PyStart #40 Biblioteka standardowa

Co wchodzi w jej skład ?

Przetwarzanie tekstów

Data i czas

Obsługa plików

Tkinter

sqlite3

Kryptografia

Liczby i matematyka

Turtle

Logowanie

System

Sieć

Debugowanie / testowanie

PyStart #40 Biblioteka standardowa

Co wchodzi w jej skład ?

https://docs.python.org/3/library/

PyStart #41 Jak pracować z datą i czasem?

Unix Epoch - start!

01.01.1970 00:00:00

PyStart #41 Jak pracować z datą i czasem?

Jaki dzisiaj jest dzień?

```
1  from datetime import date
2
3  today = date.today()
4  print(f'Dziś jest: {today}')
5
```

PyStart #41 Jak pracować z datą i czasem?

Formatowanie obiektu datetime

```
1  from datetime import date
2
3  today = date.today()
4  formatted = today.strftime("%d.%m.%Y")
5  print(formatted)
6
```

PyStart #41 Jak pracować z datą i czasem?

Formatowanie obiektu datetime

Oznaczenie	Wyjaśnienie	Przykład
%a	skrócony dzień tygodnia	Mon, Tue, Wed
%A	pełny dzień tygodnia	Monday, Tuesday
%w	dzień tygodnia cyfrą	0-6
%d	dzień miesiąca z zerem	01-31
%-d	dzień miesiąca bez zera	1-31
%b	skrócony miesiąc	Jan...Dec
%B	pełen miesiąc	January - December
%m, %-m	cyfra miesiąca	01-12 lub 1-12 (jak dzień)

PyStart #41 Jak pracować z datą i czasem?

Formatowanie obiektu datetime

Więcej przykładów

https://docs.python.org/3/library/datetime.html#strftime-and-strptime-format-codes

PyStart #41 Jak pracować z datą i czasem?

Operacje na dacie i czasie

➔ Utworzenie obiektu daty i czasu

```
1  from datetime import date
2
3  today = date.today()
4  day = date(today.year, 11, 9)
5  formatted = day.strftime("%d.%m.%Y")
6  print(f'Moje urodziny w tym roku: {day}')
7
```

PyStart #41 Jak pracować z datą i czasem?

Operacje na dacie i czasie

➔ Porównywanie oraz Ilość dni między dwiema datami

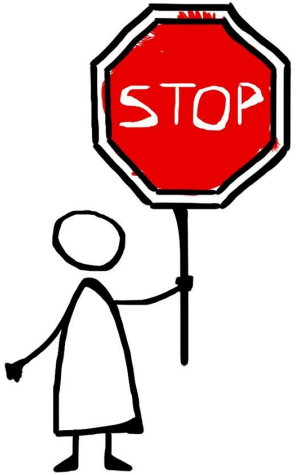
```
from datetime import date

today = date.today()
birthday = date(today.year, 11, 9)

if birthday > today:
    diff = birthday - today
    print(f'Do urodzin pozostało dni: {diff.days} ')
else:
    diff = birthday - today
    print(f'Urodziny były dni temu: {diff.days} ')

```

PyStart #41 Jak pracować z datą i czasem?

Zadania dla nabrania wprawy

Przygotuj program, który wyświetli datę Twoich następnych urodzin oraz odpowie na pytanie ile musisz na nie czekać dni.

Jeśli dzień Twoich urodzin w danym roku już minął powinna pokazać się data w roku następnym.



PyStart #42 Jak pracować z datą i czasem?

Timedelta - różnica

```
1  from datetime import date, timedelta
2
3  start = date.today()
4  diff = timedelta(days=7)
5  end = start + diff
6  print(end.strftime('%d.%m.%y'))
7
```

PyStart #42 Jak pracować z datą i czasem?

Datetime - Aktualny czas

```
1  from datetime import datetime
2
3  event = datetime.now()
4  print(event.hour)
5  print(event.minute)
6  print(event.strftime('%H:%M'))
7
```

PyStart #42 Jak pracować z datą i czasem?

Odebranie daty od użytkownika

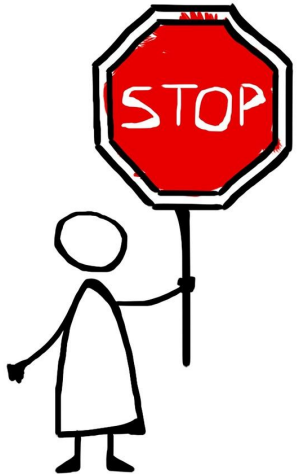
➔ Jeśli data będzie podawana w innym formacie program wyrzuci błąd. Na chwilę obecną nic z tym nie możemy zrobić.

```
1  from datetime import datetime
2  birthday = input('Podaj datę urodzenia dd.mm.rrrr: ')
3  d = datetime.strptime(birthday, '%d.%m.%Y')
4
```

PyStart #42 Jak pracować z datą i czasem?

Zadania dla nabrania wprawy

1. Poproś użytkownika o datę rozpoczęcia, datę zakończenia, a także o jego dniówkę. W odpowiedzi powinna wyświetlić się informacja ile użytkownik zarobi.
2. Spróbuj wyświetlić za pomocą pętli wszystkie dni pomiędzy dwiema datami z poprzedniego zadania.
3. Policz podwójnie wynagrodzenie pracownika za pracę w soboty i niedziele.



PyStart #43 Tkinter - środowisko graficzne

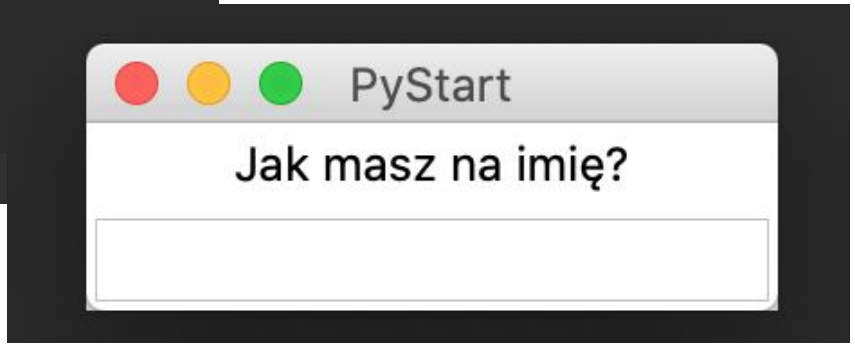
Pierwsze okienko

```
1  import tkinter as tk
2
3  window = tk.Tk()
4  window.title('PyStart')
5  window.mainloop()
6
```

PyStart #43 Tkinter - środowisko graficzne

Pole do wpisywania

```
1   import tkinter as tk
2
3   window = tk.Tk()
4   window.title('PyStart')
5
6   label = tk.Label(window, text="Jak masz na imię?")
7   label.pack()
8
9   first_name = tk.Entry()
10  first_name.pack()
11
12  window.mainloop()
13
```



PyStart #43 Tkinter - środowisko graficzne

Guzik

```
4   def send_name():
5       print(first_name.get())
6       print('click')
7
8
9   window = tk.Tk()
10  window.title('PyStart')
11
12  label = tk.Label(window, text="Jak masz na imię?")
13  label.pack()
14
15  first_name = tk.Entry()
16  first_name.pack()
17
18  button = tk.Button(text="OK", command=send_name)
19  button.pack()
```

PyStart #43 Tkinter - środowisko graficzne

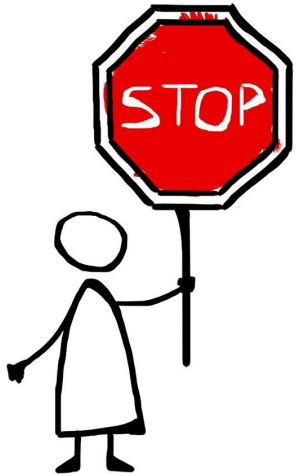
Obsługa i dodatkowe okienko

```
4
5   def send_name():
6       if first_name.get() == 'Kacper':
7           messagebox.showinfo(title='OK!', message='Witam Pana!')
8       else:
9           messagebox.showerror(title='Kurza twarz', message='Nie znam Cię..')
10
```

PyStart #43 Tkinter - środowisko graficzne

Zadania dla nabrania wprawy

1. Przygotuj grę w “Zgadnij liczbę” z interfejsem graficznym:
    a. Program losuje liczbę od 1 - 50
    b. Jeśli użytkownik będzie podawał wartości coraz bliższe zgadywanej liczbie program powinien wyświetlić “ciepło”
    c. Jeśli użytkownik będzie podawał wartości coraz dalsze powinien wyświetlić “zimno”.
2. Po wpisaniu poprawnej liczby program powinien wyświetlić stosowny komunikat i odpowiedzieć w ilu krokach użytkownik zgadł.



PyStart #44 Obsługa pliku testowego

Otwarcie pliku, odczytanie linii

```
2    handler = open('test.txt')
3    line = handler.readline()
4    print(line)
5    handler.close()
```

PyStart #45 Obsługa pliku testowego

Otwarcie pliku, odczytanie linii

→ Korzystając z tzw. context managera

```
2  with open('test.txt') as handler:
3      line = handler.readline()
4      print(line)
5
```

PyStart #44 Obsługa pliku testowego

Porównanie

```
2    handler = open('test.txt')
3    line = handler.readline()
4    print(line)
5    handler.close()
```

```
2    with open('test.txt') as handler:
3        line = handler.readline()
4        print(line)
5
```

PyStart #44 Obsługa pliku testowego

Otwarcie pliku

builtins.py

```
353 *  def open(file, mode='r', buffering=None, encoding=None,
```

Tryb	Opis
r	tylko do odczytu
r+	odczyt i zapis, wskaźnik na początku
w	tylko zapis, **plik jest czyszczony**
w+	zapis i odczyt, **plik jest czyszczony**
a	zapis, wskaźnik na końcu

PyStart #44 Obsługa pliku testowego

Odczyt całego pliku

```
1
2   with open('test.txt', 'r') as handler:
3       for line in handler:
4           print(line.strip())
5
```

PyStart #44 Obsługa pliku testowego

Odczyt całego pliku

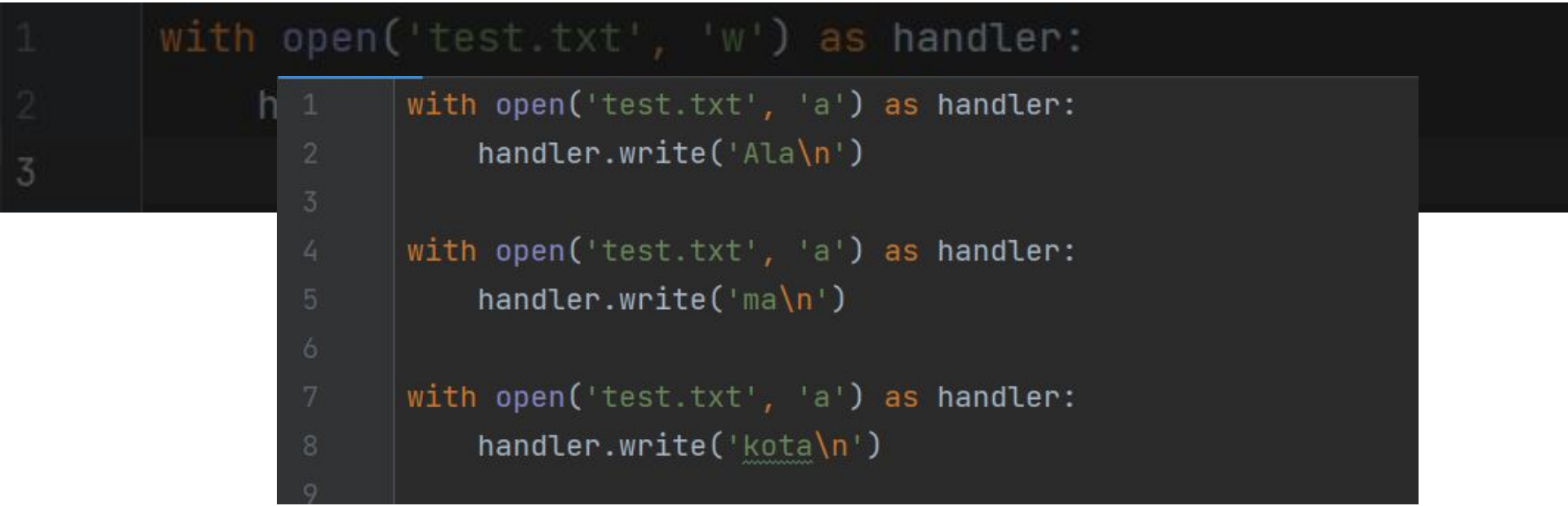
PyStart #44 Obsługa pliku testowego

Zapis do pliku

```
1  with open('test.txt', 'w') as handler:
2      handler.write('Ala ma kota\n')
3
```

PyStart #44 Obsługa pliku testowego

Zapis do pliku

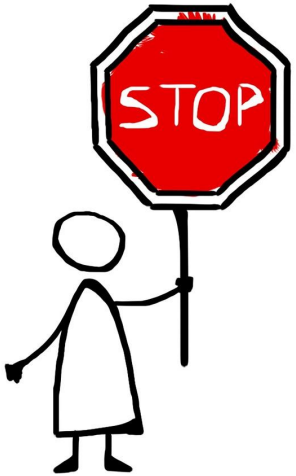


PyStart #44 Obsługa pliku tekstowego

Zadania dla nabrania wprawy

1. Plik csv to plik w którym kolumny rozdzielone są średnikami.
   Możesz je odczytać za pomocą funkcji split.
   Z pliku transakcje.txt wybierz tylko te, które mają dodatnią wartość i zapisz je do osobnego pliku o nazwie przychody.txt

2. Napisz program, który otworzy plik przychody.txt, odczyta wartości zapisane w kolejnych wierszach i wyświetli ich sumę.



PyStart #45 JSON

Jak to i po co?

data.json

```
{
  "id": 1,
  "name": "PyStart",
  "members": [
    {
      "id": 1,
      "first_name": "Jan",
      "last_name": "Kowalski"
    }, {
      "id": 2,
      "first_name": "John",
      "last_name": "Smith"
    }
  ]
}
```

- ➔ Struktura bardzo podobna do list i słowników
- ➔ “Klucze” muszą być w cudzysłowie
- ➔ Format służący do przenoszenia danych pomiędzy systemami np. poprzez API.

PyStart #45 JSON

Kilka przydatnych funkcji

```
1   from json import load, loads, dump, dumps
```

“s” na końcu oznacza string! bez “s” oznacza plik

- ➔ load — plik .json => listy/słowniki
- ➔ loads — string json => listy/słowniki
- ➔ dump — listy/słowniki => zapisuje plik.json
- ➔ dumps — lista/słowniki => zapisuje do stringa

PyStart #45 JSON

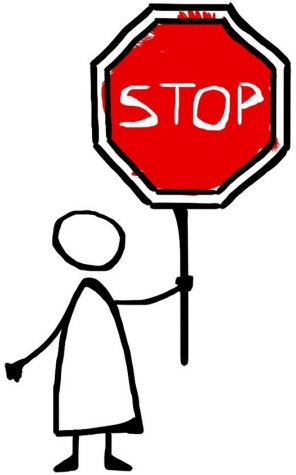
Zadania dla nabrania wprawy

Napisz program zarządzający wydatkami w Twoim domu.

Program powinien umożliwić dodanie nowego wydatku.

Przechowuj je w pliku wydatki.json

```
(msvenv) D:\Projects\pystart_code>python wydatki.py
Co chcesz zrobić? [d] Dodaj wydatek [w] Wypisz wszystkie: d
Czego dotyczy wydatek? Opłaty
Jaka kwota została wydana? 1000
```



PyStart #46 Dane zewnętrzne

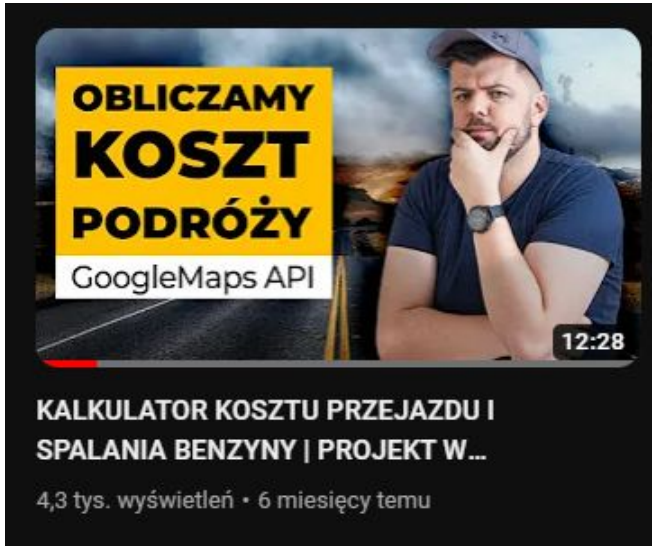
Czym jest API?

- API - Application Programming interface
  - Pobieranie publicznie dostępnych informacji z publicznych API np. kursy walut, pogodę, dług publiczny
  - Komunikacja z zewnętrznymi usługami

    np. płatności internetowe, pobieranie trasy, wysyłka maila z serwisu mailingowego
  - Własne usługi, wewnętrzne API

PyStart #46 Dane zewnętrzne

Przykład



PyStart #46 Dane zewnętrzne

Przykład

```
url = "https://maps.googleapis.com/maps/api/distancematrix/json"

payload = {
    'origins': 'Warszawa, Plac Zbawiciela 3',
    'destinations': 'Warszawa, Emilii Platter 13',
    'key': api_key
}
```

```
(msvenv) D:\Projects\trip_cost_calc>python main.py
{'destination_addresses': ['Emilii Plater 13, 00-699 Warszawa, Poland'], 'origin_addresses': ['plac Zbawiciela 3, 00-642 Warszawa, Poland'],
': [{'elements': [{'distance': {'text': '1.5 km', 'value': 1536}, 'duration': {'text': '6 mins', 'value': 357}, 'status': 'OK'}]}], 'status':
}
```

PyStart #46 Dane zewnętrzne

Czas na demonstrację!

Udostępnione API z walutami:

http://api.nbp.pl/api/exchangerates/tables/A/?format=json

Zainstaluj moduł requests

pip install requests

PyStart #46 Dane zewnętrzne

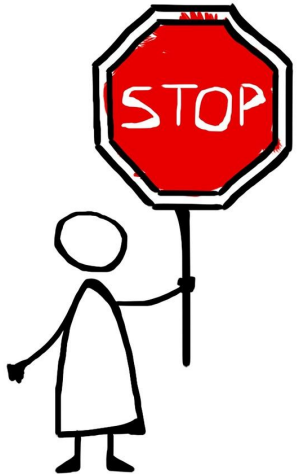
Zadania dla nabrania wprawy

Na podstawie danych udostępnionych bezpłatnie przez:

https://danepubliczne.imgw.pl/apiinfo

Przygotuj skrypt, który wyświetli informację o temperaturze oraz ciśnieniu w mieście najbliższym miejscu Twojego zamieszkania.


```
[
  {
    "id_stacji": "12295",
    "stacja": "Białystok",
    "data_pomiaru": "2021-04-20",
    "godzina_pomiaru": "20",
    "temperatura": "8.9",
    "predkosc_wiatru": "1",
    "kierunek_wiatru": "230",
    "wilgotnosc_wzgledna": "68.6",
    "suma_opadu": "0",
    "cisnienie": "1012"
  },
```

SIÓDMA PRACA DOMOWA

UWAGA! UWAGA!

- Napisz program, który wyświetli bieżącą datę i godzinę w formacie "dd/mm/rrrr, godzina:minuta:sekunda".
- Napisz program, który wyświetli ile dni zostało do kolejnego lata (21 czerwca). Weź pod uwagę sytuację gdy data jest przed nami oraz za nami.
- Napisz program, który pyta użytkownika o jego imię i nazwisko oraz zapisuje te informacje w pliku tekstowym.
- Stwórz interfejs graficzny do powyższego programu. Jeśli imię i nazwisko występowało już wcześniej w pliku to wyświetl komunikat informujący o tym i nie dodawaj duplikatu.

SIÓDMA PRACA DOMOWA

UWAGA! UWAGA!

- W pliku `bałagan.txt` jest wiele linii zawierających wyrażenie “Java” oraz “Java Script”. Wszystkie wyrażenia Java wymień na Python, natomiast linie zawierające Java Script pozostaw niezmienione.
- Odczytaj zawartość pliku `logi.txt` i policz ile czasu poszczególni użytkownicy spędzili w aplikacji. Plik załączony pod filmem.
- Korzystając z API dostarczonego przez serwis wolnelektury pobierz wszystkie dzieła konkretnego autora. Imię i nazwisko autora powinno zostać przekazane jako odpowiedź na pytanie użytkownika. Zapisz wynik do pliku o nazwie: `imie-nazwisko-autora.json`