Cheatography

Python 3 Basics Cheat Sheet Cheat Sheet

by trashstuff via cheatography.com/34979/cs/10983/

Operators

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

String operations (string s)

`s.coun t(s ubs tring )`	Count occurences
`s.find (su bst ring)`	Index of first occurence
`s.join (se quence)`	Concatenate sequence
`s.plit ([d eil imi t er])`	Split into list

List operations (list l, element e)

`l.appe nd(e)`	Add e
`l.remove`	Remove e
`l.pop(e)`	Remove and return e
`l.count(e)`	Count occurences
`l.reve rse()`	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 file at path as 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

When using:
with open(path) as f:
the file gets opened as f and closes after leaving the "with" statement

Base functions

`int(), float(), str(), bool() ...`	Type casting
`len(data)`	Return length of data
`min(va lues), max(values)`	Minimum / Maximum
`pow(x, y, [z])`	x to the [mod z]
`range( start, stop, [step])`	Ordered list
`input(), print()`	Console input/ output
`filter (fu nction, iterable)`	Filter iterable
`map(fu nction, iterable)`	Map function onto iterable
`id(object)`	Unique object ID
`round(n, [x])`	Round n [x decimal places]

create your own functions with:
def functioname:

Control flow

List comprehensions

Type list comprehensions can be used to generate lists in a single line of code:

```
S = [x**2 for x in range(10)]
M = [x for x in S if x % 2 == 0]
noprimes = [j for i in range(2, 8) for j in range(i*2, 50, i)]
primes = [x for x in range(2, 50) if x not in noprimes]
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]
```

Module Import

`import module`	Import module
`import module as x`	Import module x
`from module import submodule`	Import specific submodule

`if(cond): <co de> else: <code>`	If-else statement
`if(cond): <co de> elif(c ond): <co de> else: <co de>`	If-elseif-else statement
`for i in range( [st art], stop, [step]): <co de>`	For loop over range
`for i in items: <co de>`	For loop over iterable
`while( cond): <co de>`	While loop
`break`	Exit loop
`continue`	Skip to next iteration

Object-oriented	
`class Person:`	Class definition
`x = Person (age, height )`	Object creation
`x.age`	Filed access
`x.birt hday()`	Method access

By **trashstuff**
cheatography.com/trashstuff/

Not published yet.
Last updated 23rd February, 2017.

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