

# Лабораторная работа №2

## Введение в Python

### Задание и порядок выполнения

Основная задача данной работы - знакомство с базовым синтаксисом Python.  
Дополнительная задача - знакомство с git и github. Git вам понадобится для выполнения и сдачи домашнего задания.

1. Создайте новый проект в PyCharm
  - a. в поле Location **untitled** заменить, например, на **lab2**
  - b. в поле Interpreter должен быть Python 3.5.x
2. Массивы
  - a. Добавьте в проект новый **Python File** с именем **arr\_algs.py**
  - b. Реализуйте в нём следующие функции:
    - i. Нахождение минимума в массиве
    - ii. Нахождение среднего арифметического в массиве
  - c. Проверьте правильность работы ваших функций, вызвав их несколько раз в том же модуле в конце файла
3. Строки
  - a. Добавьте в проект новый **Python File** с именем **str\_algs.py**
  - b. Реализуйте в нём следующие функции:
    - i. Переворот строки ("hello, world" -> "dlrow ,olleh")
  - c. Проверьте правильность работы ваших функций, вызвав их несколько раз в том же модуле в конце файла
4. Словари
  - a. Добавьте в проект новый **Python File** с именем **dict\_algs.py**
  - b. Реализуйте в нём следующий алгоритм:
    - i. есть несколько сотрудников, описанных в виде массива словарей `emps` (данные приведены ниже в конце этого раздела)
    - ii. выведите имена тех сотрудников, у которых есть дети старше 18 лет
  - c. Проверьте правильность работы вашего алгоритма, вызвав его в том же модуле в конце файла
5. Github
  - a. Зайдите на <https://github.com>
  - b. Создайте репозиторий с названием **lab2\_repo**
  - c. Склонируйте его себе на диск
  - d. Добавьте туда ваши .py файлы
  - e. commit, push
  - f. подробнее - см. в разделе Теория.Git

Данные для выполнения задания со словарями:

```
ivan = {  
    "name": "ivan",  
    "age": 34,  
    "children": [{  
        "name": "vasja",  
        "age": 12,  
    }, {  
        "name": "petja",  
        "age": 10,  
    }],  
}
```

```
darja = {  
    "name": "darja",  
    "age": 41,  
    "children": [{  
        "name": "kirill",  
        "age": 21,  
    }, {  
        "name": "pavel",  
        "age": 15,  
    }],  
}
```

```
emps = [ivan, darja]
```

## Теория и примеры

### Python

#### Теория

Python - высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Код в Python организовывается в функции и классы, которые могут объединяться в модули (они в свою очередь могут быть объединены в пакеты).

Python поддерживает динамическую типизацию, то есть тип переменной определяется только во время исполнения. Поэтому вместо «присваивания значения переменной» лучше говорить о «связывании значения с некоторым именем». В Python имеются встроенные типы: булевый, строка, Unicode-строка, целое число произвольной точности, число с плавающей запятой, комплексное число и некоторые другие. Из коллекций в Python встроены:

1. список (list),

2. кортеж (неизменяемый список, tuple),
3. словарь (dict),
4. множество (set),
5. и другие.

Все значения являются объектами, в том числе функции, методы, модули, классы.

## Пример синтаксиса

Функция `filter_tweets` принимает 2 аргумента: массив твитов и пороговое значение числа подписчиков пользователя. Если у автора текущего твита достаточно подписчиков, текущий твит пройдет фильтрацию и будет выведен на экран.

```
def filter_tweets(tweets, followers_count_limit):
    filtered = []
    for tweet in tweets:
        if tweet['user']['followers_count'] >= followers_count_limit:
            filtered.append(tweet)

    return filtered

if __name__ == '__main__':
    # test data
    all_tweets = [{
        'user': {
            'name': 'vasja',
            'followers_count': 1189,
        },
        'text': 'hello, world'
    }, {
        'user': {
            'name': 'petja',
            'followers_count': 20,
        },
        'text': 'some message'
    }]

    # call our function
    filtered_tweets = filter_tweets(all_tweets, 100)
    print('Filtered tweets: ')
    print(filtered_tweets)
```

Проверка `if __name__ == '__main__':`

служит для того, выполняется только в том случае, если вы запустили непосредственно этот скрипт на выполнение, а не импортировали его для использования в другом модуле.

## Git

Git - распределённая система управления версиями.

Что позволяет git:

1. Не потерять код вашего проекта при неисправности диска
2. Иметь доступ к вашему коду в любой точке мира
3. Работать над одним проектом командой из нескольких человек
4. Работать одновременно над несколькими версиями продукта
5. Иметь возможность откатить изменения на любое состояние в прошлом, даже без подключения к интернету

Тьюториал по использованию git можно посмотреть, например, здесь:

<https://githowto.com/ru>

Github - это популярный публичный хостинг git-репозитория.

Git представляет собой набор консольных утилит, доступ к которым осуществляется через консольную команду **git**.

Настройка git:

1. Откройте консоль git'a в папке с проектами для этого:
  - a. Откройте папку с проектами на диске в проводнике например, C:\Projects
  - b. Откройте в ней консоль  
ПКМ -> Git Bash Here
2. Выполните

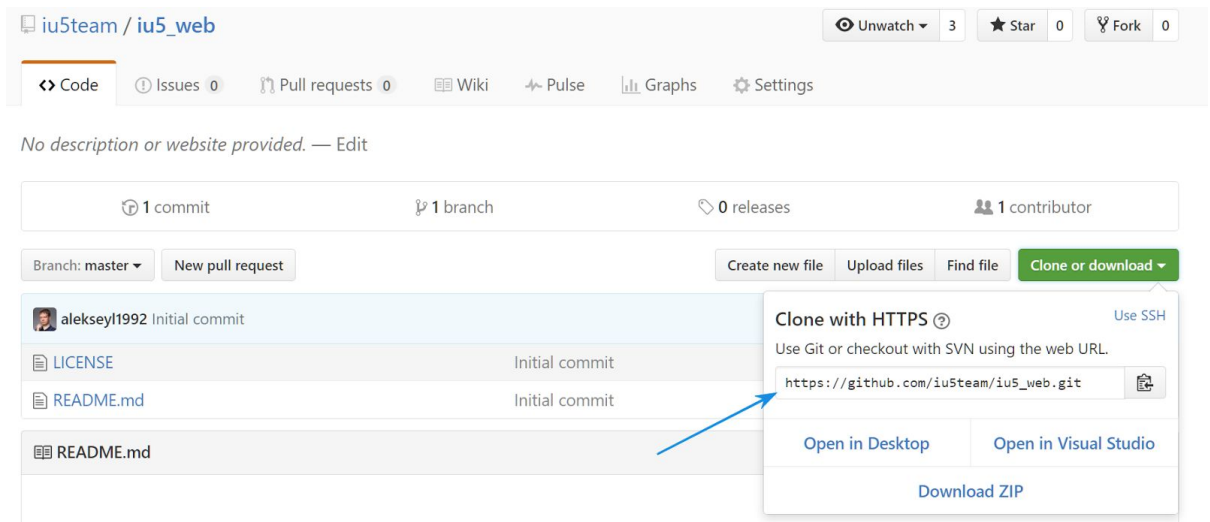
```
git config --global user.name "username"
git config --global user.email "your\_email@whatever.com"
```

(используйте логин и почту, указанные при регистрации на github)

Давайте рассмотрим, как выгрузить ваш код на сервера github.

1. Перейдите на страницу вашего репозитория в браузере

2. Скопируйте ссылку на ваш репозиторий:



3. Откройте папку с проектами на диске  
например, C:\Projects
4. Откройте в ней консоль  
ПКМ -> Git Bash Here
5. Выполните команду  
**git clone [https://github.com/iu5team/iu5\\_web.git](https://github.com/iu5team/iu5_web.git)**  
(ссылку замените на свою)
6. Теперь перенесите .py файлы вашего проекта в локальную копию репозитория  
(из папки lab2 в lab2\_repo)
7. Перейдите в клонированную папку  
**cd lab2\_repo**
8. Выполните команду  
**git add .**  
это добавит все новые файлы в индекс  
т.е. git будет теперь за ними следить  
точка тут означает текущую папку
9. Для фиксации изменений (эдакая точка отката) выполните  
**git commit -am "Added lab2 scripts"**  
(сообщение в кавычках произвольное, это описание того, что вы сделали в этом коммите)
10. Теперь отправим изменения на сервер github  
**git push origin master**
11. Введите пароль
12. Убедитесь, что файлы добавились, открыв страницу вашего репозитория в браузере

## Вспомогательные материалы

1. <https://ru.wikipedia.org/wiki/Python>
2. Справочник по Python: <https://docs.python.org/3/tutorial/index.html>
3. Тьюриал по Git: <https://githowto.com/ru>

4. Справочник по Git: <https://git-scm.com/doc>

## Контрольные вопросы

1. Отличия Python от C++ и C#
2. Типы данных в Python
3. Операторы Python (циклы, условия)
4. Назначение git
5. Основные команды git