

**Московский государственный технический университет имени Н.Э. Баумана**

**Кафедра ИУ5  
«Системы обработки информации и управления»**

**Отчет по лабораторной работе №2**

**«Введение в Python»**

**по дисциплине «Разработка Интернет-приложений»**

**Выполнил:  
студент группы ИУ5-53  
Слимов Никита**

**Москва, 2016**

## Задание

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.

## Работа с массивами

Нахождение минимума в массиве

Нахождение среднего арифметического в массиве

```
def my_min(arr):
    if len(arr) == 0:
        return None

    m = arr[0]

    for i in arr:
        if i < m:
            m = i

    return m

def my_avg(arr):
    if len(arr) == 0:
        return None

    s = 0

    for i in arr:
        s += i

    return s / len(arr)

if __name__ == '__main__':
    assert my_min([]) is None
    assert my_min([6, -1, -10, 5, 3, 7, 9]) == -10

    assert my_avg([]) is None
    assert my_avg([5, 0, 3, 1, 1]) == 2
```

## Строки

Переворот строки ("hello, world" -> "dlrow ,olleh")

```
def my_reverse(str):
    return ''.join(reversed(list(str)))

if __name__ == '__main__':
    assert my_reverse("hello, world") == "dlrow ,olleh"
    assert my_reverse("арозаупаланалапуазора") == "арозаупаланалапуазора"
```

## Словари

Имена тех сотрудников, у которых есть дети старше 18 лет

```
ivan = {
    "name": "ivan",
    "age": 34,
    "children": [
        {
            "name": "vasja",
            "age": 10,
        },
        {
            "age": 12,
            "name": "petja",
        }
    ],
}
darja = {
    "name": "darja",
    "age": 41,
    "children": [
        {
            "age": 15,
            "name": "kirill",
        },
        {
            "age": 21,
            "name": "pavel",
        }
    ],
}
emps = [ivan, darja]
```

```
def my_filter(employyes):
    filtered = []

    for emp in employyes:
        for child in emp['children']:
            if child['age'] > 18:
                filtered.append(emp)
                break

    return filtered

if __name__ == '__main__':
    filtered = my_filter(emps)
    print(list(map(lambda x: x.get('name'), filtered)))
    assert darja in filtered
    assert ivan not in filtered
```

---

```
/usr/local/bin/python3 /Users/nikita/www/python_labs/lab2/dict_algs.py
['darja']
```