, uj_nev)

while os.path.exists(mentes_helye):
    verzio += 1
    uj_nev = f"{alap_nev}_v{verzio}{kiterj}"
    mentes_helye = os.path.join(kimeneti_mappa, uj_nev)

kep.save(mentes_helye)

print(f"Kész! Mentve ide: {mentes_helye}")
```

[Utolsó oldalon található egy példa futtatás minden funkció használatával.](#)

Parancssori futtatásra példa:

„python main.py --kep kep1.webp --meret 2000 --forgatas 180 --szin piros”

```
PS C:\Users\zsolt\Desktop\Egyetem\Haladó prog\Képeszerkesztő python> python main.py --kep kep1.webp -  
● -meret 2000 --forgatas 180 --szin piros  
Kész! Mentve ide: szerkesztett_kepek\kep1.webp  
○ PS C:\Users\zsolt\Desktop\Egyetem\Haladó prog\Képeszerkesztő python>
```

Eredeti kép: (275x183 pixel)



Szerkesztett kép: (2000x1330 pixel)



Források:

chatgpt:

- v5.1-et használtunk a parancssori argumentumok definiálására.
- v5.1-et használtunk a filterek alkalmazásának megoldására.

google:

- <https://www.geeksforgeeks.org/python/python-pillow-a-fork-of-pil/>