# Cheatography

# Python 3 Cheat Sheet

by monika.g via cheatography.com/175674/cs/36853/

## File operations

| | |
|---|---|
| `f = open(path)` | open file |
| `f.read()` | read file |
| `f.read line()` | read line from file |
| `f.read lines()` | return list of lines of file |
| `f.write(s)` | write s to file |
| `f.close()` | close file |
| `with open(path) as name:` *statements* | open and after statements close file |

## Logic and Math Operators

| | |
|---|---|
| `x + y` | addition |
| `x - y` | subtraction |
| `x * y` | multiplication |
| `x / y` | division |
| `x // y` | floor |
| `x ** y` | exponent |
| `x % y` | modulo |
| `x == y` | equal |
| `x != y` | not equal |
| `x > y` | greater than |
| `x >= y` | greater than or equal |
| `x < y` | less than |
| `x <= y` | less than or equal |

## Converting Data Types

| | |
|---|---|
| `str(num)` | number to string |
| `str(txt, encoding)` | encoded bytes to string |
| `int("st r", base = 10)` | number string to integer |
| `hex(int)` | integer to hex string |

## Converting Data Types (cont)

| | |
|---|---|
| `bin(int)` | integer to binary string |
| `int(float)` | float to integer |
| `float(int or str)` | integer / string to float |
| `ord(str)` | string to ASCII |
| `chr(int)` | integer to ASCII |

## Base Functions

| | |
|---|---|
| `int(), float(), str(), bool() ...` | type casting |
| `len(data)` | return length of data |
| `min(va lue), max(value)` | minimum<br>maximum |
| `pow(x, y, [z])` | x to the power y [mod z] |
| `range( start, stop, [step])` | range of |
| `input(), print()` | console input<br>console output |
| `filter (fu nction, iterable)` | filter iterable |
| `id(object)` | unique object ID |
| `map(fu nction, iterable)` | map function onto iterable |
| `round(n, [x])` | round n to x decimal places |

[ _ ] optional argument

## Module Import

```
import module as name
```

```
from module import submodule as name
```

## Exception Handling

```
try:
 statements
except [exception type]:
 statements
finally:
 statements
```

## Dictionaries

| | |
|---|---|
| `d = {}` | create dictionary |
| `d = { key : val }` | create dictionary with values |
| `d[key] = val` | assign a value |
| `d[key]`, `d.get(key)` | access value at key |
| `d.keys()` | iterable view of keys |
| `d.values()` | iterable view of values |
| `d.items()` | iterable view of (key, value) tuples |
| `d.clear()` | clear dictionary |
| `key in d` | determine if key exists |

## Lists

| | |
|---|---|
| `n = []` | create list |
| `n[index] = val` | assign value at index |
| `n[index]` | access value at index |
| `n.appe nd(val)` | add to list |
| `n.inse rt( pos ition, val)` | insert into list |
| `n.coun t(val)` | count occurness |
| `n.remo ve(val)` | remove item |
| `n.pop(val)` | remove an return item |
| `n.reve rse()` | reverse n |
| `n.sort()` | sort n |

## Slicing and Indexing

| **x[start:stop:step]** | **x = [4, 8, 9, 3, 0]** |
|---|---|
| `x[0]` | 4 |
| `x[-1]` | 0 |
| `x[:3]` | [4, 8, 9] |
| `x[3:]` | [3, 0] |
| `x[:-2]` | [4, 8, 9] |
| `x[::2]` | [4, 9, 0] |
| `x[::-1]` | [0, 3, 9, 8, 4] |