



# Возможности интерпретатора Python в NX-OS

Anton Tugai  
Customer Support Engineer, Cisco TAC

October, 2014

# Cisco Support Community – Expert Series Webcast

Сегодня на семинаре Эксперт Cisco TAC **Антон Тугай** расскажет о возможностях интерпретатора Python в NX-OS



Антон Тугай  
Инженер центра  
технической поддержки  
Cisco TAC в Брюсселе

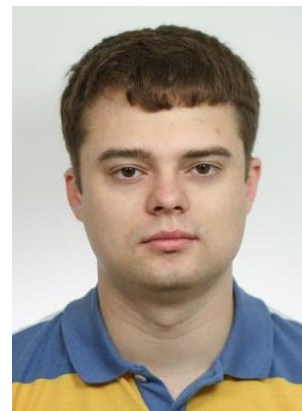
# Тема: Возможности интерпретатора Python в NX-OS

Дата проведения вебинара: 21 октября 2014 года

## Технические Эксперты



Александр Нестеров  
Инженер центра  
технической поддержки  
Cisco TAC в Брюсселе



Борис Берлог  
Инженер центра  
технической поддержки  
Cisco TAC в Брюсселе

# Спасибо, что посетили наш вебинар сегодня

Сегодняшняя презентация включает опросы аудитории

Пожалуйста, участвуйте!



# Спасибо, что присоединились к нам сегодня

Скачать презентацию Вы можете по ссылке:

<https://supportforums.cisco.com/ru/document/12330196>



# Присылайте Ваши вопросы!

Используйте панель Q&A, чтобы задать вопрос.  
Наши эксперты ответят на них.

## Вопрос №1

# Используете ли вы в своей работе/хобби Python ?

- a) **Не использую**
- b) **Хочу попробовать, но не знаю куда его можно применить**
- c) **Вместо python использую TCL/Perl**
- d) **Да использую, удобный инструмент**

# Cisco Support Community – Expert Series Webcast

## Возможности интерпретатора Python в NX-OS

Антон Тугай

Инженер центра технической поддержки Cisco TAC в Брюсселе

Октябрь, 2014



# Содержание

- Несколько слов о Python
- Интеграция Python в NX-OS
- Примеры и демонстрация

# Несколько слов о Python

# Основные понятия


Guido van Rossum (Google / Dropbox), 1989

- Python – интерпретируемый язык программирования (scripting), нет предварительной компиляции в исполнимый (бинарный) файл как в C++
- Интерпретация во время выполнения (interactive shell) или трансляция в байт-код и выполнение Python VM
- Доступные версии для Linux, Windows, Mac
  - Основные версии: 2.7(x) and 3.3(x)
  - 2.7 наиболее широко используемая версия нежели 3.3 на момент подготовки презентации

# Происхождения названия языка



**Monty Python**



The Python troupe in 1969  
Back row: Graham Chapman, Eric Idle, Terry Gilliam  
Front row: Terry Jones, John Cleese, Michael Palin

**Medium** Television, film, theatre, audio recordings, books

**Nationality** British<sup>[1]</sup>

**Years active** 1969–1983; 2013–2014

**Genres** Satire, surreal humour, dark comedy

# Производительность

## Производительность?

- (намного) медленней чем C/C++ ... но при этом более гибкий
- Достаточно производительный во многих случаях
- Пример Openstack / Dropbox написаны на Python
- PyPy JIT компилятор для Python (pypy.org)

# Почему Python?

- Отсутствие в CLI языковых конструкций (циклы, условные переходы)
- Современный, многофункциональный и простой язык, наличие большого количества библиотек.
  - TCL: менее популярный, устаревший, not cool.
  - Perl: популярный, “hacky” (некрасивое, громоздкое), отсутствие OOP.
- Легко интегрируемый

## Вопрос №2

# Сталкивались ли вы уже с возможностями Python в NX-OS ?

- a) Нет, не слышал ничего
- b) Слышал о возможностях но не интересовался
- c) Слышал, интересовался, имею представление
- d) Уже использовал или ближайшее время попробую

# Интеграция Python в NX-OS



# Интерпретатор Python в Nexus

- Возможность выполнения python скриптов в коммутаторах серии Nexus (Interactive Mode and Non-interactive Mode (scripts))
- Выполнение CLI команд внутри скрипта.
- Python скрипты могут парсить CLI выходы и выполнять условные действия (e.g syslog, shut/no shut, logging etc.)
- Интеграция с EEM.
- Вызов другого скрипта из текущего
- Не является одинаковым для всех Nexus платформ



Native Python Available from:  
Nexus 5000 – 5.2(1)N1(1)  
Nexus 6000 – day 1  
Nexus 7000 – 6.1(2)  
Nexus 9000 – day 1

Python version :  
Nexus 5000 – 2.7.2  
Nexus 6000 – 2.7.2  
Nexus 7000 – 2.7.2  
Nexus 9000 – 2.7.5

# Запуск Python на Nexus 7K-5K

- Доступен на платформах N3K, N5K, N6K, N7K в “свежих” релизах
- Доступен on N9K с момента запуска платформы (FCS)
- Запускается командой “python” в консоли

```
DCN-N5548UP-3# python
Python 2.7.2 (default, Nov 27 2012, 17:50:33)
[GCC 4.3.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Loaded cisco NxOS lib!
>>> import cisco
>>> dir(cisco)
['BGPSession', 'BufferDepthMonitor', 'CLI', 'CheckPortDiscards', 'CiscoSecret',
'CiscoSocket', 'Feature', 'History', 'IPv4ACL', 'IPv6ACL', 'Interface', 'Key',
'LineParser', 'MacAddressTable', 'OSPFSession', 'Routes', 'SectionParser', 'VRF',
'Vlan', '__all__', '__builtins__', '__doc__', '__file__', '__name__', '__packag
e__', '__path__', 'acl', 'bfd', 'bgp', 'buffer_depth_monitor', 'check_port_disca
rds', 'cisco_secret', 'cisco_socket', 'cli', 'dhcp', 'eigrp', 'feature', 'get_gl
obal_vrf', 'get_valid_port', 'history', 'hsrp', 'interface', 'interface-vlan',
'key', 'lacp', 'line_parser', 'mac_address_table', 'md5sum', 'msdp', 'ospf', 'osp
fv3', 'pim', 'private-vlan', 'ptp', 'rip', 'routes', 'scheduler', 'section_parse
r', 'set_global_vrf', 'show_queues', 'show_run', 'ssh', 'tacacs', 'telnet', 'tra
nsfer', 'udld', 'vlan', 'vpc', 'vrf', 'vrrp', 'vtp']
>>> █
```

# Вызов Python на Nexus 7K-5K

Nexus поддерживает Python v2.7.2 в 2-х режимах

- Интерактивный
- Не интерактивный

## Interactive Mode

```
switch# python
Copyright (c) 2001-2012
Python Software
Foundation; All Rights
Reserved
```

```
switch# >>> print "hello
world"
hello world
switch# >>> exit()
```

## Non Interactive (script) Mode

```
Switch # source crc.py
```

```
-----
Started running CRC checker script
```

```
Finished running CRC checker script
-----
```

```
Switch # dir bootflash:scripts
```

```
946  Oct 30 14:50:36 2013 crc.py
7009 Sep 19 10:38:39 2013
myScript.py
22760 Oct 31 02:51:41 2012 poap.py
```

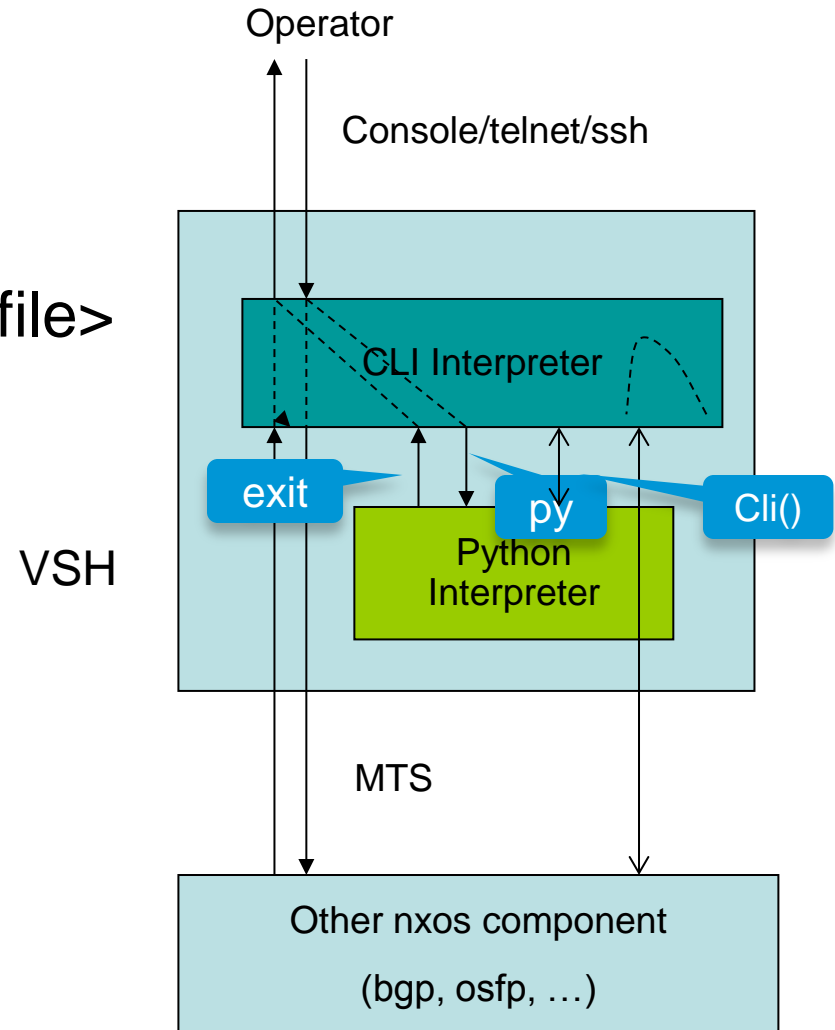


**bootflash:scripts**  
Директория для  
скриптов

# Интеграция Python (1/2)

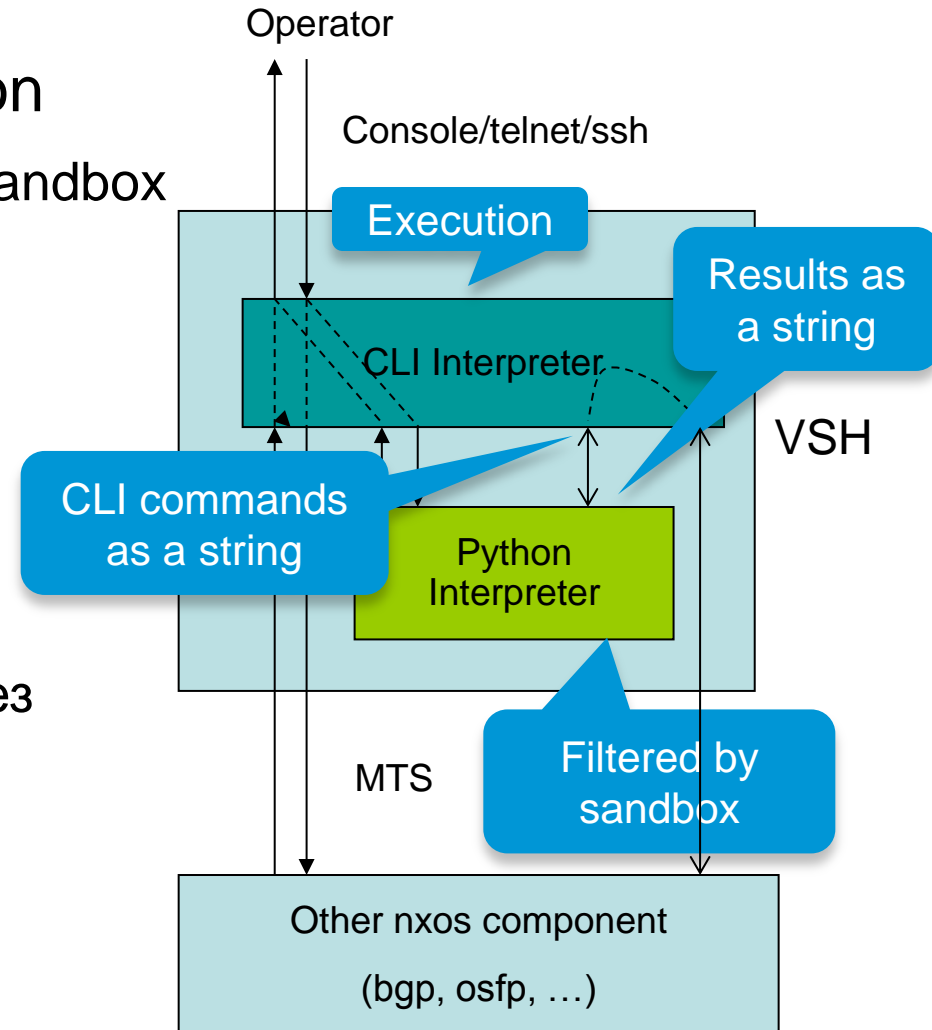
Переключение между CLI и Python

- Интерактивный: `python / exit`
- Не Интерактивный: `source <file>`



## Интеграция Python (2/2)

- Стандартные методы Python
  - Но, могут быть заблокированы sandbox
- Новые функции “cli()”
  - Выполняет CLI команду
    - например, cli("show version")
  - Принимает CLI команду как строковый аргумент
  - CLI команда выполняется через cli интерпретатор
  - Результат возвращается как строка или словарь



# Переключение между CLI и Python

```
switch# show clock
23:54:55.872 UTC Wed May 16 2012
switch# python
switch# >>> cli("conf term ; interface loopback 1")
switch(config-if)# >>> cli("ip address 1.1.1.1/24")
switch(config-if)# >>> cli("exit")
switch(config)# >>> cli("exit")
switch# >>> i=0
switch# >>> while i<8:
switch# ...   i=i+1
switch# ...   cmd = "show module %i" % i
switch# ...   r=clid(cmd)
switch# ...   if "TABLE_modinfo/model" in r.keys():
switch# ...     if r["TABLE_modinfo/model"] == "Nurburgring":
switch# ...       print "got a racer in slot %d" % i
switch# ...       - пустой ввод для окончания цикла
got a racer in slot 3
switch# >>> exit          -- возврат в cli
```

# Особенности реализации Python

- Состояние не сохраняется между вызовами

```
switch# python -- create python interpreter
switch# >>> i = 2
switch# >>> print "var i = %d" % i
var i = 2
switch# >>> cli("configure terminal")
switch(config)# >>> blabla
switch(config)# >>> exit -- destroy python interpreter
switch# - cli interp still at exec mode ("conf t" is lost)
switch# python -- new python interp
switch# >>> print "var i = %d" % i -- previous python cli mode and vars gone
Error: variable 'i' undefined.
switch# >>> exit
switch# conf t ; inter lo 1
switch(config-if)# python -- new python interp
switch(config-if)# >>> -- inherits the cli mode (forked from cli).
```

# CLI Форматирование (1/2)

```
string = cli ("cli-command")      -- возвращает строку  
dictionary = clid ("cli-command") -- возвращает словарь  
clip ("cli-command")             -- стандартный вывод в stdout
```

```
switch# >>> cli("conf ; interface loopback 1")
```

Enter configuration commands, one per line. End with CNTL/Z.

```
switch(config-if)# >>> clip('where detail')
```

```
mode:          conf  
              interface loopback1  
username:      root  
vdc:           switch  
routing-context vrf: default
```

```
switch(config-if)# >>> cli('where detail')
```

```
\x1b[00m mode:          conf\n              interface loopback1\nusername:  
root\n vdc:           switch\n routing-context vrf: default\n
```

```
switch(config-if)# >>> r = cli('where detail') ; print r
```

(same output as clip() above!)



# CLI Форматирование (2/2)

```
switch(config-if)# >>> i=0
switch(config-if)# >>> while i<3:
switch(config-if)# ...     i=i+1
switch(config-if)# ...     cli('ip addr 1.1.1.1/24')
switch(config-if)# ...
switch(config-if)# >>> cli('end')
switch# >>> r = clid('show version')
switch# >>> for k in r.keys():
switch# ...     print "%30s" % k, " = %s" % r[k]
switch# ...
                cpu_name      = Intel(R) Xeon(R) CPU
                rr_sys_ver     = 6.2(0.110)
                manufacturer    = Cisco Systems, Inc.
                isan_file_name  = bootflash:///full
                rr_ctime        = Wed May 16 02:40:57 2012
                proc_board_id   = JAF1417AGCB
                bios_cmpl_time  = 02/20/10
                kickstart_ver_str = 6.1(1) [build 6.1(0.292)] [gdb]
                isan_tmstamp    = 05/16/2012 02:26:02
switch# >>> exit
```

# Не интерактивный режим Python

```
switch# show file bootflash:scripts/test1.py  
#!/bin/env python
```

```
i=0
```

Вызов python

Директория для скриптов

```
while i<3:  
    r=clip('show version')  
    uptime_name='/@/show/version/__readonly__/kern_uptm_secs'  
    print uptime_name, r[uptime_name]  
    clid('sleep 1')  
    i=i+1
```

```
switch# source test1.py -- default directory is /bootflash/scripts  
/@/show/version/__readonly__/kern_uptm_secs 36  
/@/show/version/__readonly__/kern_uptm_secs 38  
/@/show/version/__readonly__/kern_uptm_secs 40  
switch#
```

# Python в NxOS: где мой скрипт?

- Если импортировать модуль OS (`import os`) можно напрямую обращаться к своему скрипту:

```
DCN-N5548UP-3# python
Python 2.7.2 (default, Nov 27 2012, 17:50:33)
[GCC 4.3.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Loaded cisco NxOS lib!
>>> import os
>>> os.chdir('/bootflash')
>>> "mybasicscript.py" in os.listdir(os.getcwd())
True
```

- Возможность добавления внешних скриптов и модулей и их использование

```
>>> mybasicscript = open('mybasicscript.py','r')
>>> for line in mybasicscript:
...     print(line)
...
#!/usr/bin/env python

import sys

argvList = sys.argv[1:]

for argv in argvList:

    print 'Hello ' + argv
```

# Sandbox (песочница) в действии

- Блокирование потенциально опасных вызовов

```
switch# python
```

```
Copyright (c) 2001-2012 Python Software Foundation; All Rights Reserved
```

```
switch# >>> import os
```

```
switch# >>> os.getcwd()
```

```
'/bootflash'
```

```
switch# >>> os.chdir("/isan/bin")
```

```
Permission denied. Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
```

```
OSError: [Errno 13] Permission denied: '/isan/bin'
```

```
switch# >>> os.system("cd /isan/bin")
```

```
system(cd /isan/bin): rejected!
```

```
-1
```

```
switch# >>> f=open("/isan/bin/vsh", "r")
```

```
Permission denied. Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
```

```
IOError: [Errno 13] Permission denied: '/isan/bin/vsh'
```

```
switch# >>> f=open("/bootflash/alias", "r")
```

```
switch# >>>
```

# Python интегрирован с EEM

- Любой Python скрипт может быть вызван как EEM действие.

```
n7k# conf
n7k# event manager applet link_monitor
n7k# event syslog pattern "IF_UP"
n7k# action 1 cli source cdp_description.py
n7k# exit
```

```
n7k# conf
n7k# event manager applet link_monitor
n7k# event syslog pattern "IF_UP:|IF_.*DOWN:"
n7k# action 1 cli source link_monitor_nexus7000.py -m eth1/2-5 -a
eth1/7-8 -l "$_syslog_msg"
n7k# exit
```

Исходный код `cdp_description.py` и `link_monitor_nexus7000.py` находятся по ссылке <https://github.com/datacenter/nexus7000>

# Примеры и демонстрация

# Настройка CLI через Python

- Возможно использовать только метод `cli()` для перехода в режим конфигурации и настройки :

```
TME-1-9508-1(config)# >>> cli("configure terminal")
''
TME-1-9508-1(config)# >>> cli("interface loop 9")
''
TME-1-9508-1(config-if)# >>> cli("ip address 1.1.1.1/32")
''
TME-1-9508-1(config-if)# >>> cli("exit")
''
```

```
TME-1-9508-1# 2013 Oct 10 09:02:12 TME-1-9508-1 %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configured
TME-1-9508-1# sh int loop 99508-1 %ETHPORT-5-IF_UP: Interface loopback9 is up
loopback9 is up
admin state is up,
Hardware: Loopback
Internet Address is 1.1.1.1/32
MTU 1500 bytes, BW 8000000 Kbit, DLY 5000 usec
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation LOOPBACK, medium is broadcast
Auto-mdix is turned off
  0 packets input 0 bytes
  0 multicast frames 0 compressed
```

# Особенности использования cli()

- В случае если необходимо ввести несколько команд в одну строку, используется пробел и “;” как показано:

```
>>> import cli
>>> cli.cli("conf t ; int loop 999 ; ip addr 1.2.3.4/32")
''
```

```
>>> cli.cli('show int lo999')
'loopback999 is up\nadmin state is up,\n Hardware: Loopback\n Internet Address is 1.2.3.4/32\n /255, rxload 1/255\n Encapsulation LOOPBACK, medium is broadcast\n Auto-mdix is turned off\n s 0 frame 0 overrun 0 fifo\n 0 packets output 0 bytes 0 underruns\n 0 output errors 0 colli
>>> █
```



# Получение данных от пользователя

- Пример использования:

```
DCN-N5548UP-3# python
Python 2.7.2 (default, Nov 27 2012, 17:50:33)
[GCC 4.3.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Loaded cisco NxOS lib!
>>> help(raw_input)
Help on built-in function raw_input in module __builtin__:

raw_input(...)
    raw_input([prompt]) -> string

    Read a string from standard input.  The trailing newline is stripped.
    If the user hits EOF (Unix: Ctl-D, Windows: Ctl-Z+Return), raise EOFError.
    On Unix, GNU readline is used if enabled.  The prompt string, if given,
    is printed without a trailing newline before reading.

>>> userin=raw_input("Please enter an IP address:")
Please enter an IP address:5.6.7.8
>>> userin
'5.6.7.8'
>>>
```

# NxOS Python APIs

- NxOS поставляется с нативными APIs которые контролируют некоторые функции (features)
  - К примеру “show vlan” CLI можно получить через объект Vlan() вызвав метод show\_vlan() на нем.
  - APIs позволяют модифицировать маршруты, Vlans, ARP записи и т/д
- Документация оформлена не наилучшим образом ССО:
- [http://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus5000/sw/python/api/python\\_api/getting\\_started.html](http://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus5000/sw/python/api/python_api/getting_started.html)
- [http://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/6\\_x/nx-os/fundamentals/configuration/guide/b\\_Cisco\\_Nexus\\_7000\\_Series\\_NX-OS\\_Fundamentals\\_Configuration\\_Guide\\_Release\\_6-x/b\\_Cisco\\_Nexus\\_7000\\_Series\\_NX-OS\\_Fundamentals\\_Configuration\\_Guide\\_Release\\_6-x\\_chapter\\_01011.html](http://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/6_x/nx-os/fundamentals/configuration/guide/b_Cisco_Nexus_7000_Series_NX-OS_Fundamentals_Configuration_Guide_Release_6-x/b_Cisco_Nexus_7000_Series_NX-OS_Fundamentals_Configuration_Guide_Release_6-x_chapter_01011.html)
- Не все платформы предоставляют одинаковый набор API
  - N3K и N9K более развиты нежели N5K/N7K

# Доступ объектам NxOS (модули cisco.xxx)

- Пример создания интерфейсов:

```
>>> from cisco.interface import *
>>> Interface.interfaces()
['mgmt0', 'Ethernet1/1', 'Ethernet1/2', 'Ethernet1/3', 'Ethernet1/4', 'Ethernet1/5',
 'Ethernet1/12', 'Ethernet1/13', 'Ethernet1/14', 'Ethernet1/15', 'Ethernet1/16', 'Et
Ethernet1/23', 'Ethernet1/24', 'Ethernet1/25', 'Ethernet1/26', 'Ethernet1/27', 'Et
thernet1/34', 'Ethernet1/35', 'Ethernet1/36', 'loopback999']
>>> my_int = Interface("Loopback999")
>>> my_int.config()
'\n!Command: show running-config interface loopback999\n!Time: Thu May  8 03:25:14
>>> my_int.show()
<cisco.interface.ShowInterface object at 0xe4e9788c>
>>> my_int.parsed_if("Loopback999")
('loopback', [999])
>>> my_int.set_description("Python set this description!")
True
>>> my_int.set_state(s="shut")
True
>>> my_int.set_state(s="no shut")
True
>>> my_int.set_ipaddress(ip_address="1.2.3.4", mask=32)
True
>>> my_int.config()
'\n!Command: show running-config interface loopback999\n!Time: Thu May  8 03:26:54
n!\n  ip address 1.2.3.4/32\n\n'
>>> █
```

# Модули Python в N9K

- Автоматически подключаемые модули `cisco.nnn` при импорте “`import cisco`”:

```
NAME
  cisco

FILE
  /isan/python/scripts/cisco/__init__.py

PACKAGE CONTENTS
  acl
  bgp
  buffer_depth_monitor
  check_port_discards
  cisco_secret
  feature
  history
  interface
  ipaddress
  key
  line_parser
  mac_address_table
  nxccli
  ospf
  routemap
  routes
  section_parser
  ssh
  system
  tacacs
  transfer
  vlan
  vrf
```

# Python NX-OS API примеры

- Работа с таблицей маршрутизации:

```
>>> from cisco.routes import *
>>> my_route = Routes()
>>> my_route.add_route(srcIp='172.16.65.0', prefix='24', mask="24", nexthop="10.48.59.192")
Vrf: default
Route to be added: ip route 172.16.65.0/24 10.48.59.192
''
>>>
```

- Создание ACL:

```
>>> from cisco.acl import *
>>> my_acl = IPv4ACL("bar")

>>> my_acl.deny("ip", "any", "any")
True
>>> my_acl.create()
True
>>> my_acl.show()

IP access list bar
    10 deny ip any any

['', 'IP access list bar', '\t10 deny ip any any ', '']
>>>
```

- Системные объекты

```
>>> from cisco.system import *
>>> my_system = System()
>>> my_system.get_mgmt0_ip()
'10.48.59.195/25'
>>>
```

# Импортирование внешних модулей

- Скопировать tar.gz архив на bootflash:
- Распаковать (`tar extract module-name.tar.gz`)
- `cd bootflash/module-name`
- Скопировать все файлы на bootflash:
- Запустить `python setup.py install`
- Пример с модулем prettytable:

```
Installed /isan/lib/python2.7/site-packages/prettytable-0.7.2-py2.7.egg
Processing dependencies for prettytable==0.7.2
Finished processing dependencies for prettytable==0.7.2
DCN-N5548UP-3# python
Python 2.7.2 (default, Nov 27 2012, 17:50:33)
[GCC 4.3.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
imLoaded cisco NxOS lib!
>>> import prettytable
>>>
```

# Форматирование вывода с Prettytable

- Пример работы Prettytable из NxOS

```
>>> x=prettytable.PrettyTable()
>>> x.add_column("City name",["Adelaide","Brisbane","Darwin","Hobart","Sydney","Melbourne","Perth"])
>>> x.add_column("Area", [1295, 5905, 112, 1357, 2058, 1566, 5386])
>>> x.add_column("Population", [1158259, 1857594, 120900, 205556, 4336374, 3806092, 1554769])
>>> x.add_column("Annual Rainfall",[600.5, 1146.4, 1714.7, 619.5, 1214.8, 646.9, 869.4])
>>> print(x)
+-----+-----+-----+-----+
| City name | Area | Population | Annual Rainfall |
+-----+-----+-----+-----+
| Adelaide | 1295 | 1158259 | 600.5 |
| Brisbane | 5905 | 1857594 | 1146.4 |
| Darwin | 112 | 120900 | 1714.7 |
| Hobart | 1357 | 205556 | 619.5 |
| Sydney | 2058 | 4336374 | 1214.8 |
| Melbourne | 1566 | 3806092 | 646.9 |
| Perth | 5386 | 1554769 | 869.4 |
+-----+-----+-----+-----+
>>> █
```

# Генерация syslog сообщений

Скрытая команда  
“logit” в CLI

```
>>>  
>>> cisco.cli('logit Hello World!')  
  
>>> 2011 Jan 7 13:13:01 AE-N5548-1 %USER-2-SYSTEM_MSG: systest from 0: Hello World! - vsh
```

Используя syslog модуль

```
>>> import syslog  
>>> syslog.syslog(1, 'Hello World!')  
>>> 2011 Jan 7 13:12:05 AE-N5548-1 %USER-1-SYSTEM_MSG: Hello World! - python  
>>> syslog.syslog(2, 'Hello World!')  
>>> 2011 Jan 7 13:12:18 AE-N5548-1 %USER-2-SYSTEM_MSG: Hello World! - python
```

## Вывод в консоли:

```
AE-N5548-1# sh log last 3  
2011 Jan 7 13:08:59 AE-N5548-1 %USER-2-SYSTEM_MSG: systest from 0: 1, hello - vsh  
2011 Jan 7 13:12:05 AE-N5548-1 %USER-1-SYSTEM_MSG: Hello World! - python  
2011 Jan 7 13:12:18 AE-N5548-1 %USER-2-SYSTEM_MSG: Hello World! - python
```



# Автоматизация рутинных задач

## Добавление описания интерфейсов на основании вывода “show cdp neighbors”

Выполнение “show cdp neighbors”

Выделение информации о локальном порте и подключенном устройстве

```
Switch# Conf t
Switch(config)# Interface eth x/y
Switch(config-int)# description
“Connected to ABC switch port EFG”
```

## Логирование и выключение порта в случае превышения количества ошибок CRC свыше порога threshold

Run “show interface”

Поиск CRC ошибок, какие интерфейсы подвержены

Сравнение с порогом

**Выполнение действий**

1. Логирование ошибки
2. “Shut interface”.

# Добавление описания интерфейсов на основании вывода “show cdp neighbors”

Using cisco module

Using xml module

Execute cli(“show cdp neighbors | xml”)

Iterate over each row in the xml output

Using cli() function to Add interface description

```
#!/usr/bin/env python
from cisco import cli
import sys
import xml.etree.cElementTree as ET
cdp_dict = {}

# Get interface information in XML format
print 'Executing add interface description script '

def extract_cdp_info():
    #cli | xml
    raw = cli('show cdp neighbors | xml | exclude "]]>]]>")
    # Load and parse XML
    tree = ET.ElementTree(ET.fromstring(raw))
    data = tree.getroot()
    cdp_info = '{http://www.cisco.com/nxos:6.2.5.:cdpd}'
    for i in data.iter(cdp_info + 'ROW_cdp_neighbor_brief_info'):
        #parse interface, port and platform info from xml output and
        #save the info in a dictionary
        if i.find(cdp_info + 'intf_id').text not in cdp_dict:
            cdp_dict[intf_id] = {}
            cdp_dict[intf_id]['intf_id'] = i.find(cdp_info + 'intf_id').text
            cdp_dict[intf_id]['platform_id'] = i.find(cdp_info +
'platform_id').text
            cdp_dict[intf_id]['port_id'] = port_id = i.find(cdp_info +
'port_id').text

#add description based on the cdp information
def add_description():
    for key, value in cdp_dict.items():
        if 'port_id' in value and 'platform_id' in value and 'intf_id' in
value:
            cli("conf t ")
            cli('interface ' + value['intf_id'] + ' ; description connected to '
+ value['platform_id'] + ' ' + value['port_id'])
def main():
    #extract cdp neighbors info
    extract_cdp_info()
    #add description to interfaces based on cdp info.
    add_description()
```

# Добавление описания интерфейсов на основании вывода `show cdp neighbors`

До

```
n7k-148# sh cdp neighbors
```

Device-ID Port ID	Local Intrfce	Hldtme	Capability	Platform
n7k1-ace-comp(TBM14375659) mgmt0	130	R S I s	N7K-C7010	Eth7/3
n7K-147(JAF1545CJCL) Eth1/15	134	R S s	N7K-C7010	Eth1/3
n7K-147(JAF1545CJCL) Eth1/17	136	R S s	N7K-C7010	Eth1/4

```
n7k-148# sh int Eth1/15 desc
```

```
-----  
Port      Type  Speed  Description  
-----  
Eth1/15   eth   10G    --
```

```
n7k-148# sh int Eth1/17 desc
```

```
-----  
Port      Type  Speed  Description  
-----  
Eth1/17   eth   10G    --
```

Выполнение скрипта  
n7k-148# **source cdp\_description.py**

После

```
n7k-148# source cdp_description.py  
Executing add interface description script
```

```
n7k-148# sh int Eth1/15 desc
```

```
-----  
Port      Type  Speed  Description  
-----  
Eth1/15   eth   10G    connected to N7K-C7010  
Ethernet1/3
```

```
n7k-148# sh int Eth1/17 desc
```

```
-----  
Port      Type  Speed  Description  
-----  
Eth1/17   eth   10G    connected to N7K-C7010  
Ethernet1/4
```

# Выключение интерфейса если количество CRC errors > порога

Использование  
встроенных модулей  
re, sys

Создание лога  
syslog ()

cli() API для  
выполнения  
CLI команд

```
#!/bin/env python
from cisco import *
import re
import sys
threshold = 10

def shut_interface_and_creat_syslog(counter):
    # print "input counter:",counter
    interfaces_list = cli("show int description | grep eth")
    s = interfaces_list.split("\n")
    match = re.search("(.* eth \s*", s[counter])
    match.group(1)
    #create a syslog
    syslog(1,"Found CRC errors > threshold on intreface eth",
    match.group(1));

    #call cli to shut the interface.
    cli("conf t")
    cli("interface " + match.group(1))
    cli("shut")
    print "shutting interface ", match.group(1)

def main():
    counter = 0
    print "Started running CRC checker script"
    crc = cli("show interface | grep CRC")
    s=crc.split("\n")

    for line in s:
        match = re.search("\s* giants (.*) CRC/FCS \s*", line)
        if match and match.group(1):
            if int(match.group(1)) > threshold:
                print "found CRC errors > ", threshold
                shut_interface(counter)
            counter +=1
    print "finished running CRC checker script"

if __name__ == "__main__":
    sys.exit(main())
```

## Вопрос №3

# Насколько полезно наличие Python в NX-OS полезна и применима в ваших условиях

- a) **Скорее бесполезно**
- b) **Определенные интересные моменты есть**
- c) **Полезно и применимо**
- d) **Еще не определился, надо попробовать**

# Отправьте свой вопрос сейчас!

Используйте панель Q&A, чтобы задать вопрос.  
Эксперты ответят на Ваши вопросы.

# Сессия «Спросить Эксперта» с Антоном Тугаем



Получить дополнительную информацию, а также задать вопросы эксперту в рамках данной темы Вы можете на странице, доступной по ссылке:

<https://supportforums.cisco.com/community/russian/expert-corner>

Вы можете получить видеозапись данного семинара и текст сессии Q&A в течении ближайших 5 дней по следующей ссылке

<https://supportforums.cisco.com/community/russian/expert-corner/webcast>

# Вебинар на русском языке

**Тема:** Основные элементы архитектуры и настройки решения Cisco Collaboration Edge



**во вторник, 25 ноября, в 12.00 мск**

Присоединяйтесь к эксперту Cisco

**Михаилу Щекотилову**

В рамках сессии Михаил расскажет об архитектуре решения Cisco Collaboration Edge, а затем проведет живую демонстрацию настройки основных элементов этого решения.



# Приглашаем Вас активно участвовать в Cisco Support Community и социальных сетях

<https://supportforms.cisco.com/community/russian>



<http://www.facebook.com/CiscoSupportCommunity>



[http://twitter.com/#!/cisco\\_support](http://twitter.com/#!/cisco_support)



<http://www.youtube.com/user/ciscosupportchannel>



<https://plus.google.com/110418616513822966153?prsrc=3#110418616513822966153/posts>



<http://itunes.apple.com/us/app/cisco-technical-support/id398104252?mt=8>



[https://play.google.com/store/apps/details?id=com.cisco.swtg\\_android](https://play.google.com/store/apps/details?id=com.cisco.swtg_android)



<http://www.linkedin.com/groups/CSC-Cisco-Support-Community-3210019>



Newsletter Subscription:

[https://tools.cisco.com/gdrp/coiga/showsurvey.do?surveyCode=589&keyCode=146298\\_2&PHYSICAL%20FULFILLMENT%20Y/N=NO&SUBSCRIPTION%20CENTER=YES](https://tools.cisco.com/gdrp/coiga/showsurvey.do?surveyCode=589&keyCode=146298_2&PHYSICAL%20FULFILLMENT%20Y/N=NO&SUBSCRIPTION%20CENTER=YES)

# Мы также предоставляем Вашему вниманию Сообщества на других языках

Если Вы говорите на Испанском, Португальском или Японском, мы приглашаем Вас принять участие в Сообществах:

- Русском → <https://supportforums.cisco.com/community/russian>
- Испанском → <https://supportforums.cisco.com/community/5591/comunidad-de-soporte-de-cisco-en-espanol>
- Португальском → <https://supportforums.cisco.com/community/5141/comunidade-de-suporte-cisco-em-portugues>
- Японском → <http://www.csc-china.com.cn/>

*Спасибо за Ваше время*

*Пожалуйста, участвуйте в опросе*

