PYTHON ДЛЯ СЕТЕВЫХ ИНЖЕНЕРОВ

СОЗДАНИЕ БАЗОВЫХ СКРИПТОВ

© 2017 Наташа Самойленко

СОЗДАНИЕ БАЗОВЫХ СКРИПТОВ

Файл access_template.py:

```
access_template = ['switchport mode access',
                   'switchport access vlan {}',
                   'switchport nonegotiate',
                   'spanning-tree portfast',
                   'spanning-tree bpduguard enable']

print('\n'.join(access_template).format(5))
```

© 2017 Наташа Самойленко

СОЗДАНИЕ БАЗОВЫХ СКРИПТОВ

Так выглядит выполнение скрипта:

```
$ python access_template.py
switchport mode access
switchport access vlan 5
switchport nonegotiate
spanning-tree portfast
spanning-tree bpduguard enable
```

© 2017 Наташа Самойленко

ИСПОЛНЯЕМЫЙ ФАЙЛ

Для того, чтобы файл был исполняемым, и не нужно было каждый раз писать python перед вызовом файла, нужно:

- сделать файл исполняемым (для linux)
- в первой строке файла должна находиться строка `#!/usr/bin/env python` или `#!/usr/bin/env python3`, в зависимости от того, какая версия Python используется по умолчанию

© 2017 Наташа Самойленко

ИСПОЛНЯЕМЫЙ ФАЙЛ

Пример файла access_template_exec.py:

```
#!/usr/bin/env python3

access_template = ['switchport mode access',
                   'switchport access vlan {}',
                   'switchport nonegotiate',
                   'spanning-tree portfast',
                   'spanning-tree bpduguard enable']

print('\n'.join(access_template).format(5))
```

© 2017 Наташа Самойленко

ИСПОЛНЯЕМЫЙ ФАЙЛ

После этого:

```
chmod +x access_template_exec.py
```

Теперь можно вызывать файл таким образом:

```
$ ./access_template_exec.py
```

© 2017 Наташа Самойленко

ПЕРЕДАЧА АРГУМЕНТОВ СКРИПТУ (ARGV)

© 2017 Наташа Самойленко

ПЕРЕДАЧА АРГУМЕНТОВ СКРИПТУ

В модуле sys есть очень простой и удобный способ для работы с аргументами - argv.

Файл access_template_argv.py:

```
from sys import argv

interface, vlan = argv[1:]

access_template = ['switchport mode access',
                   'switchport access vlan {}',
                   'switchport nonegotiate',
                   'spanning-tree portfast',
                   'spanning-tree bpduguard enable']

print('interface {}'.format(interface))
print('\n'.join(access_template).format(vlan))
```

© 2017 Наташа Самойленко

ПЕРЕДАЧА АРГУМЕНТОВ СКРИПТУ

```
$ python access_template_argv.py Gi0/7 4
interface Gi0/7
switchport mode access
switchport access vlan 4
switchport nonegotiate
spanning-tree portfast
spanning-tree bpduguard enable
```

© 2017 Наташа Самойленко

ПЕРЕДАЧА АРГУМЕНТОВ СКРИПТУ

Аргументы, которые были переданы скрипту, подставляются как значения в шаблон.

- argv - это список
- все аргументы находятся в списке в виде строк
- argv содержит не только аргументы, которые передали скрипту, но и название самого скрипта

В данном случае в списке argv находятся такие элементы:

```
['access_template_argv.py', 'Gi0/7', '4']
```

Сначала идет имя самого скрипта, затем аргументы, в том же порядке.

© 2017 Наташа Самойленко

РАСПАКОВКА

В Python есть возможность за раз присвоить значения нескольким переменным:

```
In [16]: a = 5
In [17]: b = 6
In [18]: c, d = 5, 6
In [19]: c
Out[19]: 5

In [20]: d
Out[20]: 6
```

© 2017 Наташа Самойленко

РАСПАКОВКА

Если вместо чисел список, как в случае с argv:

```
In [21]: arg = ['Gi0/7', '4']
In [22]: interface, vlan = arg

In [23]: interface
Out[23]: 'Gi0/7'

In [24]: vlan
Out[24]: '4'
```

© 2017 Наташа Самойленко

ВВОД ИНФОРМАЦИИ ПОЛЬЗОВАТЕЛЕМ

© 2017 Наташа Самойленко

ВВОД ИНФОРМАЦИИ ПОЛЬЗОВАТЕЛЕМ

Для получения информации от пользователя используется функция `input()`:

```
In [1]: print(input('Твой любимый протокол маршрутизации? '))
Твой любимый протокол маршрутизации? OSPF
OSPF
```

© 2017 Наташа Самойленко

ВВОД ИНФОРМАЦИИ ПОЛЬЗОВАТЕЛЕМ

В данном случае информация просто тут же выводится пользователю, но, кроме этого, информация, которую ввел пользователь, может быть сохранена в какую-то переменную и может использоваться далее в скрипте.

```
In [2]: protocol = input('Твой любимый протокол маршрутизации? ')
Твой любимый протокол маршрутизации? OSPF

In [3]: print(protocol)
OSPF
```

© 2017 Наташа Самойленко

ВВОД ИНФОРМАЦИИ ПОЛЬЗОВАТЕЛЕМ

В скобках обычно пишется какой-то вопрос, который уточняет, какую информацию нужно ввести.

Текст в скобках, в принципе, писать не обязательно.
И можно сделать такой же вывод с помощью функции **print**:

```
In [4]: print('Твой любимый протокол маршрутизации?')
Твой любимый протокол маршрутизации?

In [5]: protocol = input()
OSPF

In [6]: print(protocol)
OSPF
```

© 2017 Наташа Самойленко

ВВОД ИНФОРМАЦИИ ПОЛЬЗОВАТЕЛЕМ

```
interface = input('Enter interface type and number: ')
vlan = input('Enter VLAN number: ')

access_template = ['switchport mode access',
                   'switchport access vlan {}',
                   'switchport nonegotiate',
                   'spanning-tree portfast',
                   'spanning-tree bpduguard enable']

print('\n' + '-' * 30)
print('interface {}'.format(interface))
print('\n'.join(access_template).format(vlan))
```

© 2017 Наташа Самойленко

ВВОД ИНФОРМАЦИИ ПОЛЬЗОВАТЕЛЕМ

```
$ python access_template_input.py
Enter interface type and number: Gi0/3
Enter VLAN number: 55

-----------------------------
interface Gi0/3
switchport mode access
switchport access vlan 55
switchport nonegotiate
spanning-tree portfast
spanning-tree bpduguard enable
```

© 2017 Наташа Самойленко