

Operators	
Assignment	=
Arithmetic	+, -, *, / , %
Comparison	>, >=, <, <=, ==, !=
Logical	not, and, or

String operations (string s)	
s.count(s ubstring)	Count occurrences
s.find (substring)	Index of first occurrence
s.join (sequence)	Concatenate sequence
s.split([delimiter])	Split into list

List operations (list l, element e)	
l.append(e)	Add e
l.remove(e)	Remove e
l.pop(e)	Remove and return e
l.count(e)	Count occurrences
l.reverse()	Reverse l
l.sort()	Sort l

Dictionary operations (dict d, key k)	
d.clear()	Clear d
d.get(k)	Return d[k]
d.keys()	Return keys in d
d.values()	Return values in d
d.items()	Return key-value pairs in d

File operations (file f)	
f = open(path)	Open f
f.read()	Read f
f.read_line()	Read line from f
f.read_lines()	Return list of lines in f
f.write(s)	Write s to f
f.close()	Close f

In-built functions		Useful standard library modules	
<code>int()</code> , <code>float()</code> , <code>str()</code> , <code>bool()</code> ...	Type casting	<code>math</code> , <code>numpy</code> , <code>scipy</code>	Math
<code>len(data)</code>	Length	<code>matplotlib</code>	Graph plotting
<code>min(values)</code> , <code>max(values)</code>	Minimum / Maximum	<code>random</code>	Random generators
<code>pow(x, y, [z])</code>	X to the power Y [mod Z]	<code>datetime</code>	Date and time
		<code>timeit</code>	Performance
		<code>re</code>	Regular expressions
<code>range(start, stop, [step])</code>	Ordered list	<code>os</code>	OS interaction
<code>input()</code> , <code>print()</code>	Console Input/-Output	<code>sys</code>	<code>stdin</code> , <code>stdout</code> , <code>stderr</code> , <code>version</code>
		<code>urllib</code>	Internet access
<code>filter(function, array)</code>	Filter array	<code>zlib</code>	Data compression
		Object-oriented	
<code>map(function, array)</code>	Map function onto array	<code>class Person:</code>	Class definition
		<code>x = Person(age, height)</code>	Object creation
<code>id(object)</code>	Unique object ID	<code>x.age</code>	Field access
<code>round(n, [x])</code>	Round n [x decimal places]	<code>x.birthday()</code>	Method access

Module import
<code>import module</code>
<code>from module import submodule</code>

<code>if(cond): <co de> else: <co de></code>	If-else
<code>if(cond1): <co de> elif(c ond1): <co de> else: <co de></code>	If-elif-else
<code>for i in range([st art], stop, [step]): <co de></code>	For loop over range
<code>for i in items: <co de></code>	For loop over iterable
<code>while(con dit ion): <co de></code>	While loop
<code>break</code>	Exit first enclosing loop
<code>continue</code>	Skip to next iteration



By **paul_benn**
cheatography.com/paul-benn/

Published 17th December, 2016.
 Last updated 10th August, 2016.
 Page 1 of 2.

Sponsored by **Readable.com**
 Measure your website readability!
<https://readable.com>