Основы Python

Списки

Список (List)

Список это изменяемый упорядоченный тип данных.
Список в Python это последовательность элементов, разделенных между собой запятой и заключенных в квадратные скобки.

```
In [9]: [1,2,3,4,5]
Out[9]: [1, 2, 3, 4, 5]

In [10]: ['switchport', 'mode', 'access']
Out[10]: ['switchport', 'mode', 'access']

In [11]: [[1, 2, 3], 4, 5]
Out[11]: [[1, 2, 3], 4, 5]
```

Список (List)

Так как список, это упорядоченный тип данных, то, как и в строках, в списках можно обращаться к элементу по номеру, делать срезы:

```
In [4]: list3 = [1, 20, 4.0, 'word']

In [5]: list3[1]
Out[5]: 20

In [6]: list3[1::]
Out[6]: [20, 4.0, 'word']

In [7]: list3[-1]
Out[7]: 'word'

In [8]: list3[::-1]
Out[8]: ['word', 4.0, 20, 1]
```

Так как списки изменяемые, элементы списка можно менять:

```
In [13]: list3
Out[13]: [1, 20, 4.0, 'word']

In [14]: list3[0] = 'test'

In [15]: list3
Out[15]: ['test', 20, 4.0, 'word']
```

Полезные методы для работы со списками

При рассмотрении полезных методов для строк, мы остановились на том, что получили итоговый список vlans:

```
In [15]: vlans = ['10', '20', '30', '100-200']
```

Теперь, допустим, что после необходимых операций с списком VLAN(отбросим последний диапазон), нужно опять записать их файл. То есть надо сделать так, чтобы теперь все значений списка были собраны в одну строку, но записаны через запятую (сделать операцию обратную split()).

Метод join() позволяет собрать список строк в одну строку с разделителем, который указан в join():

```
In [16]: vlans = ['10', '20', '30', '100-200']

In [17]: ','.join(vlans[:-1])
Out[17]: '10,20,30'
```

Полезные методы для работы со списками

Метод append() позволяет добавить в конец списка указанные элемент:

```
In [18]: vlans = ['10', '20', '30', '100-200']

In [19]: vlans.append('300')

In [20]: vlans
Out[20]: ['10', '20', '30', '100-200', '300']
```

Если нужно объединить два списка, то можно использовать два способа. Метод extend() и операцию сложения:

```
In [21]: vlans = ['10', '20', '30', '100-200']

In [22]: vlans2 = ['300', '400', '500']

In [23]: vlans.extend(vlans2)

In [24]: vlans
Out[24]: ['10', '20', '30', '100-200', '300', '400', '500']

In [25]: vlans + vlans2
Out[25]: ['10', '20', '30', '100-200', '300', '400', '500', '300', '400', '500']

In [26]: vlans
Out[26]: ['10', '20', '30', '100-200', '300', '400', '500']
```

Обратите внимание, что метод extend() расширяет список "на месте", а при операции сложения, выводится итоговый суммарный список, но исходные списки не меняются.

Полезные методы для работы со списками

Метод pop() позволяет удалить элемент, который соответствует указанному номеру. По умолчанию берется последний элемент списка. При этом метод выводит этот элемент:

```
In [28]: vlans = ['10', '20', '30', '100-200']

In [29]: vlans.pop(-1)
Out[29]: '100-200'

In [30]: vlans
Out[30]: ['10', '20', '30']
```

В методе remove() надо указывать сам элемент, который надо удалить, а не его номер в списке. Кроме того, remove() не отображает удаленный элемент:

```
In [31]: vlans = ['10', '20', '30', '100-200']

In [32]: vlans.remove(-1)
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-32-f4ee38810cb7> in <module>()
----> 1 vlans.remove(-1)

ValueError: list.remove(x): x not in list

In [33]: vlans.remove('20')

In [34]: vlans
Out[34]: ['10', '30', '100-200']
```

Полезные методы для работы со списками

Метод index() используется для того, чтобы определить под каким номером в списке хранится элемент:

```
In [35]: vlans = ['10', '20', '30', '100-200']

In [36]: vlans.index('30')
Out[36]: 2
```

Вставить элемент на определенное место в списке:

```
In [37]: vlans = ['10', '20', '30', '100-200']

In [38]: vlans.insert(1,'15')

In [39]: vlans
Out[39]: ['10', '15', '20', '30', '100-200']
```

Варианты создания списка

Создание списка с помощью литерала:

```
In [1]: vlans = [10, 20, 30, 50]
```

Создание списка с помощью функции list():

```
In [2]: list1 = list('router')

In [3]: print list1
['r', 'o', 'u', 't', 'e', 'r']
```

Генераторы списков (list comprehention):

```
In [4]: list2 = ['FastEthernet0/'+ str(i) for i in range(10)]

In [5]: list2
Out[6]:
['FastEthernet0/0',
 'FastEthernet0/1',
 'FastEthernet0/2',
 'FastEthernet0/3',
 'FastEthernet0/4',
 'FastEthernet0/5',
 'FastEthernet0/6',
 'FastEthernet0/7',
 'FastEthernet0/8',
 'FastEthernet0/9']
```

Python для сетевых инженеров

Автор курса: Наташа Самойленко
nataliya.samoylenko@gmail.com