

Основы Python

Python для сетевых инженеров

Типы данных в Python

Типы данных в Python

В Python есть несколько стандартных типов данных:

- Numbers (числа)
- Strings (строки)
- Lists (списки)
- Dictionary (словари)
- Tuples (кортежи)
- Sets (множества)
- Boolean

Типы данных в Python

Изменяемые:

- Списки
- Словари
- Множества

Неизменяемые

- Числа
- Строки
- Кортежи

Упорядоченные:

- Списки
- Кортежи
- Строки

Неупорядоченные:

- Словари
- Множества

Числа

Пример различных типов числовых значений:

int (40, -80, 0x0800)

float (1.5, -30.7)

long (52934861L)

In [1]: 3 * 5

Out[1]: 15

In [2]: 3**5

Out[2]: 243

In [3]: 100%3

Out[3]: 1

In [4]: 3**50

Out[4]: 717897987691852588770249L

In [5]: 10.0/3

Out[5]: 3.3333333333333335

Строки (Strings)

Строки это неизменяемый, упорядоченный тип данных. Строка в Python это последовательность символов, заключенная в кавычки.

```
In [6]: 'Hello'
Out[6]: 'Hello'
```

```
In [7]: "Hello"
Out[7]: 'Hello'
```

```
In [8]: """
...: interface Tunnel0
...:   ip address 10.10.10.1 255.255.255.0
...:   ip mtu 1416
...:   ip ospf hello-interval 5
...:   tunnel source FastEthernet1/0
...:   tunnel protection ipsec profile DMVPN
...: """
```

```
Out[8]: '\ninterface Tunnel0\n ip address 10.10.10.1 255.255.255.0\n ip mtu 1416\n ip ospf hello-interval 5\n tunnel source FastEthernet1/0\n tunnel protection ipsec profile DMVPN\n'
```

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]
```

Словарь (Dictionary)

Словари – это изменяемый, неупорядоченный тип данных.

```
In [12]: london = {'name': 'London1', 'location': 'London Str',  
                  'vendor': 'Cisco', 'model': '4451', 'IOS': '15.4'}
```

```
In [13]: london = {  
    .....:     'id': 1,  
    .....:     'name': 'London',  
    .....:     'IT_VLAN': 320,  
    .....:     'User_VLAN': 1010,  
    .....:     'Mngmt_VLAN': 99,  
    .....:     'to_name': None,  
    .....:     'to_id': None,  
    .....:     'port': 'G1/0/11'  
    .....: }
```


Кортеж (Tuple)

Кортеж это **неизменяемый** упорядоченный тип данных.

Кортеж в Python это последовательность элементов, разделенных между собой запятой и заключенных в скобки.

```
In [14]: (1,2,4,5,6,7)
Out[14]: (1, 2, 4, 5, 6, 7)
```

```
In [15]: 6,7,8,8,9
Out[15]: (6, 7, 8, 8, 9)
```

```
In [16]: (2,)
Out[16]: (2,)
```

Множество (Set)

Множество это **изменяемый неупорядоченный** тип данных. В множестве всегда содержатся только **уникальные** элементы.

Множество в Python это последовательность элементов, разделенные между собой запятой и заключенные в фигурные скобки.

```
In [17]: vlans = [10, 20, 30, 40, 100, 10]
```

```
In [18]: set(vlans)
```

```
Out[18]: set([40, 100, 10, 20, 30])
```

```
In [19]: {1,2,2,3,5,4,6,3}
```

```
Out[19]: set([1, 2, 3, 4, 5, 6])
```

Упорядоченные типы данных

Упорядоченные типы данных

В Python такие типа данных являются упорядоченными:

- Списки
- Кортежи
- Строки

Для всех упорядоченных типов данных есть общие операции.

Операции работающие для всех упорядоченных типов данных

<code>x in s</code>	True если элемент в <i>s</i> равен <i>x</i> , иначе False
<code>x not in s</code>	False если элемент в <i>s</i> равен <i>x</i> , иначе True
<code>s + t</code>	объединение <i>s</i> и <i>t</i>
<code>s * n, n * s</code>	эквивалентно добавлению <i>s</i> к себе <i>n</i> раз
<code>s[i]</code>	<i>i</i> -ый элемент <i>s</i> , начиная с 0
<code>s[i:j]</code>	срез <i>s</i> с <i>i</i> до <i>j</i>
<code>s[i:j:k]</code>	срез <i>s</i> с <i>i</i> до <i>j</i> с шагом <i>k</i>
<code>len(s)</code>	длина <i>s</i>
<code>min(s)</code>	наименьший элемент <i>s</i>
<code>max(s)</code>	наибольший элемент <i>s</i>
<code>s.index(x)</code>	индекс первого местонахождения <i>x</i> в <i>s</i>
<code>s.count(x)</code>	количество элемента <i>x</i> в <i>s</i>

В таблице, *s* и *t* это последовательности одного типа; *n*, *i* и *j* числа

Операции работающие для всех упорядоченных типов данных на примере списка

```
In [26]: s = [0,1,2,3,4,5]
```

```
In [27]: 1 in s  
Out[27]: True
```

```
In [28]: 1 not in s  
Out[28]: False
```

```
In [29]: s + [6,7,8]  
Out[29]: [0, 1, 2, 3, 4, 5, 6, 7, 8]
```

```
In [30]: s * 2  
Out[30]: [0, 1, 2, 3, 4, 5, 0, 1, 2, 3, 4, 5]
```

```
In [31]: 2 * s  
Out[31]: [0, 1, 2, 3, 4, 5, 0, 1, 2, 3, 4, 5]
```

```
In [32]: s[3]  
Out[32]: 3
```

```
In [34]: s[1:4]  
Out[34]: [1, 2, 3]
```

```
In [35]: s[1:5:2]  
Out[35]: [1, 3]
```

```
In [36]: len(s)  
Out[36]: 6
```

```
In [37]: min(s)  
Out[37]: 0
```

```
In [38]: max(s)  
Out[38]: 5
```

```
In [39]: s.index(4)  
Out[39]: 4
```

```
In [40]: s.count(4)  
Out[40]: 1
```

Операции работающие для всех упорядоченных типов данных на примере строк

```
In [43]: s = "String is a sequence"
```

```
In [44]: 'is' in s  
Out[44]: True
```

```
In [45]: 'is' not in s  
Out[45]: False
```

```
In [46]: s + " of letters"  
Out[46]: 'String is a sequence of letters'
```

```
In [47]: s * 2  
Out[47]: 'String is a sequenceString is a  
sequence'
```

```
In [48]: 2 * s  
Out[48]: 'String is a sequenceString is a  
sequence'
```

```
In [49]: s[3]  
Out[49]: 'i'
```

```
In [50]: s[7:10]  
Out[50]: 'is '
```

```
In [51]: s[1:14:2]  
Out[51]: 'tigi e'
```

```
In [52]: len(s)  
Out[52]: 20
```

```
In [53]: min(s)  
Out[53]: ' '
```

```
In [54]: max(s)  
Out[54]: 'u'
```

```
In [55]: s.index('is')  
Out[55]: 7
```

```
In [56]: s.count('i')  
Out[56]: 2
```

Python для сетевых инженеров

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