Python Cheat Sheet

Base Types

integer, float, boolean, string, bytes

int	163	0	−192	$\underbrace{\text{0b110}}_{\text{binary}}$	$\underbrace{\text{0x3F}}_{\text{hex}}$
float	9.32	0.0	$-1.7\underbrace{\text{E}-6}_{\times 10^{-6}}$		
bool	$\underbrace{\text{False}}_{0}$	$\underbrace{\text{True}}_{1}$			
str	'some text' or "some text"				
bytes	b"text\xfe\775"				

Container Types

ordered containers — repeatable values

list	[1,5,3]	["a",1,5,5]	[5]	[]
tuple	(1,5,3)	"a",1,5,5	(5,)	()
str	"153"			""

Immutable (non-modifiable values)

key containers — no order, unique keys

set	{"key1","key2"}	{1,9,3,0}	**set**()
dict	{"key1":value1,"key2":value2}	**dict**(a=3,b="v")	{}

Integer Sequences

range([start,] end [,step])

start default is 0 (inclusive), end (exclusive), step default is 1.

range(5) → 0,1,2,3,4
range(2,5) → 2,3,4
range(2,12,3) → 2,5,8,11
range(20,5,−5) → 20,15,10

Operations on Sets

Operators

|	.union
&	. intersection
−	. difference

Methods

s.add(key)   s.update(s2)
s. clear ()   s.remove(key)

Operations on Lists

Methods

a.append(value)   a.extend(a2)
s. insert (idx,value)   a.pop()

Conversions

type(expression)

int('153') → 15	
int('3f',16) → 63	(Specify base in 2nd parameter)
int(−11.24e8) → −1124000000	
int(15.56) → 15	(Truncate decimal point)
round(15.58,1) → 15.6	(Round to 1 decimal place)
float('15.56') → 15.56	
bool(x)	(False for None, zero or empty containers)
str(x)	(String representation of x.)

chr(65) → 'A'   code ↔ char   **ord**('A') → 65

list ('abc') → ['a','b','c']
dict ([(3,'three'), (1,'one')]) → {3:'three', 1:'one'}
set(['one','two']) → {'one','two'}

(Split string using a separator, **str** → **list** of **str**)
'random:data:666'.split(':') → ['random', 'data', '666']

(Join a list of strings, **list** of **str** → **str**)
':'.join(['random', 'data', '666']) → 'random:data:666'

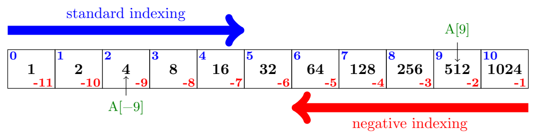
(Convert each element in a collection)
[**int**(x) **for** x **in** ['1','29','−3']] → [1, 29, −3]

Generic Operations on Containers

min(c)   **max**(c)   **sum**(c)   **sorted**(c)
len(c)   (Number of elements in collection c)

all(c) → True if **all** items in c evaluate to True, else False.
any(c) → True if **at least one** item in c evaluate to True, else False.

Sequence Containers Indexing



a[3:6] → [8, 16, 32]
a[1:−1] → [2, 4, 8, 16, 32, 64, 128, 256, 512]
a[::−1] → [1024, 512, 256, 128, 64, 32, 16, 8, 4, 2, 1]

Looping over Collections

(Loop over values)
```
for value in A:
   print(value)
```

(Count and loop over values)
```
for k,value in enumerate(A):
   print(k, value)
```

(While loop)
```
k = 0
while k<len(A):
   print(k, A[k])
   k += 1
```
Initialisation **before** loop.
update **within** loop.

break immediatly exits loop. **continue** skips to next iteration.
else block for normal loop exit.