

Cheat Sheet

Python

Control Flow

```
while condition:
    statements

for var in some_list:
    statements

break # Exit the loop

continue # Next iteration

if condition1:
    statements
elif condition2:
    statements
else:
    statements

def func_or_method(args):
    statements

try:
    statements
except SomeException:
    # Catch the SomeException
    statements
```

Strings

```
some_string.lower()
some_string.upper()
# some_string to lower/upper case

s.join(some_list)
# string formed by concatenating
# some_list with s as separator
```

```
string_a + string_b
# concatenate two strings

s.find(char)
# Returns index of char if found
# -1 otherwise
```

Lists

```
# Check if a exists in some_list
if a in some_list:
    statements

some_list.count(a)
# Returns the number of times
# a is in some_list

some_list.append(a)
# Append a to some_list
```

Classes

```
# Access property_name as an attribute
@property
def property_name(self):
    statements

# Called when assigning a value
# to property_name
@property_name.setter
def property_name(self, value):
    statements
```

Abstract classes

```
import ABC, abstractmethod

# A is will be an abstract class
class A(ABC):

    # abstract method
    @abstractmethod
    def some_method(self, args):
        statements
```

Pytest

```
import pytest

# function starts with test_
# picked up automatically by pytest
def test_something():
    assert boolean_expression

# Test will pass if SomeException is raised
# by the statements
def test_exception():
    with pytest.raises(SomeException):
        statements
```