Cheatography

Python Basics - Strings Cheat Sheet

by Mario (mariofreitas) via cheatography.com/42813/cs/13207/

String Syntax

`s1 = "this is "`	Strings can be declared with " "...
`s2 = 'a string'`	... or ' '
`s1 + s2`	Returns *s1* concatenated with *s2* ('this is a string')
`s1 * 3`	Returns *s1* concatenated with itself 3 times (this is this is this is)
`s1[3]`	Returns 4th element of s1 (s)
`s1[0:3]`	Returns 1st to 3rd element of s1 (thi)
`s1[0:7:2]`	Returns 1st to 7th element of s1 skipping one at a time (ti s)

String Methods

`s = "stRing"`	
`s.capitalize()`	Returns capitalized version of *s* (String)
`s.upper()`	Returns upper case version of *s* (STRING)
`s.lower()`	Returns lower case version of *s* (string)
`s.title()`	Returns *s* with first letter of each word capitalized (String)
`s.swapcase()`	Returns the case swapped version of *s* (STrING)
`s.replace('tR', 'l')`	Returns a copy of *s* with all *'tR'* replaced by *'l'* (sling)
`s.startswith('R')`	Returns true if *s* starts with *'R'* and false otherwise (False)
`s.endswith('ing')`	Returns true if *s* ends with *'ing'* and false otherwise (True)
`s.split('R')`	Splits the string into a list of strings. In this case, *"R"* is the splitting parameter. (["sr", "ing"])
`s.strip()`	Removes spaces in the begining and in the end of the string ("stRing")
`s.strip("g")`	Removes *"g"* in the begining and in the end of the string ("stRin")
`''.join([s, 's are cool'])`	Returns the string '' concatenated with *s* and *'s are cool'* ('stRings are cool')

String Formatting - Printf Arguments

`d`	Int
`f`	Float
`s`	String
`10d`	Reserves 10 spaces to the int
`^10d`	Reserves 10 spaces to the int and centralize the content
`<10d`	Reserves 10 spaces to the int and align the content left
`>10d`	Reserves 10 spaces to the int and align the content right
`*^10d`	Reserves 10 spaces to the int , centralize the content and fill the empty spaces with *
`0>10d`	Reserves 10 spaces to the int , align the content right and fill the empty spaces with **0s**
`0>.2f`	Format float with 2 decimal places
`0>10.2f`	Reserves 10 spaces to the float and format with 2 decimal places

String - The format() Method

```
a = 10
b = 3.5555
print("The value of a is {} and the value of b is {:.2f}".format(a, b))
```

Instead of using a formatted string (only available on Python 3.6 and up) you can also use the format method inserting **.format()** at the end of the string.

String Formatting - Example

```
a = 10.12571
print(f"The value of a is {a:.2f}")
# This code prints "The value of a is 10.13"
# Use f before starting a string to make it a formatted string
# Use {a} in a formatted string to interpolate the variable a in the string
# Use :.2f after the variable name to format it as a float with 2 decimal places
```