Cheatography

Python Course - Day 1 Cheat Sheet

by shao via cheatography.com/217383/cs/47718/

Arithmetic Operators

x + y	Addition, **adds** x to y
x - y	Substraction, **substract** y from x
x / y	Division, **divide** x by y
x * y	Multiplication, **multiply** x by y
x // y	Floor division, division operation that returns the largest integer that is **less than** or **equal** to the result of the division
x % y	Modulo, find the **remainder** after dividing x by y (26 % 5 = 1))
x**y	Exponentiation, calculate a x to the **power** of y

Comparison Operators

x == y	is x **equal** to y?
x != y	is x **not equal** to y?
x > y	is x **greater** than y?
x < y	is x **lower** than y?
x >= y	is x **greater** than or **equal** to y?
x <= y	is x **lower** than or **equal** to y?

Logical Operators

not	**opposite** Boolean value
and	**both** Boolean value must be True
or	**at least one** Boolean value must be True

Membership Operators

substring **in** my_string	checks if substring appears in my_string

Snippets - Strings

```
# Slicing - variable[start:stop:step]
my_text = " abc def gh"
print( my_ tex t[3:6]) # Output: " def "
print( my_ tex t[3 :6:2]) # Output: " df"
print( my_ tex t[::]) # Output: " abc def gh"
print( my_ tex t[: :-1]) # Output: " hgf edc ba"
# Concat enation
my_text = " Hel lo" + " " + " Wor ld"
really _happy = "I am happy" + " !" * 5
# Formatting
name = " Ali ce"
```

Snippets - Strings (cont)

> age = 25
print(f"My name is {name} and I am {age} years old.") # Output: "My name is Alice and I am 25 years old."

String operations

len(s)	Returns **length** of *s*
s.count(substring)	Returns how many times *substring* **appears in** *s*
s.find(substring)	Returns the index of the**first occurence** of *substring* in *s*, or **-1** if not found
s.lower()	Returns *s* with all characters to **lowercase**
s.upper()	Returns *s* with all characters to **uppercase**
s.strip()	Returns *s* with leading and trailing **whitespace removed**
s.capitalize()	Returns *s* with the **first letter capitalized**
s.replace(old, new)	Returns a string in which occurences of *old* in *s* are **replaced** by *new*
s.split(delimiter)	Returns a **list** by splitting *s* when a *delimiter*
separator.join(text1, text2, ..., textn)	Returns elements of an iterable into a **single** string with a separator string
s.startswith(prefix)	returns *True* if *s* **starts with** the specified *prefix*
s.endswith(suffix)	returns *True* if *s* **ends with** the specified *suffix*
s.isalpha()	returns *True* if all characters in *s* are **alphabetic**
s.isalphanum()	returns *True* if all characters in *s* are **alphanumeric**
s.isdigit()	returns *True* if all characters in *s* are **digits**

Data Types

int	scalar	Integer, represent a number from -2'147'483'648 to 2'147'483'647
float	scalar	Float, represent a floating point number
bool	scalar	Boolean, represent either **True** or **False**
None	scalar	NoneType, represent an **empty** object
str	non-scalar	String, represent a **chain of characters**
tuple	non-scalar	Tuple, represent an **immutable** and **ordered** collection of data

Snippets - Loops

```
# for - range(start, stop, step)
for i in range(10):
print(i) # Output: " 0", " 1", ..., " 9"
# for - in
my_text = " par se"
for c in my_text:
print(c) # Output: " p", " a", " r", " s", " e"
# while
continue = True
i = 0
while( con tinue):
if i >= 3:
continue = False
else:
print(i) # Output: " 0", " 1", " 2"
```