

### Data Types

Integer	-256, 15
Float	-253.23, 1.253e-10
String	"Hello", 'Goodbye', ""Multiline""
Boolean	True, False
List	[ <i>value</i> , ... ]
Tuple	( <i>value</i> , ... ) <sup>1</sup>
Dictionary	{ <i>key</i> : <i>value</i> , ... }
Set	{ <i>value</i> , <i>value</i> , ... } <sup>2</sup>

- <sup>1</sup> Parentheses usually optional  
<sup>2</sup> Create an empty set with set()

### Statements

#### If Statement

if *expression*:  
 statements  
 elif *expression*:  
 statements  
 else:  
 statements

#### While Loop

while *expression*:  
 statements

#### For Loop

for *var* in *collection*:  
 statements

#### Counting For Loop

for *i* in range(*start*, *end* [, *step*]):  
 statements  
 (*start* is included; *end* is not)

### Arithmetic Operators

$x + y$	add	$x - y$	subtract
$x * y$	multiply	$x / y$	divide
$x \% y$	modulus	$x ** y$	$x^y$

Assignment shortcuts:  $x op= y$   
 Example:  $x += 1$  increments  $x$

### Comparison Operators

$x < y$	Less	$x <= y$	Less or eq
$x > y$	Greater	$x >= y$	Greater or eq
$x == y$	Equal	$x != y$	Not equal

### Boolean Operators

not $x$	$x$ and $y$	$x$ or $y$
---------	-------------	------------

### Exception Handling

```
try:
  statements
except [ exception type [ as var ] ]:
  statements
finally:
  statements
```

### Conversion Functions

int( <i>expr</i> )	Converts <i>expr</i> to integer
float( <i>expr</i> )	Converts <i>expr</i> to float
str( <i>expr</i> )	Converts <i>expr</i> to string
chr( <i>num</i> )	ASCII char <i>num</i>

### String / List / Tuple Operations

len( <i>s</i> )	length of <i>s</i>
<i>s</i> [ <i>i</i> ]	<i>i</i> th item in <i>s</i> (0-based)
<i>s</i> [ <i>start</i> : <i>end</i> ]	slice of <i>s</i> from <i>start</i> (included) to <i>end</i> (excluded)
$x$ in <i>s</i>	<b>True</b> if <i>x</i> is contained in <i>s</i>
$x$ not in <i>s</i>	<b>True</b> if <i>x</i> is not contained in <i>s</i>
$s + t$	the concatenation of <i>s</i> with <i>t</i>
$s * n$	<i>n</i> copies of <i>s</i> concatenated
sorted( <i>s</i> )	a sorted copy of <i>s</i>
<i>s</i> .index( <i>item</i> )	position in <i>s</i> of <i>item</i>

### More String Operations

<i>s</i> .lower()	lowercase copy of <i>s</i>
<i>s</i> .replace( <i>old</i> , <i>new</i> )	copy of <i>s</i> with <i>old</i> replaced with <i>new</i>
<i>s</i> .split( <i>delim</i> )	list of substrings delimited by <i>delim</i>

### More String Operations (cont)

<i>s</i> .strip()	copy of <i>s</i> with whitespace trimmed
<i>s</i> .upper()	uppercase copy of <i>s</i>
See also	<a href="http://docs.python.org/library/stdtypes.html#string-methods">http://docs.python.org/library/stdtypes.html#string-methods</a>

### Mutating List Operations

del <i>lst</i> [ <i>i</i> ]	Deletes <i>i</i> th item from <i>lst</i>
<i>lst</i> .append( <i>e</i> )	Appends <i>e</i> to <i>lst</i>
<i>lst</i> .insert( <i>i</i> , <i>e</i> )	Inserts <i>e</i> before <i>i</i> th item in <i>lst</i>
<i>lst</i> .sort()	Sorts <i>lst</i>
See also	<a href="http://docs.python.org/library/stdtypes.html#typeseq-mutable">http://docs.python.org/library/stdtypes.html#typeseq-mutable</a>

### Dictionary Operations

len( <i>d</i> )	Number of items in <i>d</i>
del <i>d</i> [ <i>key</i> ]	Removes <i>key</i> from <i>d</i>
<i>key</i> in <i>d</i>	True if <i>d</i> contains <i>key</i>
<i>d</i> .keys()	Returns a list of keys in <i>d</i>
See also	<a href="http://docs.python.org/library/stdtypes.html#mapping-types-dict">http://docs.python.org/library/stdtypes.html#mapping-types-dict</a>

### Function Definitions

```
def name(arg1, arg2, ...):
  statements
  return expr
```

### Environment

sys.argv	List of command line arguments (argv[0] is executable)
os.environ	Dictionary of environment variables
os.getcwd()	String with path of current directory
import sys; print(sys.argv) or from sys import argv; print(argv)	



### String Formatting

```
"Hello, {0} {1}".format("abe", "jones")
Hello, abe jones
"Hello, {fn} {ln}".format(fn="abe", ln="jones")
Hello, abe jones
"You owe me ${0:,.2f}".format(253422.3)
You owe me $253,422.30
now = datetime.now()
'{:%Y-%m-%d %H:%M:%S}'.format(now)
2012-05-16 15:04:33
```

See also <http://docs.python.org/library/string.html#format-specification-mini-language>

### Useful Functions

<code>exit( code )</code>	Terminate program with <code>exit code</code>
<code>raw_input("prompt")</code>	Print <code>prompt</code> and <code>readline()</code> from <code>stdin</code> <sup>1</sup>

<sup>1</sup> Use `input("prompt")` in Python 3

### Code Snippets

#### Loop Over Sequence

```
for index, value in enumerate(seq):
    print("{} : {}".format(index, value))
```

#### Loop Over Dictionary

```
for key in sorted(dict):
    print(dict[key])
```

#### Read a File

```
with open("filename", "r") as f:
    for line in f:
        line = line.rstrip("\n") # Strip newline
        print(line)
```

### Other References

<http://rgruet.free.fr/>

*Great Python 2.x Quick Reference*

<http://www.cheatography.com/davechild/cheat-sheets/python/>

*More Python Cheatsheet Goodness*



By **sschaub**  
[cheatography.com/sschaub/](http://cheatography.com/sschaub/)

Published 21st May, 2012.  
Last updated 2nd June, 2014.  
Page 2 of 2.

Sponsored by **CrosswordCheats.com**  
Learn to solve cryptic crosswords!  
<http://crosswordcheats.com>