

# Python Keywords Cheat Sheet

## False

Used to represent the boolean value False.

Example:

```
if False:  
    print('Will not execute')
```

## None

Represents the absence of a value.

Example:

```
x = None
```

## True

Used to represent the boolean value True.

Example:

```
if True:  
    print('Always executes')
```

## and

Logical AND operator.

Example:

```
if True and True:  
    print('Both are True')
```

## as

Used for aliasing modules.

Example:

```
import numpy as np
```

## assert

Used for debugging.

Example:

```
assert 2 + 2 == 4
```

## async

Defines asynchronous functions.

Example:

```
async def fetch():  
    pass
```

## await

Waits for async operation.

Example:

```
await fetch()
```

# Python Keywords Cheat Sheet

## **break**

Exits a loop.

Example:

```
for i in range(5):  
    break
```

## **class**

Defines a class.

Example:

```
class Dog:  
    pass
```

## **continue**

Skips to the next loop iteration.

Example:

```
for i in range(3):  
    continue
```

## **def**

Defines a function.

Example:

```
def greet():  
    print('Hello')
```

## **del**

Deletes a variable or item.

Example:

```
del my_list[0]
```

## **elif**

Else if condition.

Example:

```
if x < 0:  
    pass  
elif x == 0:  
    pass
```

## **else**

Defines block when if/elif fail.

Example:

```
if x > 0:  
    pass  
else:
```

# Python Keywords Cheat Sheet

**pass**

## **except**

Handles exceptions.

Example:

```
try:  
    pass  
except:  
    print('Error')
```

## **finally**

Runs after try/except.

Example:

```
finally:  
    print('Done')
```

## **for**

Used for looping.

Example:

```
for i in range(3):  
    print(i)
```

## **from**

Used for importing.

Example:

```
from math import pi
```

## **global**

Declares global variable.

Example:

```
global x
```

## **if**

Conditional statement.

Example:

```
if x > 0:  
    pass
```

## **import**

Imports a module.

Example:

```
import os
```

## **in**

# Python Keywords Cheat Sheet

Membership test.

Example:

```
if 3 in [1, 2, 3]:  
    print('Found')
```

## is

Identity comparison.

Example:

```
if x is None:  
    print('None')
```

## lambda

Creates anonymous function.

Example:

```
x = lambda a: a + 10
```

## nonlocal

Refers to parent-scope variable.

Example:

```
nonlocal x
```

## not

Logical NOT operator.

Example:

```
if not True:  
    pass
```

## or

Logical OR operator.

Example:

```
if True or False:  
    pass
```

## pass

Empty statement.

Example:

```
pass
```

## raise

Raises an exception.

Example:

```
raise ValueError('error')
```

## return

# Python Keywords Cheat Sheet

**Exits a function with value.**

Example:

```
return x
```

## try

Tries a block of code.

Example:

```
try:  
    x = 1
```

## while

Loop while condition is true.

Example:

```
while True:  
    break
```

## with

Context manager usage.

Example:

```
with open('file.txt') as f:
```

## yield

Used in generators.

Example:

```
yield x
```

## match

Structural pattern matching.

Example:

```
match x:  
    case 1:  
        pass
```

## case

Used inside match block.

Example:

```
case 2:  
    pass
```