

Komendy wpisane w konsole:

```
cd c:\users\oskar\appdata\local\programs\python\python310\lib\site-packages
```

```
py pip install requests
```

Kod:

```
#przyklad dzialania:
#pageurl - pobrana strona, date-data w formacie rok miesiac dzien np.
20230126
#zapytanie do api:
#http://archive.org/wayback/available?url=example.com&timestamp=20060101
import webbrowser

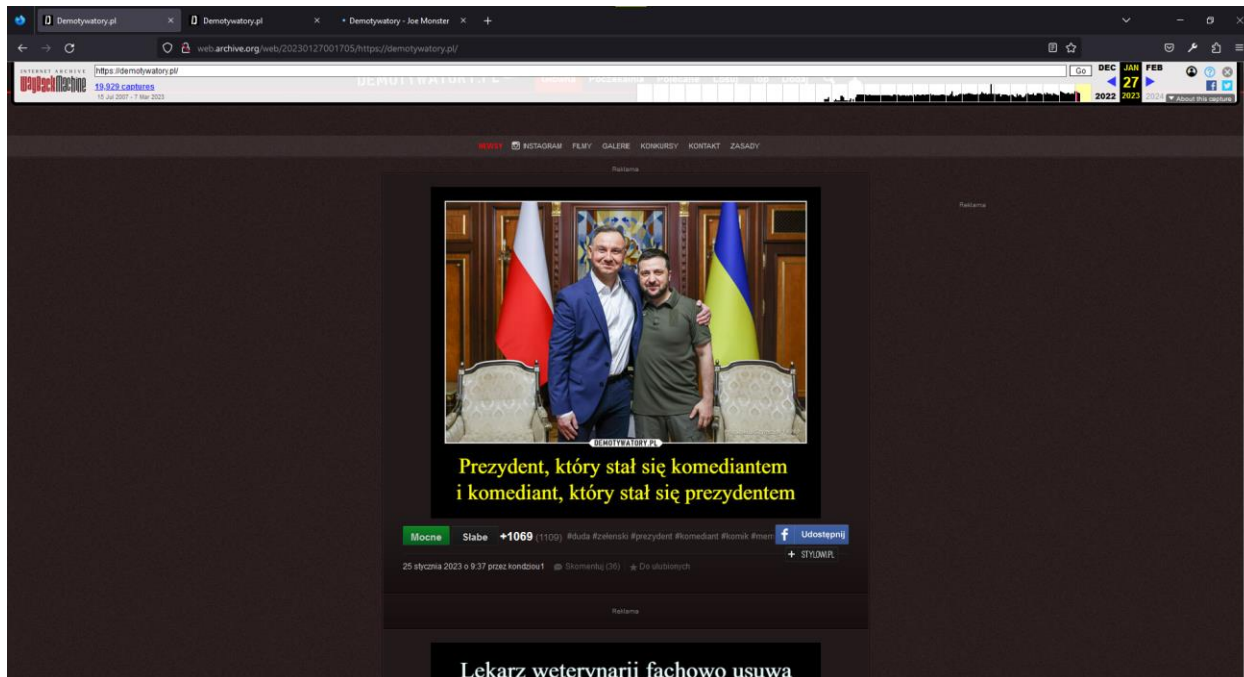
import requests as requests
pageurl = "demotywatory.pl"
url =
"http://archive.org/wayback/available?url="+pageurl+"&timestamp="+str(2023012
6)
response = requests.get(url)
d = response.json()
page = d["archived_snapshots"]["closest"]["url"]
webbrowser.open(page)

url =
"http://archive.org/wayback/available?url="+pageurl+"&timestamp="+str(2015012
6)
response = requests.get(url)
d = response.json()
page = d["archived_snapshots"]["closest"]["url"]
webbrowser.open(page)

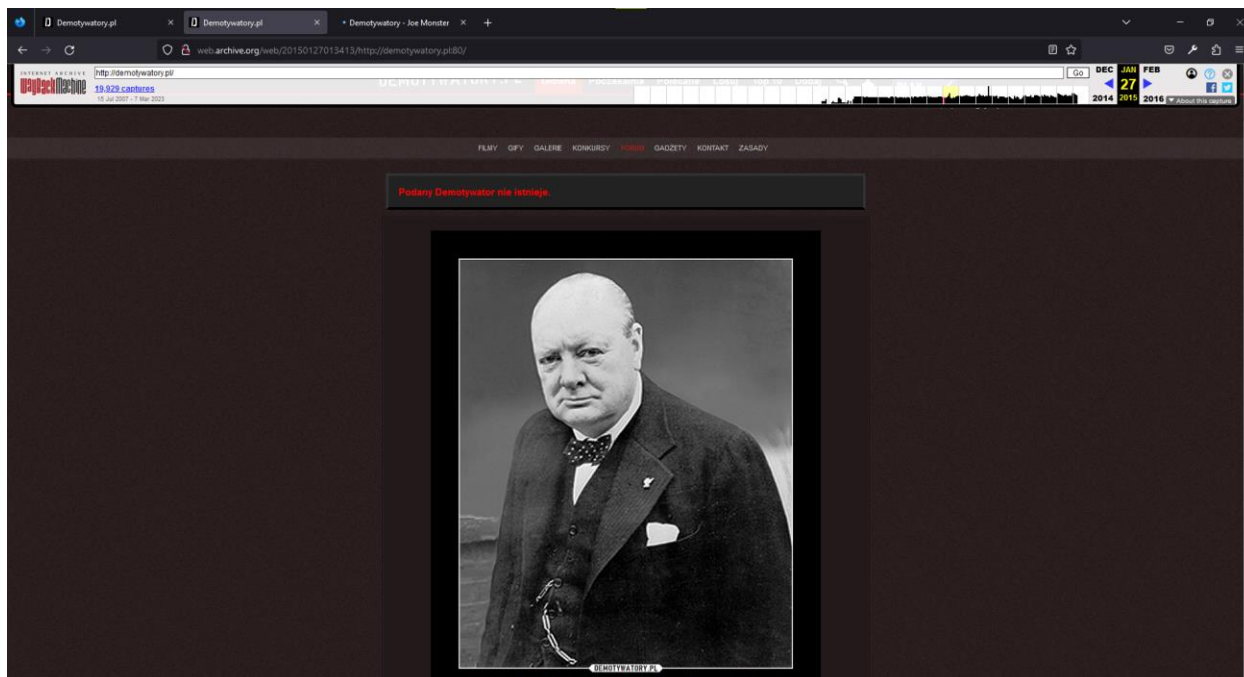
url =
"http://archive.org/wayback/available?url="+pageurl+"&timestamp="+str(2005012
6)
response = requests.get(url)
d = response.json()
page = d["archived_snapshots"]["closest"]["url"]
webbrowser.open(page)
```

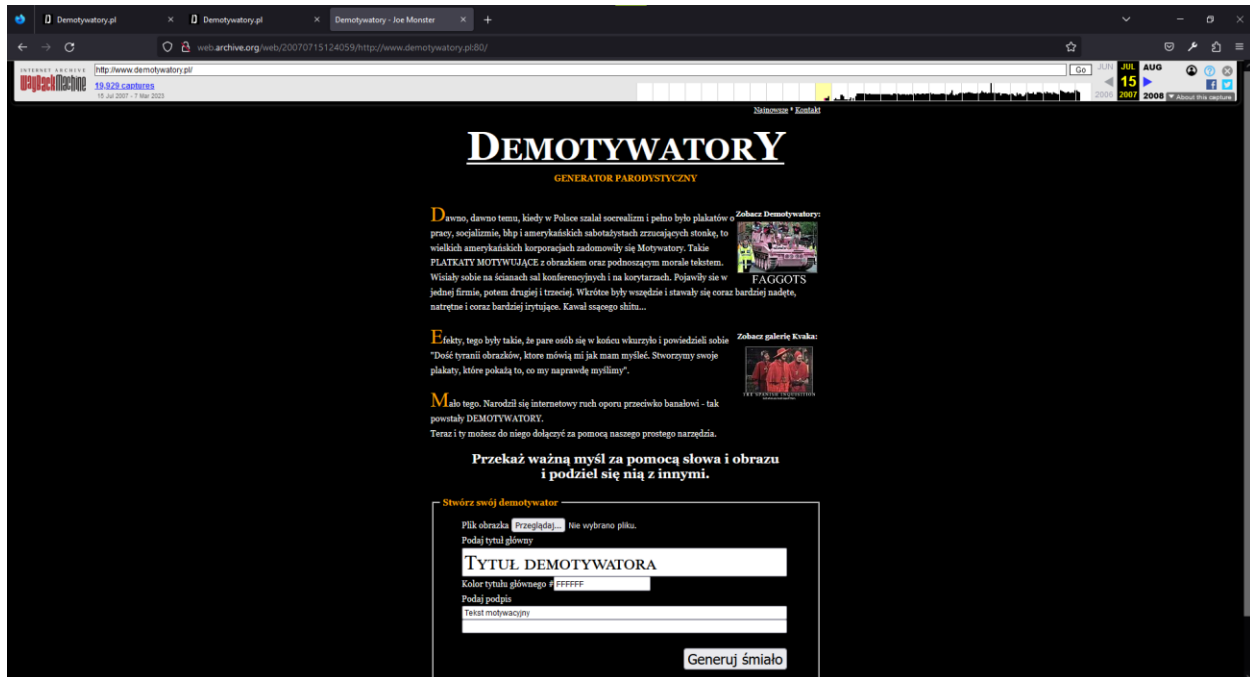
Zrzuty ekranu stron:

Strona nr 1



Strona nr 2





5.

W Pythonie mówimy o interpreterze, ponieważ jest on językiem interpretowanym. Oznacza to, że kod źródłowy jest tłumaczony linia po linii, a interpreter czyta i wykonuje kod bezpośrednio z kodu źródłowego. Interpreter Pythona wykonuje kod bezpośrednio na maszynie, bez potrzeby kompilacji kodu na pliki wykonywalne. Tymczasem Java jest językiem kompilowanym, co oznacza, że kod źródłowy jest najpierw kompilowany do kodu bajtowego, który jest następnie wykonywany na maszynie wirtualnej. JVM czyta i wykonuje kod bajtowy, a nie kod źródłowy. W skrócie, interpreter Pythona czyta i wykonuje kod źródłowy bezpośrednio, podczas gdy maszyna wirtualna Javy czyta i wykonuje kod bajtowy. Python jest wykonywany bezpośrednio na maszynie docelowej. Interpreter może przetwarzać kod szybciej niż kompilator, ale zwykle kod interpretowany działa wolniej niż kod skompilowany.