

# Использование Python для решения типовых DevOps задач



Алексей  
Метляков



**Алексей Метляков**

DevOps Engineer, OpenWay



---

# План занятия

1. [Для чего нужен Python?](#)
2. [Основы синтаксиса Python](#)
3. [Модули для работы с системой](#)
4. [Как написать первый скрипт?](#)
5. [Итоги](#)
6. [Домашнее задание](#)



# Для чего нужен Python?

---

# Для чего нужен Python, когда есть Bash?

Возможности **Python** шире, так как:

- Он является языком программирования
- Содержит большое количество встроенных функций и модулей для работы с системой, без ущерба работы с логикой
- Имеет множество загружаемых модулей для более удобной работы с любым уровнем автоматизаций

---

# Для чего нужен Python в DevOps?

В DevOps у Python очень разносторонняя судьба:

- **Автоматизация конфигурирования инфраструктуры** при помощи Ansible
- **Использование в инструментах автоматизации.** Например, в Jenkins можно описывать шаги сборки полностью на Python
- **Удобство работы с API инструментов.** Существует множество готовых решений для работы с API Bitbucket, GitLab, GitHub, Nexus, Crowd, Jira, Confluence, etc.



# Основы синтаксиса Python

# Переменные

В Python **используется неявное определение** переменных:

- Определение типа переменной происходит динамически
- Любой тип переменной можно преобразовать в любой другой тип: `int(s)`, `str(i)`, `float(i)`
- Следить за типом переменных и контролировать их переопределение - наша задача
- Определить тип переменной можно при помощи `type(имя_переменной)`



# Массивы

**Массивы в Python** - не такие, как в привычном понимании программиста.

**Существует несколько видов:**

- Упорядоченный, редактируемый (list) ['a', 23, 'hello']
- Упорядоченный, не редактируемый (tuple) (14, 'yes', 'no')
- Неупорядоченный, редактируемый, уникальный (set) {H,e,l,o}
- Неупорядоченный, key-value (dict) {1:'Январь', 2:'Февраль'}

---

# Операции

## Python поддерживает:

- весь набор арифметических операций
- весь набор логических операций
- конкатенацию строк, сравнение с эталоном, поиск подстроки

# Оператор условия

Конструкция if-elif-else в Python имеет следующий синтаксис:

```
if (условие):  
    список действий  
elif (условие):  
    список действий  
else:  
    список действий
```

# Циклы

Существует два вида конструкций циклов:

```
sample=[0,1,2,3,4,5,6]
for (i in sample):
    print(str(i))
```

Выводим построчно  
значения элементов  
массива

```
a = 0
while (a < 5):
    print(a)
    a += 1
```

Выводим значение  
переменной



# Модули для работы с системой

# Модуль sys

Модуль обеспечивает доступ к некоторым функциям и переменным, которые взаимодействуют с интерпретатором:

**sys.args** #Возвращает список параметров, переданных скрипту

**sys.exit()** #Возбуждает исключение SystemExit и завершает работу

**sys.platform** #Возвращает наименование ОС

**sys.getsizeof()** #Возвращает размер объекта в байтах

# Модуль os

Модуль позволяет взаимодействовать с ОС при помощи разнообразных функций:

```
os.getlogin() #Возвращает логин текущего пользователя
os.getuid() #Возвращает id текущего пользователя
os.uname() #Возвращает информацию о системе
os.access(path, flag) #Проверяет доступность файла
os.getcwd() #Возвращает текущий каталог
os.popen() #Выполняет системную команду и позволяет
записать вывод в переменную
os.makedirs(path) #Создаёт директорию по всему пути
os.truncate(path, length) #Обрезает файл до указанной длины
os.walk() #Рекурсивно собирает информацию о файлах
```



# Как написать первый скрипт?



# Как написать первый скрипт

Существует ряд основных **правил для формирования** правильного python-скрипта:

- Скрипт может начинаться с `#!/usr/bin/env python3`
- Стараться писать в соответствии с PEP8

---

# Итоги

Сегодня мы:

- Узнали для каких DevOps-задач можно использовать Python
- Вспомнили основы синтаксиса
- Познакомились с двумя модулями для работы с системой
- Узнали как написать свой первый скрипт



## Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задавайте **в чате** мессенджера Slack.
- Задачи можно сдавать **по частям**.
- Зачёт по домашней работе проставляется после того, как **приняты все задачи**.

**Задавайте вопросы и  
пишите отзыв о лекции!**

**Алексей Метляков**