Python Reference Guide

Version 1.3

Main data types

boolean = True / False
integer = 10
float = 10.01
string = "123abc"
list = [value1, value2, …]

Numeric Operators

+	addition
-	subtraction
*****	multiplication
/	division
******	exponent
%	modulus
//	floor division

Comparison Operators

==	equal
!=	not equal
>	higher
<	lower
>=	higher or equal
<=	lower or equal

Boolean Operators

and	logical AND
or	logical OR
not	logical NOT

Special characters

#	comment
\n	new line
\t	tab
\<char>	escape char

Assignment operators

=	simple assignment	x=y
+=	increment assignment	x+=y
-=	decrement assignment	x-=y
***=**	multiplication assignment	x*=y
%=	remainder assignment	x%=y
/=	division assignment	x/=y
//=	floor division assignment	x//=y

String operations

string[i]	retrieves character at position i
string[-1]	retrieves last character
string[i:j]	retrieves characters in range i to j

String methods

string.upper()	returns uppercase string
string.lower()	returns lowercase string
string.count(x)	counts how many times x appears
string.find(x)	position of the first occurrence of x
string.replace(x,y)	replaces x with y
string.islower()	returns True if all characters are lowercase
string.isupper()	returns True if all characters are uppercase
string.isalnum()	returns True if all characters are alphanumeric
string.isalpha()	returns True if all characters are alphabetic
string.isdigit()	returns True if all characters are digits
string.index(s)	returns index of substring s in string
string.strip(x)	returns a string with leading and trailing characters removed

List operations

list = []	defines an empty list
list[i] = x	stores x with index i
list[i]	retrieves the item with index i
list[-i]	retrieves last i item from list
list[i:j]	retrieves items in the range i to j
list[i:]	retrieves items from i to the end
del list[i]	removes the item with index i

List methods

list.append(x)	appends x to the end of the list
list.extend(L)	appends L to the end of the list
list.insert(i,x)	inserts x at i position
list.remove(x)	removes the first list item whose value is x
list.pop(i)	removes the item at position i and returns its value
list.clear()	removes all items from the list
list.index(x)	returns the position of the first occurrence of x in a list
list.count(x)	returns the number of times x appears in a list
list.sort()	sorts items in a list
list.reverse()	reverses list elements
list.copy()	returns a copy of the list

Legend: x, y = any data values; s = string; n = number; L = List

Built-in functions

print(x, sep='y')	prints x objects separated by y
input(s)	prints s and waits for an input that will be returned
len(x)	returns the length of x (s or L)
min(L)	returns the minimum value in L
max(L)	returns the maximum value in L
sum(L)	returns the sum of the values in L
range(n1,n2,n)	returns a sequence of numbers from n1 to n2 in steps of n
abs(n)	returns the absolute value of n
round(n1,n)	returns the n1 number rounded to n digits
type(x)	returns the type of x (string, float, list …)
str(x)	converts x to a string
list(x)	converts x to a list
int(x)	converts x to an integer
float(x)	converts x to a float
bool(x)	converts x to a Boolean value
pow(n1,n2)	returns n1 to the power of n2
chr(x)	returns the string value of a Unicode code
ord(x)	returns the Unicode code of a single-character string
sorted(L)	returns a new list with L items sorted
map(function, L)	applies function to values in L

Conditional statements

```
if <condition> :
        <code>
elif <condition> :
        <code>
 …
else:
        <code>

if <value> in <list>:
```

Loops

```
while <condition>:
        <code>

for <variable> in <list>:
        <code>

 for <variable> in range(start,stop,step):
        <code>
```

Loop control statements

break	finishes loop execution
continue	jumps to next iteration
pass	does nothing

Reading and writing files

```
f = open(<path>,'r')
f.read(<size>)
f.readline(<size>)
f.close()

f = open(<path>,'r')
for line in f:
        <code>
f.close()

f = open(<path>,'w')
f.write(<str>)
f.close()
```

Functions

```
def function(<params>):
        <code>
        return <data> or none
```

Modules

```
import module
module.function()

from module import *
function()
```

Legend: x, y = any data values; s = string; n = number; L = List