

Python Quickref

	<i>Syntax</i>	<i>Display</i>
Arrays	<pre>t=[1,7,'aa'] t[0] t[0:2] t[1:] t[-1] t[-2:] t=[1] t.append([2,4]) t+[3,5] len(t) range(4) range(2,5) range(2,11,3)</pre>	<pre>1 [1,7] [7,'aa'] 'aa' [7,'aa'] [1,[2, 4]] [1,[2, 4],3,5] 2 [0,1,2,3] [2,3,4] [2,5,8]</pre>
Strings	<pre>'aa'+ 'bb' 12+'aa' str(12)+'aa' 4*'a' s='abcd' s[1] s[1:2] s[1:] s[-1] s[-2:] len(s) s="aa,bb,cc" li=s.split(',') print li '/'.join(li) s.startswith('aa')</pre>	<pre>'aabb' TypeError: ... '12aa' 'aaaa' 'b' 'bc' 'bcd' 'd' 'cd' 4 ['aa','bb','cc'] 'aa/bb/cc' True</pre>
Dictionaries	<pre>d={'a':2,'b':6} len(d) d.keys() d.values() d['a'] d.get('a',None) d['c']=12 d.get('c',None) del d['a'] d.get('a',None) d</pre>	<pre>2 ['a','b'] [2,6] 2 2 12 None {'b':6,'c':12}</pre>
Special types	<pre>None True False</pre>	
If	<pre>x, y = 2, 4 if x < 0: print 'Neg.' elif x == 0: print 'Zero' else: print 'More' if x>0 and y>0: print "Both positive" if []: print "Not empty" else: print "Empty" if x in [1,2,3]: print "Small"</pre>	<pre>More Both positive Empty Small</pre>

Looping	<pre>li=[1,2,3] for i in li: if i==2: pass else: print li</pre>	<pre>1 3</pre>
	<pre>di={'a':3,'b':5} for k in di: print "d['%s']=%s" \ % (k,k[i])</pre>	<pre>d['a']=3 d['b']=5</pre>
	<pre>i=8 while i>1: print i i/=2</pre>	<pre>8 4 2</pre>
Exceptions	<pre>try: c = a/b except: print "Can't divide"</pre>	

Functions

```
def sum(a,b):
    """Return the sum"""
    return a+b

def sdiff(a,b)
    return a+b,a-b

print sum(4,3)
s,d = sumdiff(4,3)

print "%s,%s" % (s,d)

def affine(x,a=3,b=2):
    return a*x+b

print affine(4)
print affine(4,2,3)
print affine(4,2,b=3)
print affine(4,b=3)
print affine(4,a=2,3)
```

7
7,1
14
11
11
15
Error

Classes

```
class MyCalc:
    """Documentation"""
    def __init__(self,x=0,y=0):
        self.x = x
        self.y = y

    def sum(self,z=0):
        return self.x+self.y+z

calc = MyCalc(3,2)
print calc.sum()
print calc.x
hasattr(c,'x')
c.z
getattr(c,'z','rien')

dosum = c.sum
dosum(4)
```

5
3
True
Error
'rien'
9

Modules

```
import sys
from os import getenv

print argv[0]
print sys.argv[0]
print getenv('HOME')
```

Error
name
/home/...