Python 3 Cheat Sheet

October 22nd, 2018
Comments

Update (Nov 19 2018): Added exceptions and classes.

I've created this Python 3 cheat sheet to help beginners remember Python language syntax. You can also download this cheat sheet as a beautiful PDF here.

NOTE: This cheat sheet is a work in progress and is not complete yet. I'll be adding new stuff to it over the next few weeks. So, be sure to come back and get the latest version.

If you're starting out with Python and are looking for a fun and comprehensive tutorial, check out my YouTube tutorials. I have two Python tutorials. If you have no or little programming experience, I suggest you check out my Python tutorial for beginners. Otherwise, if you know the basics (eg variables, functions, conditional statements, loops) and are looking for a tutorial that gets straight to the point and doesn't treat you like a beginner, check out my Python tutorial for programmers.

If you enjoy this post, please spread the love by sharing this post with others.

Variables

```
a = 1        # integer
b = 1.1      # float
c = 1 + 2j   # complex number (a + bi)
d = "a"      # string
e = True     # boolean (True / False)
```

Strings

```python
x = “Python”
len(x)
x[0]
x[-1]
x[0:3]

# Formatted strings
name = f”{first} {last}”

# Escape sequences
\” \’ \\ \n

# String methods
x.upper()
x.lower()
x.title()
x.strip()
x.find(“p”)
x.replace(“a”, “b”)
“a” in x
```

Type Conversion

```python
int(x)
float(x)
bool(x)
string(x)
```

Falsy Values

```python
0
“”
[]
```

Conditional Statements

```python
if x == 1:
    print(“a”)
elif x == 2:
    print(“b”)
else:
    print(“c”)

# Ternary operator
x = “a” if n > 1 else “b”

# Chaining comparison operators
if 18 <= age < 65:
```

Loops

```python
for n in range(1, 10):
    print(n)

while n < 10:
    print(n)
    n += 1
```

Functions

```python
def increment(number, by=1):
    return number + by

# Keyword arguments
increment(2, by=1)

# Variable number of arguments
def multiply(*numbers):
    for number in numbers:
        print number


multiply(1, 2, 3, 4)

# Variable number of keyword arguments
def save_user(**user):
    ...


save_user(id=1, name="Mosh")
```

Lists

```python
# Creating lists
letters = ["a", "b", "c"]
matrix = [[0, 1], [1, 2]]
zeros = [0] * 5
combined = zeros + letters
numbers = list(range(20))

# Accessing items
letters = ["a", "b", "c", "d"]
letters[0]  # "a"
letters[-1] # "d"

# Slicing lists
letters[0:3]   # "a", "b", "c"
letters[:3]    # "a", "b", "c"
letters[0:]    # "a", "b", "c", "d"
letters[:]     # "a", "b", "c", "d"
letters[::2]   # "a", "c"
letters[::-1]  # "d", "c", "b", "a"

# Unpacking
first, second, *other = letters

# Looping over lists
for letter in letters:
    ...

for index, letter in enumerate(letters):
    ...

# Adding items
letters.append("e")
letters.insert(0, "-")

# Removing items
letters.pop()
letters.pop(0)
letters.remove("b")
del letters[0:3]

# Finding items
if "f" in letters:
    letters.index("f")

# Sorting lists
letters.sort()
letters.sort(reverse=True)

# Custom sorting
items = [
    ("Product1", 10),
    ("Product2", 9),
    ("Product3", 11)
]

items.sort(key=lambda item: item[1])

# Map and filter
prices = list(map(lambda item: item[1], items))
expensive_items = list(filter(lambda item: item[1] >= 10,

# List comprehensions
prices = [item[1] for item in items]
expensive_items = [item for item in items if item[1] >= 1

# Zip function
list1 = [1, 2, 3]
list2 = [10, 20, 30]
combined = list(zip(list1, list2))    # [(1, 10), (2, 20)
```

Tuples

```python
point = (1, 2, 3)
point(0:2)     # (1, 2)
x, y, z = point
if 10 in point:
    ...

# Swapping variables
x = 10
y = 11
x, y = y, x
```

Arrays

```python
from array import array

numbers = array("i", [1, 2, 3])
```

Sets

```python
first = {1, 2, 3, 4}
second = {1, 5}

first | second  # {1, 2, 3, 4, 5}
first & second  # {1}
first - second  # {2, 3, 4}
first ^ second  # {2, 3, 4, 5}

if 1 in first:
    ...
```

Dictionaries

```python
point = {"x": 1, "y": 2}
point = dict(x=1, y=2)
point["z"] = 3
if "a" in point:
    ...
point.get("a", 0)   # 0
del point["x"]
for key, value in point.items():
   ...

# Dictionary comprehensions
values = {x: x * 2 for x in range(5)}
```

Generator Expressions

```python
values = (x * 2 for x in range(10000))
len(values)  # Error
for x in values:
```

Unpacking Operator

```python
first = [1, 2, 3]
second = [4, 5, 6]
combined = [*first, "a", *second]

first = {"x": 1}
second = {"y": 2}
combined = {**first, **second}
```

Exceptions

```python
# Handling Exceptions
try:
  …
except (ValueError, ZeroDivisionError):
  …
else:
  # no exceptions raised
finally:
  # cleanup code

# Raising exceptions
if x < 1:
    raise ValueError("…")

# The with statement
with open("file.txt") as file:
   …
```

Classes

```python
# Creating classes
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def draw(self):
        …

# Instance vs class attributes
class Point:
    default_color = "red"

    def __init__(self, x, y):
        self.x = x

# Instance vs class methods
class Point:
    def draw(self):
        …

    @classmethod
    def zero(cls):
        return cls(0, 0)

```

```python

# Magic methods
__str__()
__eq__()
__cmp__()
...

# Private members
class Point:
    def __init__(self, x):
        self.__x = x


# Properties
class Point:
    def __init__(self, x):
        self.__x = x

    @property
    def x(self):
        return self.__x

    @property.setter:
    def x.setter(self, value):
        self.__x = value

# Inheritance
class FileStream(Stream):
    def open(self):
         super().open()
         …

# Multiple inheritance
class FlyingFish(Flyer, Swimmer):
    …

# Abstract base classes
from abc import ABC, abstractmethod

class Stream(ABC):
    @abstractmethod
    def read(self):
        pass

# Named tuples
from collections import namedtuple

Point = namedtuple("Point", ["x", "y"])
point = Point(x=1, y=2)
```

Mosh

Hi! My name is Mosh Hamedani. Over the last three years, I've taught over 1M students how to code or how to become a better coder through my online courses and YouTube channel. I've been working as a software engineer for the past 18 years and I love to share my knowledge with you. It's my mission to make coding accessible to everyone.

Share this:

Related

53 Python Exercises and Questions for Beginners
October 22, 2018
In "Python"

My Brand New Python Tutorial (6 Hours)
February 18, 2019
In "Python"

4 Common Mistakes with the Repository Pattern
April 3, 2017
In "C# / .NET"

Tags: cheat sheet, python

8 responses to "Python 3 Cheat Sheet"

Brian says:
November 18, 2018 at 12:34 am

Thank you Mosh!!

Reply

Mojtaba Jahani says:
November 20, 2018 at 12:27 am

Hi and thanks for this awesome course.
I think there is a typo in Type Conversion part of Python cheat sheet for string conversion.
str(x) instead of string(x)

Reply

Python For Loops – Python Tutorial for Absolute Beginners | Coding Videos says:
December 30, 2018 at 12:07 am

[...] Python 3 Cheat Sheet [...]

Reply

chakshudiwan says:
February 12, 2019 at 9:16 am

Great Cheatsheet. Just had a glance at Tuples Block.
Use Square brackets for print.

Thankyou for sharing.

Reply

antiViruss says:
February 16, 2019 at 8:31 am

thanks sir now i will be enjoying the learning

Reply

Hassan Eman says:
February 18, 2019 at 3:26 pm

you are very good teacher MOSH

Reply

Trahloc says:
March 28, 2019 at 8:51 pm

Hello, pretty sure you have a typo on the first page. “Numer functions” was probably meant to be “Number functions” since google’s only real result searching for that exact phrase and python is this pdf.

Reply

G Srinivas Rao says:
April 12, 2019 at 9:28 am

You are great Mosh!! you have given all the codes at one place. This is great for beginners like me.

Reply

Leave a Reply

Enter your comment here...

Copyright © 2015 - Programming with Mosh - Powered by Wordpress
Design credit: Shashank Mehta + One Month Rails