

Python for Beginners – Cheat Sheet

Data types and Collections

integer	10
float	3.14
boolean	True/False
string	'abcde'
list	[1, 2, 3, 4, 5]
tuple	(1, 2, 'a', 'b')
set	{'a', 'b', 'c'}
dictionary	{'a':1, 'b':2}

Operations

Index starts at 0

Strings:

s[i]	i:th item of s
s[-1]	last item of s

Lists:

l = []	define empty list
l[i:j]	slice in range i to j
l[i] = x	replace i with x
l[i:j:k]	slice range i to j, step k

Dictionaries:

d = {}	create empty dictionary
d[i]	retrieve item with key i
d[i] = x	store x to key i
i in d	is key i in dictionary

Numerical Operators

+	addition
-	subtraction
*	multiplication
/	division
**	exponent
%	modulus
//	floor division

Comparison Operators

<	less
<=	less or equal
>	greater
>=	greater or equal
==	equal
!=	not equal

List Methods

l.append(x)	append x to end of list
l.insert(i, x)	insert x at position i
l.remove(x)	remove first occurrence of x
l.reverse()	reverse list in place

Dictionary Methods

d.keys()	returns a list of keys
d.values()	returns a list of values
d.items()	returns a list of (key, value)

Logical Operators

and	logical AND
or	logical OR
not	logical NOT

Membership Operators

in	value in object
not in	value not in object

Conditional Statements

```
if condition:  
    <code>  
  
elif condition:  
    <code>  
  
else:  
    <code>
```

String Methods

s.strip()	remove trailing whitespace
s.split(x)	return list, delimiter x
s.join(l)	return string, delimiter s
s.startswith(x)	return True if s starts with x
s.endswith(x)	return True if s ends with x
s.upper()	return copy, uppercase only
s.lower()	return copy, lowercase only

Import from Module

```
from module import func      import func  
from module import func as f import func as f
```

Python for Beginners – Cheat Sheet

Built-in Functions

<code>float(x)</code>	convert x to float
<code>int(x)</code>	convert x to integer
<code>str(x)</code>	convert x to string
<code>set(x)</code>	convert x to set
<code>type(x)</code>	returns type of x
<code>len(x)</code>	returns length of x
<code>max(x)</code>	returns maximum of x
<code>min(x)</code>	returns minimum of x
<code>sum(x)</code>	returns sum of values in x
<code>sorted(x)</code>	returns sorted list
<code>round(x, d)</code>	returns x rounded to d
<code>print(x)</code>	print object x

Loops

`while condition:`
 `<code>`

`for var in list:`
 `<code>`

Control statements:

<code>break</code>	terminate loop
<code>continue</code>	jump to next iteration
<code>pass</code>	does nothing

String Formatting

```
"Put {} into a {}".format("values", "string")
'Put values into a string'

"Put whitespace after:{:<10}, or before:{:>10}".format("a","b")
'Put whitespace after: a      , or before:      b'

"Put whitespace around:{:^10}".format("c")
'Put whitespace around:    c    .'
```

Regular Expressions

```
import re
p = re.compile(pattern)  compile search query
p.search(text)           search for all matches
p.sub(sub, text)         substitute match with sub
```

.	any one character
*	repeat previous 0 or more times
+	repeat previous 1 or more times
?	repeat previous 0 or 1 times
\d	any digit
\s	any whitespace
[abc]	any character in this set {a, b, c}
[^abc]	any character *not* in this set
[a-z]	any letter between a and z
a b	a or b

Reading and Writing Files

```
fh = open(<path>,'r')
for line in fh:
  <code>
fh.close()

out = open(<path>,'w')
out.write(<str>)
out.close()
```

Functions

```
def Name(param1, param2 = val):
  <code>
  #param2 optional, default: val
  return <data>
```

sys.argv

```
import sys      import module
sys.argv[0]     name of script
sys.argv[1]     first cmd line arg
```