

Python Basics

S.R. Doty

August 27, 2008

Preview from Notesale.co.uk
Page 1 of 20

Contents

1	Preliminaries	4
1.1	What is Python?	4
1.2	Installation and documentation	4
2	Getting started	4
2.1	Running Python as a calculator	4
2.2	Quitting the interpreter	6
2.3	Loading commands from the library	6
2.4	Defining functions	7
2.5	Files	8
2.6	Testing code	8
2.7	Scripts	8

3	Python commands	9
3.1	Comments	9
3.2	Numbers and other data types	9
3.2.1	The type function	9
3.2.2	Strings	10
3.2.3	Lists and tuples	10
3.2.4	The range function	11
3.2.5	Boolean values	11
3.3	Expressions	11
3.4	Operators	11
3.5	Variables and assignment	13
3.6	Decisions	13
3.7	Loops	14
3.7.1	for loop	14
3.7.2	while loop	15
3.7.3	else in loops	15
3.7.4	break , continue , and pass	16
3.8	Lists	16
3.8.1	Length of a list, len()	17
3.8.2	Sublist (slicing)	17
3.8.3	Joining two lists	18
3.8.4	List methods	18
3.9	Strings	19

```
>>> x=range(0,20,2)
>>> x
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
>>> x[2:5]
[4, 6, 8]
>>> x[0:5]
[0, 2, 4, 6, 8]
```

When taking a slice, either parameter **start** or **end** may be omitted: if **start** is omitted then the slice consists of all items up to, but not including, the one at index position **end**, similarly, if **end** is omitted the slice consists of all items starting with the one at position **start**. For instance, with the list **x** as defined above we have

```
>>> x[:5]
[0, 2, 4, 6, 8]
>>> x[2:]
[4, 6, 8, 10, 12, 14, 16, 18]
```

In this case, **x[:5]** is equivalent to **x[0:5]** and **x[2:]** is equivalent to **x[2:len(x)]**.

There is an optional third parameter in a slice, which if present represents an increment, just as in the range function. For example,

```
>>> list = range(20)
>>> list
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]
>>> list[0:16:2]
[0, 2, 4, 6, 8, 10, 12, 14]
>>> list[0:15:2]
[0, 2, 4, 6, 8, 10, 12, 14]
```

Notice that one may cleverly use a *negative* increment to effectively reverse a list, as in:

```
>>> list[18::-1]
[17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
```

In general, the slice **x[len(x)::-1]** reverses any existing list **x**.

3.8.3 Joining two lists

Two existing lists may be *concatenated* together to make a longer list, using the **+** operator:

```
>>> [2,3,6,10] + [4,0,0,5,0]
[2, 3, 6, 10, 4, 0, 0, 5, 0]
```

3.8.4 List methods

If **x** is the name of an existing list, we can **append** an item **item** to the end of the list using **x.append(item)**

For example,

```
>>> x = [3, 6, 8, 9]
>>> x.append(999)
>>> x
[3, 6, 8, 9, 999]
```