

Python Cheatsheet

Basics

Variables

```
x = 5
y = "Hello, World!"
```

Data Types

```
x = 5          # int
y = 5.5        # float
z = 1j         # complex
a = "Hello"    # str
b = True       # bool
c = [1, 2, 3]  # list
d = (1, 2, 3)  # tuple
e = {1, 2, 3}  # set
f = {"name": "John", "age": 30} # dict
```

Type Casting

```
x = int(5.5)   # 5
y = float(5)   # 5.0
z = str(5)     # "5"
```

Strings

```
s = "Hello, World!"
print(s[0])      # H
print(s[0:5])    # Hello
print(s.lower()) # hello, world!
print(s.upper()) # HELLO, WORLD!
print(s.replace("H", "J")) # Jello, World!
print(s.split(",")) # ['Hello', ' World!']
```

Lists

```
mylist = ["apple", "banana", "cherry"]
print(mylist[1])    # banana
mylist[1] = "blackcurrant"
print(mylist)       # ['apple', 'blackcurrant', 'cherry']
mylist.append("orange")
print(mylist)       # ['apple', 'blackcurrant', 'cherry', 'orange']
mylist.remove("apple")
print(mylist)       # ['blackcurrant', 'cherry', 'orange']
```

Tuples

```
mytuple = ("apple", "banana", "cherry")
print(mytuple[1])   # banana
```

Sets

```
myset = {"apple", "banana", "cherry"}
myset.add("orange")
print(myset)       # {'banana', 'cherry', 'apple', 'orange'}
myset.remove("banana")
print(myset)       # {'cherry', 'apple', 'orange'}
```

Dictionaries

```
mydict = {
    "name": "John",
    "age": 30,
    "city": "New York"
}
print(mydict["name"])    # John
mydict["age"] = 31
print(mydict)            # {'name': 'John', 'age': 31, 'city': 'New York'}
mydict["email"] = "john@example.com"
print(mydict)            # {'name': 'John', 'age': 31, 'city': 'New York',
                          'email': 'john@example.com'}
```

Control Flow

If Statements

```
x = 10
if x > 5:
    print("x is greater than 5")
```

```
elif x == 5:
    print("x is 5")
else:
    print("x is less than 5")
```

For Loops

```
fruits = ["apple", "banana", "cherry"]
for fruit in fruits:
    print(fruit)
```

While Loops

```
i = 1
while i < 6:
    print(i)
    i += 1
```

Break and Continue

```
for i in range(10):
    if i == 5:
        break
    print(i)

for i in range(10):
    if i == 5:
        continue
    print(i)
```

Functions

```
def my_function():
    print("Hello from a function")

my_function()

def my_function_with_args(name):
    print("Hello, " + name)

my_function_with_args("John")

def my_function_with_return(x):
    return 5 * x
```

```
print(my_function_with_return(3)) # 15
```

Classes and Objects

```
class MyClass:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def myfunc(self):
        print("Hello, my name is " + self.name)

p1 = MyClass("John", 36)
p1.myfunc()
```

Modules

```
# Save this code in a file named mymodule.py
def greeting(name):
    print("Hello, " + name)

# Import the module
import mymodule
mymodule.greeting("John")
```

File Handling

```
# Write to a file
f = open("demofile.txt", "w")
f.write("Now the file has more content!")
f.close()

# Read from a file
f = open("demofile.txt", "r")
print(f.read())
f.close()
```

Exception Handling

```
try:
    print(x)
except NameError:
    print("Variable x is not defined")
```

```
except:
    print("Something else went wrong")
finally:
    print("The try-except block is finished")
```

Useful Libraries

NumPy

```
import numpy as np

a = np.array([1, 2, 3])
print(a) # [1 2 3]

b = np.array([[1, 2, 3], [4, 5, 6]])
print(b)
# [[1 2 3]
#  [4 5 6]]
```

Pandas

```
import pandas as pd

data = {
    "calories": [420, 380, 390],
    "duration": [50, 40, 45]
}

df = pd.DataFrame(data)
print(df)
#    calories  duration
# 0         420         50
# 1         380         40
# 2         390         45
```

Matplotlib

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [2, 3, 5, 7, 11]

plt.plot(x, y)
plt.xlabel('x-axis')
plt.ylabel('y-axis')
```

```
plt.title('Simple Plot')
plt.show()
```

Requests

```
import requests

response = requests.get('https://api.github.com')
print(response.status_code)
print(response.json())
```

Flask

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'

if __name__ == '__main__':
    app.run()
```

Django

```
# Install Django
# pip install django

# Create a new project
# django-admin startproject myproject

# Create a new app
# python manage.py startapp myapp

# Run the server
# python manage.py runserver
```

BeautifulSoup

```
from bs4 import BeautifulSoup
import requests

url = 'http://example.com'
```

```
response = requests.get(url)
soup = BeautifulSoup(response.text, 'html.parser')

print(soup.title)
print(soup.title.string)
```

Selenium

```
from selenium import webdriver

driver = webdriver.Chrome()
driver.get('http://example.com')

print(driver.title)
driver.quit()
```

TensorFlow

```
import tensorflow as tf

# Create a constant tensor
hello = tf.constant('Hello, TensorFlow!')
print(hello)

# Create a session to run the tensor
sess = tf.Session()
print(sess.run(hello))
```

PyTorch

```
import torch

# Create a tensor
x = torch.tensor([1, 2, 3])
print(x)

# Create a tensor with gradients
x = torch.tensor([1.0, 2.0, 3.0], requires_grad=True)
print(x)
```

OpenCV

```
import cv2
```

```
# Read an image
img = cv2.imread('example.jpg')

# Display the image
cv2.imshow('image', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Scikit-learn

```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

# Load the diabetes dataset
diabetes = datasets.load_diabetes()

# Use only one feature
diabetes_X = diabetes.data[:, np.newaxis, 2]

# Split the data into training/testing sets
diabetes_X_train, diabetes_X_test, diabetes_y_train, diabetes_y_test =
train_test_split(diabetes_X, diabetes.target, test_size=0.2,
random_state=42)

# Create linear regression object
regr = LinearRegression()

# Train the model using the training sets
regr.fit(diabetes_X_train, diabetes_y_train)

# Make predictions using the testing set
diabetes_y_pred = regr.predict(diabetes_X_test)

# The coefficients
print('Coefficients: \n', regr.coef_)
# The mean squared error
print('Mean squared error: %.2f' % mean_squared_error(diabetes_y_test,
diabetes_y_pred))
# The coefficient of determination: 1 is perfect prediction
print('Coefficient of determination: %.2f' % r2_score(diabetes_y_test,
diabetes_y_pred))
```