Automation of Repetitive Tasks

Tshimologong - Animation Technical Director
Training for Python Programmers

Temweka Chirwa

12/02/2024

Objectives

1. Understand the significance of automating repetitive tasks in animation production.
2. Gain insights into best practices for writing efficient and reusable automation scripts in Python.
3. Learn how Python scripting can streamline workflow for multimedia processing tasks.

Why Automate?

- What does automate mean?
- Why/How does it help us?
- Examples of repetitive task
  - Examples in animation?

A good engineer is a lazy engineer

Work smarter not harder*

* Sometimes in order to work smart, you have to work hard

How to Automate

1. **Identify the Task -** Identify the task that you want to automate.
2. **Break Down the Task -** Break down the task into smaller subtasks that can be automated.
3. **Research -** Research the libraries and functions you can use to automate the subtasks.
4. **Write the Code -** Write the Python code to automate the subtasks.
5. **Test the Code -** Test the Python code to ensure it works as expected. This will involve running the code with different inputs and verifying the outputs.
6. **Run the Code -** Once you have tested it, you can run it to automate the task. Depending on the task, you can schedule the code to run at specific intervals
7. **Monitor and Update -** Monitor the task to ensure it runs smoothly. If there are any issues or errors, update the code to fix the problem.

Key Features of Automation Script

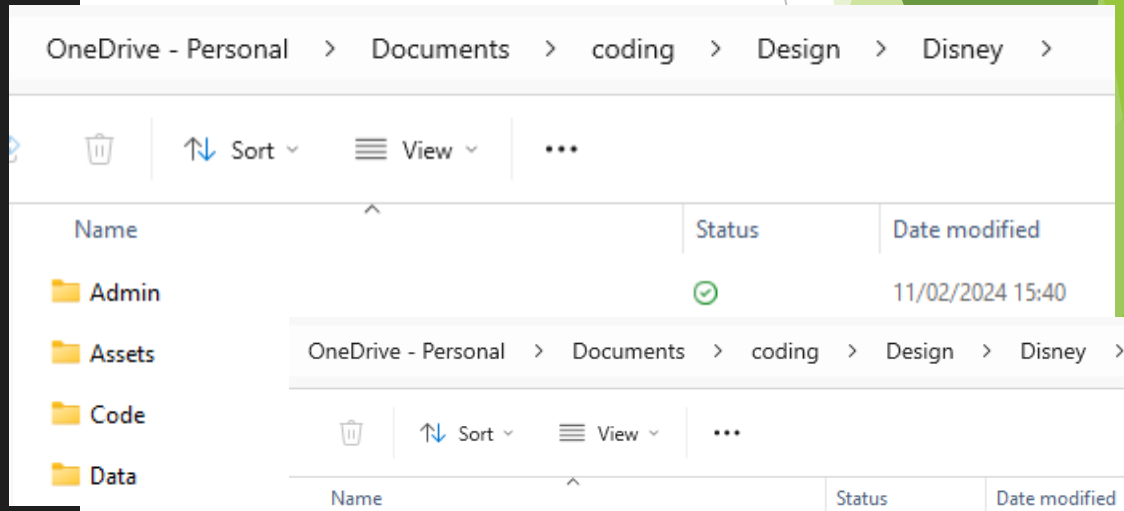
- Readability – clear and well documented
- Flexibility - adaptable to different environments, input data formats, and use cases
- Robustness - handle various edge cases and error scenarios gracefully
- Documentation - Comprehensive documentation, including usage instructions, code comments, and README files

Ex 1: Folder Template

- Each project folder should have the same format

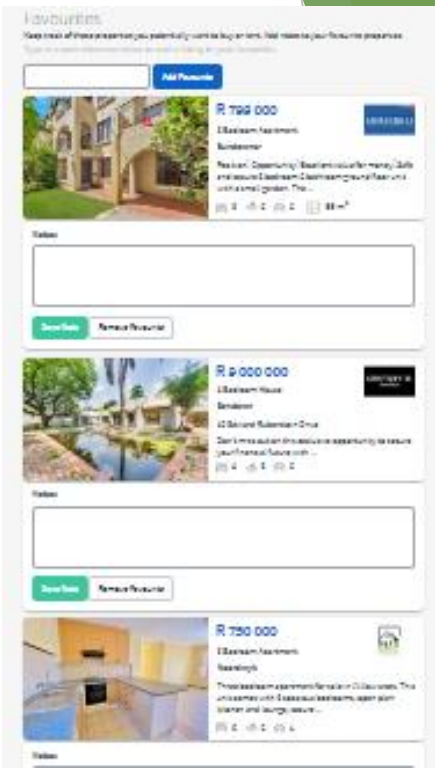
dir_creator.py > ...

```python
import os

main_dir = os.getcwd()

def create_folders(proj, path = "Design"):
    '''

    '''
    proj_dir = os.path.join(path, proj)

    try:
        os.mkdir(proj_dir)
    except FileExistsError:
        print("Error: Folder already exists, try a different name")

    #Internal Folders
    os.mkdir(os.path.join(proj_dir, "Assets"))
    os.mkdir(os.path.join(proj_dir, "Admin"))
    os.mkdir(os.path.join(proj_dir, "Data"))
    os.mkdir(os.path.join(proj_dir, "Code"))
    os.mkdir(os.path.join(proj_dir, "Code", "Code Reviews"))

    os.chdir(main_dir)


name = input("Name of the Project Directory you'll like to create: ")
proj_type = int(input("Type of Project you'll like to create: \n[1] Design [2] Data Analysis [3] Literature
```



Ex 2: Webscaping

- Extracting similar information from different websites/webpages



Webscraping Property24 for homes

In this second attempt, will try to scrape from my logged in account, in order to access my favourited properties.
Resources: How to Scrape Websites Behind a Login with Python by Shane Lee

```
import pandas as pd
import numpy as np
import requests
from bs4 import BeautifulSoup
```

```
with open('My Property24 - Favourites.html', 'r') as f:
    html_string = f.read()
```

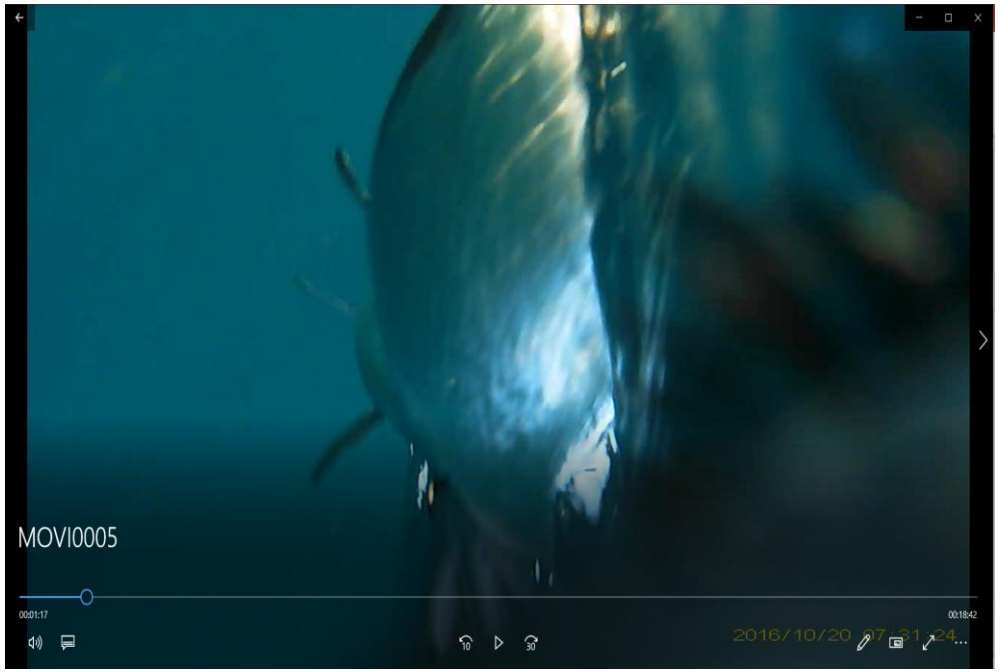
```
for_sale_list = []
url1 = 'https://www.property24.com'
url2 = url1 + ext

for list2 in list1:
    top10= str(list2.find("a", class_="")).split("\"")[0:10]

    for item in top10:
        if "/for-sale/" in item:
            ext = item
            break

    for_sale_list.append(url1+ext)
```

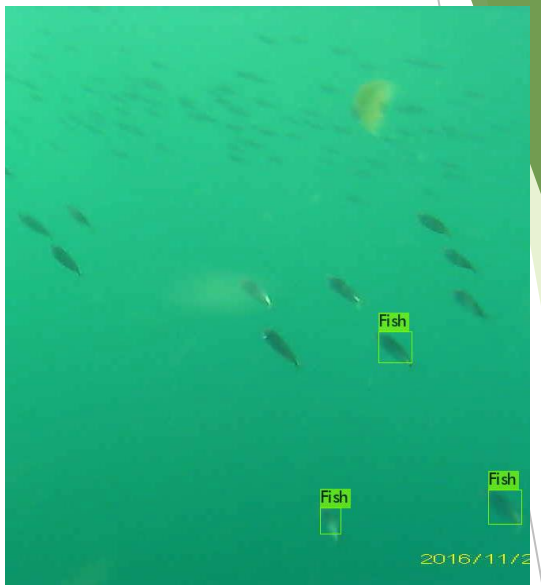
```
for_sale_list
```

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	ID	location	price	Type of Property	Listing Date	Floor Size	Price per mÂ²	Levies	Rates and Taxes	Pets Allowed	Bedrooms	Bathrooms	Garage
2	113806277	sharonlea	950000	Townhouse	05-Jan-24	162	0	3112	1039	-	3	2	-
3	113845464	north-riding	950000	Apartment / Flat	15-Jan-24	91	10440	2470	600	Yes	3	2	
4	113247411	vorna-valley	980000	Apartment / Flat	01-Sep-23	95	10316	1835	282	No	3	2	-
5	113269920	paulshof	993000	Apartment / Flat	07-Sep-23	87	11414	2816	0	No	3	2	-
6	108905623	maroeladal	885000	Apartment / Flat	06-Jun-23	102	8676	3523	633	-	3	2	-
7	112082292	noordwyk	999000	Apartment / Flat	01-Dec-22	119	8395	2000	620	No	3	2	
8	113358231	northwold	999000	Townhouse	22-Sep-23	0	0	0	0	No	3	2	-
9	113570043	northwold	999000	Townhouse	03-Nov-23	117	0	2398	592	Yes	3	2	-
10	112693740	northwold	995000	Apartment / Flat	23-May-23	94	10585	1923	500	Yes	3	2	-
11	112093071	north-riding-ah	950000	Townhouse	05-Dec-22	137	0	3476	720	-	3	2	
12	113539958	north-riding-ah	950000	Townhouse	27-Oct-23	71	0	1900	561	-	3	2	-
13	113335029	northwold	949000	Townhouse	18-Sep-23	119	0	3905	533	Yes	3	2	-
14	113451570	northwold	949000	Townhouse	13-Oct-23	119	0	5000	866	-	3	2	-
15	112732814	northwold	930000	Apartment / Flat	01-Jun-23	105	8857	604	2455	-	3	2	-
16	113554372	north-riding	930000	Apartment / Flat	31-Oct-23	0	0	2021	102	Yes	3	2	-
17	112074644	north-riding	885000	Townhouse	29-Nov-22	137	0	0	0	No	3	2	

Ex 3: Video Processing

- Extracting images from video footage





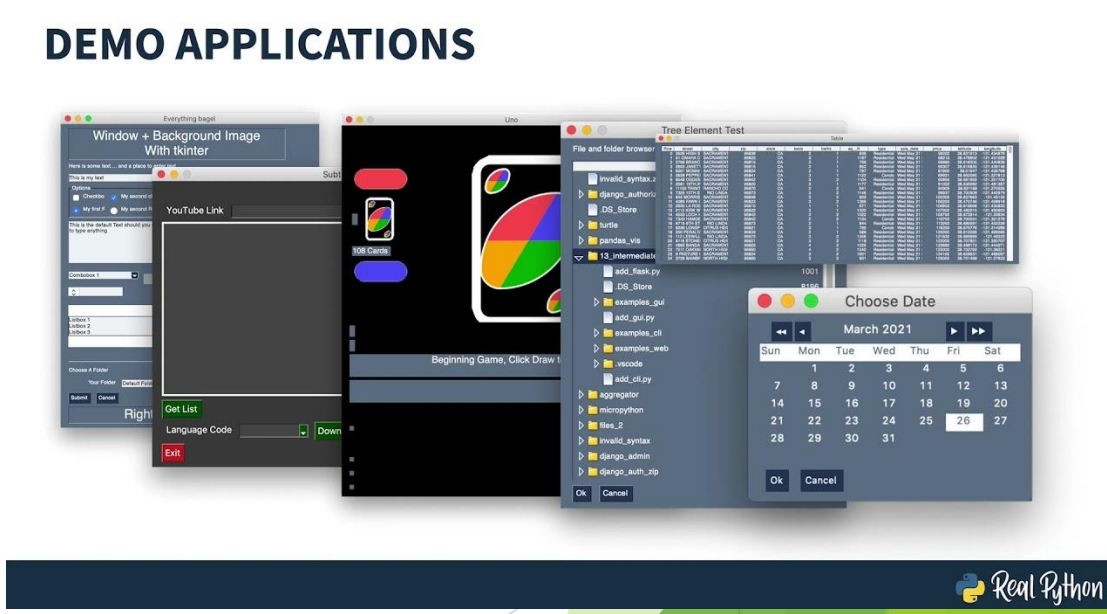
Useful link: https://www.ffmpeg.org/

Tip: Modular Programming

- Modular programming is a general programming concept where developers separate program functions into independent pieces.
- Pros:
  - Easier to read
  - Easier to test
  - Easily find things later
  - Easier to reuse

Tip: GUI widgets

- Graphical User Interface
- Visual Elements eg icons, buttons, input boxes
- Examples of GUI widgets: tkinter, pyQt, kivy



Useful links:
https://www.geeksforgeeks.org/python-gui-tkinter/
https://www.tutorialspoint.com/pyqt/index.htm

Self-Study

- Modular programming
- GUI toolkits eg PyQt, Tkinter etc
- APIs

Activity

- See Attached pdf
- Make a GUI for the To-do List
- Due: 17:00 16/02/2024

References and Resources

- Automate Everything With Python: A Comprehensive Guide to Python Automation
  URL: https://www.analyticsvidhya.com/blog/2023/04/python-automation-guide-automate-everything-with-python/
- Modular programming: beyond the spaghetti mess
  URL: https://www.tiny.cloud/blog/modular-programming-principle/
- Refactoring
  URL: https://refactoring.guru/refactoring/
- Python and PyQt: Building a GUI Desktop Calculator
  URL: https://realpython.com/python-pyqt-gui-calculator/
- FFmpeg
  URL: https://www.ffmpeg.org/