Python (Basics) Cheat Sheet

by Guillermo (guilleaf) via cheatography.com/19177/cs/2233/

Built-In data types

None	NoneType	None object
bool	Boolean	True, False
int	Integer	-2, -1, 0, 1, 2
long	Long Integer	123456789-012345L
float	Floating point	3.1415
complex	complex	3 + 4*1j
str	String	'alice', 'bob'
unicode	Unicode String	u'alice', u'bob'
list	List	[1, 'two', 3.0]
tuple	Tuple	(1, 'two', 3.0)
dict	Dictionary	{'name': alice, 'age': 7}

Indentation matters

```
if Cheshire.is_visible:
        pri nt('I can see you')
        ali ce.t alks()
else:
        pri nt( "I can't see you")
```

Four spaces is the recommended by PEP 8 (Style Guide for Python Code)

String Literals

```
'alice'
" bob "
"The Hatter's Tea"
" " " Twi nkle, twinkle, little bat!
How I wonder what you're at!" " "
```

String Methods

s.lower()	Lowercased copy
s.upper()	Uppercased copy
s.isalpha()	True if alphabetic
s.isalnum()	True if alphanumeric
s.split(t)	Split s using t as separator
s.zfill(w)	Padding with leading zeros
s.strip()	Removes beg/end spaces

Lists

lst.append(x)	Append x to the end of lst
lst.count(x)	How many times x is in lst
lst.extend(itr) Same as lst += itr	Append all of iterable itr to lst
lst.index(x)	Index position of the first occurrence
lst.insert(i,x)	Insert item x into lst at index i
lst.pop()	Removes the last item in lst
lst.pop(i)	Removes item with index i
lst.remove(x)	Removes the first occurrence of x in lst
lst.reverse()	Reverse the list lst in-place
lst.sort()	Sorts lst in-place

List Comprehensions

```
>>> [2**x for x in range(10)]
[1, 2, 4, 8, 16, 32, 64, 128, 256, 512]
>>> [2**x for x in range(10) if x%2 == 0]
 [1, 4, 16, 64, 256]
```

Dictionaries

d.clear()	All items from dict d
d.copy()	Shallow copy of dict d
d.keys()	Read-only iterable of all keys
d.values()	Read-only iterable of all values
d.items()	Read-only iterable of all (key, value)
d.pop(k)	Removes (k, value) and returns d[k]

Conditionals

```
if x > 0 and y > 0:
        pri nt( 'First Quadrant')
elif x < 0 and y > 0:
        pri nt( 'Second Quadrant')
elif y < 0:
        pri nt( 'Third and Fourth')
else:
        pri nt('On the axis')
```

Loops (for)

```
a=1
b=2
for i in range(10):
        c=a+b
        print c
       a=b; b=c
```

Loops (while)

```
import random
x=1.0
while True:
        x*= ran dom.ra ndom()
        print x
       if x<0.5:
```



By **Guillermo** (guilleaf)
cheatography.com/guilleaf/
bit.ly/guilleaf

Not published yet.
Last updated 11th May, 2016.

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

Python (Basics) Cheat Sheet

by Guillermo (guilleaf) via cheatography.com/19177/cs/2233/

Loops (while) (cont)

```
>       x*=3
        continue
    if x>0.8:
        break
```

You can use 'break' to escape from the inner loop and 'continue' to jump to the next item in the loop

Classes

```
class Quaternion(object):
        def __init __( self,
four_t uple):
                    sel f.v alu -
es= fou r_tuple
         @pr operty
        def real(s elf):
                    return self.v -
alu es[0]
```

Special Methods

__bool__(self)	True of False return
__init__(self, args)	Object Initialization
__hash__(self)	To be use as key
__repr__(self)	String eval(repr(x))==x
__str__(self)	Human readable string

Comparison Special Methods

__lt__(self, other)	<
__le__(self, other)	<=
__gt__(self, other)	>
__ge__(self, other)	>=
__eq__(self, other)	==
__ne__(self, other)	!=

os Module

os.listdir(path)	Equivalent to 'ls'
os.getcwd()	Current Working Directory (pwd)
os.mkdir(path)	Make directory (mkdir)
os.chdir(path)	Change directory (cd)
os.path.isfile(-path)	Check if path is a file
os.path.isdi-r(path)	Check if path is directory
os.path.exis-ts(path)	Check existence as file or directory

sys Module

sys.argv	List of arguments
sys.version	Python version
sys.platform	OS platform
sys.maxint	Max integer
sys.stdout	Standard output
sys.stderr	Standard error

itertools Module

product('ABCD', repeat=2)

AA AB AC AD BA BB BC BD CA CB CC CD DA DB DC DD

permutations('ABCD', 2)

AB AC AD BA BC BD CA CB CD DA DB DC

combinations('ABCD', 2)

AB AC AD BC BD CD

combinations_with_replacement('ABCD', 2)

AA AB AC AD BB BC BD CC CD DD

math Module

sin, cos, tan	Trigonometrical
asin, acos, atan	Inv. Trigonometrical
sinh, cosh, tanh	Hyperbolic
asinh, acosh, atanh	Inverse Hyperbolic
ceil, floor, trunc	Truncation
log, log10	Logarithms
sqrt, exp, pow	Exponentials
degrees, radians	Angular conversion
pi, e	Math Constants

json Module

```
>>> import json
>>> dic={' nam e': 'Al ice',
'age': 7.5}
>>> dic_js on= jso n.d ump -
s(dic,
                                -
          sor t_k eys =True,
                                -
          ind ent=4,
                                -
          sep ara tor s=(',', ':
'))
>>> print dic_json
{
        " age ": 7.5,
        " nam e": " Ali ce"
}
>>> dic2=j son.lo ads (dic)
>>> dic2
  {u' age': 7.5, u'name':
u'Alice'}
```



By **Guillermo** (guilleaf)
cheatography.com/guilleaf/
bit.ly/guilleaf

Not published yet.
Last updated 11th May, 2016.

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