Python 3 Cheat Sheet

Comments

```
# This is a comment

'''
This is a
block
of
comments
'''

"""
This is another
block
of
comments
"""
```

Strings

```
# use single or double quotes
str_1 = "This is a string"
str_2 = 'This is another string'
```

Length of Strings

```
length = str(len(str_1))
```

Special Characters

```
print("Col 1\tCol 2\tCol 3")
print("Line 1\nLine 2\nLine 3")

path_1 = "c:\\windows\\"
sentence_1 =
    'He\'s allergic to cats'
```

Multi-lines

```
long_str = """This is a long
    long
    long
    string"""
```

Raw String

```
raw1 =
  r"The \n is the newline character"
```

String Formatting

```
fname = "Wei-Meng"
lname = "Lee"

print("%s %s" %(fname, lname))
# Wei-Meng Lee

print("%s %s" %(lname, fname))
# Lee Wei-Meng

print("{} {}" .format(fname, lname))
# Wei-Meng Lee

print("{0} {1}" .format(fname,
      lname))
# Wei-Meng Lee

print("{1} {0}" .format(fname,
      lname))
# Lee Wei-Meng

print(f"{fname} {lname}")
# Wei-Meng Lee
```

String Concatenation

```
word1 = "Hello"
word2 = "Python"
sentence = word1 + ", " + word2

words =
    ["The","quick","brown","fox",
     "jumps","over","the","lazy",
     "dog"]
print(' '.join(words))
print(', '.join(words))
```

Getting Inputs

```
name = input(
           "What is your name?")
year_born = input(
           "Enter your year born: ")

current = 2018
my_age = current - int(year_born)
     # need to convert string to int
```

Defining Function

```
def do_something():
    print("In do_something()")
```

Calling Function

```
do_something()
```

Function with Parameters

```
def add_nums(x,y):
    return x+y

print(add_nums(5,6))          # 11
print(add_nums(y = 6,x = 5))  # 11
```

Optional Parameters

```
def merge(lst, spacer=" "):
    s = ""
    for item in lst:
        s = s + str(item) + spacer
    return s

list1  = [1,2,3,4,5]
print(merge(list1))
print(merge(list1,"-"))
print(merge(list1, spacer = "*"))
print(merge(spacer = "*",
            lst = list1))
```

Global Variable

```
count = 0
def addcounter():
    global count
    count = count + 1
```

Lambda Functions

```
def square(n):
    return n**2

print(square(5))  # 25

g = lambda n: n**2
print(g(5))       # 25
```

Filter, Map, and Reduce

```
from functools import reduce

lst = [1,2,3,4,5,6,7,8,9]

evens = filter(
    lambda x: x % 2 == 0, lst)
print(evens)          # [2, 4, 6, 8]

squares = map(lambda x: x ** 2, lst)
print(squares)        # [1, 4, 9,
                      # 16, 25, 36,
                      # 49, 64, 81]

total = reduce(
    lambda s, x: s + x, lst)
print(total)          # 45
```

Exceptions Handling

```
try:
    a = input('Enter first num: ')
    b = input('Enter second num: ')
    result = a/b
except ZeroDivisionError as e:
    print(e)
except:       # catch all exceptions
    e = sys.exc_info()[0]
    print(e)
else:                 # no exception
    print(result)
finally:     # do this regardless of
             # exception or not
    print("End of program")
```

Raising Exceptions

```
def perform_division(a,b):
    if b == 0:
        raise ZeroDivisionError
    return a/b

try:
    result = perform_division(5/0)
except ZeroDivisionError as e:
    print(e)
```

Lists

```
int_list = [1,2,3,4,5]
mix_list = [3.14, 5, "Hello"]
list_of_list = [int_list, mix_list]

print(len(int_list))     # 5
print(len(mix_list))     # 3
print(len(list_of_list)) # 2

print(int_list[0])       # 1
print(int_list[1])       # 2
```

Inserting and Removing Items

```
# insert 9 into index 3
int_list.insert(3, 9)
print(int_list)
# [1, 2, 3, 9, 4, 5]

# insert 8 into index 20; appended
# to the end instead
int_list.insert(20, 8)
print(int_list)
# [1, 2, 3, 9, 4, 5, 8]

# remove element at index 3
n = int_list.pop(3)
print(n)                 # 9
print(int_list)
# [1, 2, 3, 4, 5, 8]

# error; pop index out of range
n = int_list.pop(20)
```

Ranging

```
r = range(5)   # r is [0,1,2,3,4]
print(r[0])    # 0
print(r[-1]    # 4; -1 is for last
               # element
print(r[-2])   # 3; -2 is for second
               # last element
```

Slicing

```
print(r[-2:])  # [3,4]; second last
               # element onwards
print(r[:3])   # [0,1,2]; first 3
               # element
print(r[1:3])  # [1,2]; index 1 and
               # up to before 3
print(r[1:-1]) # [1,2,3]; without
               # the first and last
print(r[2:-2])

x = r[:]       # make a copy of r
print(x)       # [0,1,2,3,4]
```

Membership

```
two_in_list = 2 in r
nine_in_list = 9 in r

print(two_in_list)   # True
print(nine_in_list)  # False
```

List Concatenation

```
r.extend([5,6])
print(r)             # [0, 1, 2, 3, 4,
                     # 5, 6]
s = r + [7,8,9]
print(s)             # [0, 1, 2, 3, 4,
                     # 5, 6, 7, 8, 9]
s.append(10)
print(s)             # [0, 1, 2, 3, 4,
                     # 5, 6, 7, 8, 9,
                     # 10]
```

List Unpacking

```
scores = [70,55,90]
math, science, english = scores

print("math: %d" %math)
# math: 70

print("science: %d" %science)
# science: 55

print("english: %d" %english)
# english: 90

_, _, english = scores
# only interested in english
print("english: %d" %english)
# english: 90
```

Enumerating a List

```
seasons = ['Spring', 'Summer',
           'Fall', 'Winter']
for i, season in enumerate(seasons):
    print(i, season)

for i, season in enumerate(seasons,
    start=1):
    print(i, season)
```

List Comprehension

```
nums = [1,2,3,4,5]
cubes = [n ** 3 for n in nums]
print(cubes)
'''
[1, 8, 27, 64, 125]
'''

even_cubes = [ n ** 3 for n in nums
    if n % 2 == 0 ]
print(even_cubes)
'''
[8, 64]
'''
```

Making Decisions

```
raining = True
windy = False

if raining:
    print("Wet!!!")

if not raining:
    print("Dry!!!")

if raining and windy:
    print("Cold!!!")

if raining or windy:
    print("Cool!!")
```

Comparison

```
your_age = 30
my_age = 35

if my_age > your_age:
    print("I am older than you!")
elif my_age < your_age:
    print("You are older than me!")
else:
    print("We are both as old!")

if (your_age == 100):
    print("You are a centenarian!")

if (your_age != 100):
    print("You are not a
centenarian!")
```

Ternary Condition

```
num = 5
parity =
   "even" if num % 2 == 0 else "odd"
print(parity)
```

For Loop

```
for x in range(1, 15, 2):
    print(x)     # prints 1 3 5 7 9
                 # 11 13

for x in range(10):
    print(x)     # prints 0 to 9
```

Break

```
for x in range (10):
    print(x)     # prints 0 to 5
    if (x == 5):
        break
```

Continue

```
for x in range (10):
    if (x == 5):
        continue
    print(x)     # prints 0 to 9,
                 # excluding 5
```

While Loop

```
max = 10
count = 1
while (count <= max):
    print(count) # prints 1 to 10
    count += 1
```

Tuples

```
pt_a = (7,8)
pt_b = 3,4
print(pt_a)         # (7, 8)
print(pt_b)         # (3, 4)

pt_a[1] = 7   # Error
```

Dictionaries

```
heights = {}
heights = { "john": 176,
            "peter": 158,
            "susan":170
          }
print(heights["peter"])
print(heights["jack"])     # Error

heights["mary"] = 168
print(heights["mary"])
```

Dictionary Membership

```
have_john = "john" in heights
have_jack = "jack" in heights

print(have_john)   # True
print(have_jack)   # False
```

Sets

```
items = set()
items.add("Apple")
items.add("Orange")
items.add(1)

items.add("Durian")
items.remove("Durian")
```