Python3 Quick Reference Sheet
Version 1.0.1 by Joff Thyer

Strings

UTF8 (default) encoded string object

```
myvar = "This is a string"
myvar = 'This is a string'
len(myvar) ⇒ 16
```

Byte string object (ASCII only)

```
myvar = b"Byte obj str"
myvar = "Byte obj str".encode()
```

Decode Byte object to String (UTF8 default)

```
myvar = b'decode this'.decode()
```

String Prefixes

```
myvar = r'this is a raw string'
myvar = b'this is a byte encoded string'
myvar = f'quick format str with var {a}'
** Raw string will not encode escaped chars.
```

String Methods

```
myvar = 'some string'
myvar.upper() ⇒ 'SOME STRING'
myvar.lower() ⇒ 'some string'
myvar.title() ⇒ 'Some String'
myvar.replace('s','x') ⇒ 'xome xtring'
myvar.replace('s','x', 1) ⇒ 'xome string'
```

String Split/Join

```
'hello world'.split() ⇒ ['hello', 'world']
':'.join(['hello', 'world']) ⇒ 'hello:world'
```

Lists

Empty list	myvar = []
Append to list	myvar.append(23) ⇒ [23]
Insert into list	myvar.insert('chuck', 0') ⇒ ['chuck', 23]
Remove from list	myvar.remove(23) ⇒ ['chuck']
Find in list	myvar.index('chuck') ⇒ 0
Length of list	len(myvar) ⇒ 1

Dictionaries

Empty dictionary	myvar = {}
Initialize dictionary	myvar = {'j': 'jim', 3: 'three'}
Assign key/value pair	myvar['z'] = 'zulu'
Access Specific Value	myvar['z'] myvar.get('z')
Does key exist?	'z' in myvar ⇒ True
Get view of keys	myvar.keys()
Get view of values	myvar.values()
Get view of keys and values	myvar.items()

Data Conversion

Convert to string	str(99)	'99'
Convert to integer	int("99")	99
Conver to integer from base16	int("FF", 16)	255
Hexadecimal String	hex(255)	'0xff'
Binary String	bin(255)	'0b11111111'
Convert float to integer	int(2.134)	2
Ordinal value of character	ord('z')	122
	ord('😀')	128512
Character from integer	chr(122)	'z'
	chr(128512)	'😀'

Slicing (y = "Python3 is the coolest")

** Syntax is y[start:end:step]*
*** end is not inclusive*
*** slicing work on lists or strings*

y[-1]	't'
y[0]	'P'
y[0:4]	'Pyth'
y[0:8:2]	'Pto3'
y[:5]	'Pytho'
y[5:]	'n3 is the coolest'
y[::-1]	'tselooc eht si 3nohtyP'
y[::-2]	'teocets nhy'

Math Operations (consider x=100, and y=22)

Addition	x + y	122
Subtraction	x - y	78
Multiplication	x * y	2200
Division	x / y	4.545454
Integer Division (Floor)	x // y	4
Modulus (remainder after div)	x % y	12
Shift Left	x << 2	400
Shift Right	x >> 2	25
Exclusive OR (XOR)	x ^ y	114
Raise to the power of	x ** 2	10000

Logical Comparison / Operators

Equality	x == y	False
Inequality	x != y	True
Less than	x < y	False
Less than or equal to	x <= y	False
Greater than	x > y	True
Greater than or equal to	x >= y	True
Logical AND	True and False	False
	True and True	True
Logical OR	True or False	True
	True or True	True
Logical NOT	not False	True

If, then, and else code blocks (x = 1, y = 5)

```
# code blocks indented by 4 spaces
# more than one statement can execute in a code block
if x > 1:
    print('x is greater than 1')
    print('I promise it is')
elif x <= 1 and y > 5:
    print('x is <= 1 and y > 5')
else:
    print('I am confused')
    print('really I am so confused')
```

Loops (consider x='Cool', y=['my', 'list'])

** range() function generates an interator of numbers*
** iterator variable (after 'for' is arbitrary choice)*
** enumerate() function produces python tuple*
** code block is execute multiple times*
** 'while' loops can loop forever!*
** 'continue' statement goes to next loop iteration*
** 'break' statement breaks immediately out of a loop*

loop from 1 to 9	for i in range(1,10):
loop from 0 to 99	for i in range(100):
loop through string characters	for cr in x:
loop through list items	for li in y:
keep track of index	for i, cr in enumerate(x):
while loop that might loop forever	while len(y) == 2:

Format String Examples

```
'The sum of {} + {} = {}'.format(1,2,3)
'{0:d} / {1:d} = {2:4.2f}'.format(x, y, x / y)
f'{x:d} / {y:d} = {x / y:4.2f}'
```

Miscellanous Items

input() used to accept keyboard input	input('please enter your name')
sum() adds up list of numbers	sum([1,2,3]) 6
map() maps function onto iterable	list(map(ord, "123")) [49, 50, 51]

```
str() casts/converts to string object
list() casts/converts to list object
dict() casts/converts to dictionary object
'del <varname>' can be used to delete any object
abs() takes absolute value
round() can be used for rounding up to a precision
bool() performs logical evalation
type() shows object type
dir() lists object attributes and methods
```