

Python 3 Beginner's Reference Cheat Sheet

Alvaro Sebastian
<http://www.sixthresearcher.com>

Main data types

boolean = *True / False*
integer = 10
float = 10.01
string = "123abc"
list = [value1, value2, ...]
dictionary = { key1:value1, key2:value2, ... }

Numeric operators

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

Comparison operators

== equal
!= different
> higher
< lower
>= higher or equal
<= lower or equal

Boolean operators

and logical AND
or logical OR
not logical NOT

Special characters

coment
\n new line
<char> scape char

String operations

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

List operations

list = [] defines an empty list
list[i] = x stores x with index i
list[i] retrieves the item with index i
list[-1] retrieves last item
list[i:j] retrieves items in the range i to j
del list[i] removes the item with index i

Dictionary operations

dict = {} defines an empty dictionary
dict[k] = x stores x associated to key k
dict[k] retrieves the item with key k
del dict[k] removes the item with key k

String methods

string.upper() converts to uppercase
string.lower() converts to lowercase
string.count(x) counts how many times x appears
string.find(x) position of the x first occurrence
string.replace(x,y) replaces x for y
string.strip(x) returns a list of values delimited by x
string.join(L) returns a string with L values joined by string
string.format(x) returns a string that includes formatted x

List methods

list.append(x) adds 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 a list of values delimited by x
list.count(x) returns a string with list values joined by S
list.sort() sorts list items
list.reverse() reverses list elements
list.copy() returns a copy of the list

Dictionary methods

dict.keys() returns a list of keys
dict.values() returns a list of values
dict.items() returns a list of pairs (key,value)
dict.get(k) returns the value associated to the key k
dict.pop() removes the item associated to the key and returns its value
dict.update(D) adds keys-values (D) to dictionary
dict.clear() removes all keys-values from the dictionary
dict.copy() returns a copy of the dictionary

Legend: **x,y** stand for any kind of data values, **s** for a string, **n** for a number, **L** for a list where **i,j** are list indexes, **D** stands for a dictionary and **k** is a dictionary key.

Python 3 Beginner's Reference Cheat Sheet

Alvaro Sebastian
<http://www.sixthresearcher.com>

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, L or D)
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, dict ...)
str(x)	converts x to string
list(x)	converts x to a list
int(x)	converts x to a integer number
float(x)	converts x to a float number
help(s)	prints help about x

Conditional statements

```
if <condition> :  
    <code>  
else if <condition> :  
    <code>  
...  
else:  
    <code>  
  
if <value> in <list>:
```

Data validation

```
try:  
    <code>  
except <error>:  
    <code>  
else:  
    <code>
```

Working with files and folders

```
import os  
os.getcwd()  
os.makedirs(<path>)  
os.chdir(<path>)  
os.listdir(<path>)
```

Loops

```
while <condition>:  
    <code>  
  
for <variable> in <list>:  
    <code>  
  
for <variable> in  
range(start,stop,step):  
    <code>  
  
for key, value in  
dict.items():  
    <code>
```

Loop control statements

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

Running external programs

```
import os  
os.system(<command>)
```

Functions

```
def function(<params>):  
    <code>  
    return <data>
```

Modules

```
import module  
module.function()  
  
from module import *  
function()
```

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()
```

Legend: x,y stand for any kind of data values, s for a string, n for a number, L for a list where i,j are list indexes, D stands for a dictionary and k is a dictionary key.