



# ПРАКТИКА 7

Простой скрипт на Python



# Цель задания

- Получить базовые навыки работы с Python
- Научиться читать файлы в Python
- Научиться работать с базой данных в Python

# Задание

- На портале открытых данных выбрать и скачать файл, данные которого будут записаны в БД. Формат файла: csv, xml, json (один на выбор)

Портал открытых данных: <https://data.gov.ru/opendata>

- Определить структуру БД для этих данных (минимум 5 полей), создать таблицу (таблицы при необходимости)
- Написать скрипт для чтения данных из файла и записи их в БД
- Написать скрипт, получающий данные из БД и проводит с ними некую работу (вывод на экран, подсчёт чего-либо и т.д.)

# Что нужно сдать

- Ссылка на набор данных (или сам файл с данными)
- Скрипт для создания структуры таблицы
- Скрипт чтения данных
- Скрипт работы с данными

# Установка Python и IDE

- Python ставится с официального сайта

<https://www.python.org/downloads/>

Ставить можно любую (но обязательно Python 3, Python 2 уже не поддерживается)

- После установки проверить, установился ли pip

`pip --version`

Если отображается версия, то всё отлично, можно продолжать работу.

- Установить PyCharm Community Edition для упрощения работы

<https://www.jetbrains.com/ru-ru/pycharm/>

# Структура программы

- Код в главном файле, отступ - 4 пробела:

Вариант 1:

```
if __name__ == "__main__":  
    do_smth()
```

Вариант 2 (без отступов):

```
do_smth()
```

- Комментарии пишутся после #

Длинные комментарии можно писать как

""" тут комментарий с переносами """

# Нет явного указания типов

- `my_int = 5 # int`
- `my_str = "some string" # str`
- `is_valid = False # boolean`

# Объявление функций

```
def my_func():
```

```
    pass
```

- `pass` - команда, которая делает ничего, но функцию нельзя оставить пустой, поэтому пишется `pass`
- В скобках - параметры функции

```
def youtube_func(url):
```

```
    return -1
```



# print

- Выводит данные

```
print("hello world")
```

- Конкатенация строк при выводе

```
my_var = "something"
```

```
print("I want to print: " + my_var)
```

- f-строки

```
my_var = "something"
```

```
print(f"I want to print: {my_var}")
```

# Импорт функций из соседнего файла - import

- Дано:

- *main.py* - где содержится главная функция
- *youtube.py* - содержится функция *youtube\_func*, которую хочется вызвать в главном модуле

- В *main.py* нужно написать

## Вариант 1

```
from youtube import youtube_func
```

Тогда можно вызывать напрямую *youtube\_func()*

## Вариант 2

```
import youtube
```

Но вызов идёт через указание модуля, в котором содержится функция  
*youtube.youtube\_func()*

# Чтение данных из файла

# Вариант 1

```
with open(file_name) as fp:
```

```
    all_data = fp.read()
```

```
    print(all_data)
```

# Вариант 2

```
fp = open(file_name)
```

```
all_data = fp.read()
```

```
print(all_data)
```

# Чтение файла построчно

```
fp = open("my_file")
```

```
for line in fp:
```

```
    line = line.strip()
```

```
    my_array = line.split(" ")
```

```
    print(my_array)
```

line - строка из файла

.strip() - удаление конечных пробелов

.split() - разбить строку по разделителям, возвращает list данных

# JSON

- Встроенный модуль json

```
import json
```

```
# вариант 1
```

```
fp = open(file_name)
```

```
json_file = json.load(fp)
```

```
print(json_file)
```

```
# вариант 2
```

```
fp = open(file_name)
```

```
all_data = fp.read()
```

```
json_file = json.loads(all_data)
```

```
print(json_file)
```

# Работа с Базой Данных

- Модуль psycopg2
- `pip install psycopg2-binary`
- Хороший мануал
- <https://habr.com/ru/post/321510/>