Cheatography

strings, lists, tuples and dictionaries in python Cheat Sheet

by corisco via cheatography.com/167022/cs/34954/

STRINGS

Strings	string = "words" or 'words'
conect	string1+string2
multiply	string*3
compare ('if' or 'if not')	string1 == string2 (or != or > or <)
check ('if or 'if not')	'substring' in string
position	string[0] or string[-1] or string[x]
slicing	string[start:end:step]
	string[:3] or string[1:]
	string[::-1] inverted

Slicing strings

slicing:	string[start:end:step]
from start to end	string[2:5]
from start	string[:5]
to the end	string[2:]
jumping	string[2:5:2]
negative/inverted	string[::-1]

Modify strings

remove whitespaces before and after the string	string.strip()
or remove any 'characters'	string.strip('ch')
only before(left) the string	string.lstrip()
only after(right) the string	string.rstrip()
replace string	string.replace("old value", "new value", counter)
counter is how many occurences from start, nothing is default and means every occurrence	
use it to remove whitespaces inside the string	string.replace(" ", "")
split strings create list from strings	string.split(separator,maxsplit)
separator is optional, whitespace is default	string.split(",")

Modify strings (cont)

maxsplit is optional, all occurrences is default(-1)	string.split("-",1)
find string	string.find("value",start,end)
find the first occurrence of a substring in a string	string.find("substring")
start and end are optional	string.find("substring",3,10)
returns the position of the substring	
index and find are the same, except that when false find returns -1 and index returns error	string.index("substring")
lowercase	string.lower()
uppercase	string.upper()
capitalize first char upper	string.capitalize()
title first char of each word upper	string.title()
join values of a list/dict/tuple/ into string	' '.join(list)
count the occurences of substring in a string	string.count('substring')

Format strings

str.format	'Hello, {} and {}'.format(string, string2)
f-string (3.6+)	f'Hello, {string} and {string2}'
*Integer numbers into strings**	f'Hello, {string:_}'
'b' - binary	{string:b}
'c' - character (ASCII)	'Hello, {example:c}'.format(example = 'String', ...)
'd' - decimal	'Hello, {0:d} and {1}.format(string,string2)
'o' - octal	{string:o}
'x' - hexadecimal, lowercase	'Hello, {0.x} and {0:X}.format(string)

Format strings (cont)

'X' - hexadecimal, uppercase	{string:X}
'n' - number, decimal in local language OS	{string:n}
Floating-point	f'{integer:f}' (6 standard)
round(a,2)	f'{integer:.2f}' (2 decimal digits)
'e' ou 'E' - scientific notation (6 standard)	{:e}.format(999)
'E' - scientific notation (6 standard)	f'{999:.3E}'
'g' precisão >=1, round digits *p* to significant digit	{:g}.format(12.1231235843)
'n' - same as 'g', but local language OS	f'{12.1231235843:n}'
'%' - multiply the number *100 and '%' after	f'{0.05:%}'
two digits before '.' and two after	'{:2.2%}'.format(0.05)
Spacing	
'<' left align / '^' center align / '>' right align	'{0:<16}'.format(string) / f.'{string:^16}' / {:>16}
number of digits	{:4)
whitespaces if the string has less than 16..	{string:16}
choose char instead of whitespace	{*:16}

COLLECTIONS

List	collection which is ordered and changeable. Allows duplicate members.
Tuple	collection which is ordered and unchangeable. Allows duplicate members.
Dictionary	collection which is ordered and changeable. No duplicate members.
Set	collection which is unordered, unchangeable, and unindexed. No duplicate members.

LISTS

Lists	list = ["string", "string2", integer, Bool]
ordered, changeable, allow duplicates	list = [item1, item2, item3]
Lenght	len(list)
Access *[start index included:end index not]*	list[1] / list[-1], list[2:5], list[:5]
Check if Item exists	if "item" in list:
Change items	list[3] = "new_value"
range of items	list[2:5] = "new_value1", "new_value2"
insert(index,value) *without replacing any item*	list.insert(2, "item")
append() *add item to the end*	list.append("item")
extend() *add items from other list*	list.extend(list2)
works with tuples also	list.extend(tuple)
remove() *remove first matching value*	list.remove("string")
pop() *remove specified index, and returns it*	list.pop(1)
del() *remove specified index*	list.del(3)
clear() *empties the list*	list.clear()
sort() *sort the list alphanumerically*	list.sort()
sort descending	list.sort(reverse=True)
reverse() *order*	list.reverse()
copy() make a copy	list2 = list.copy()
or use list()	list3 = list(list2)
Concatenate Lists	list3 = list1 + list2
or use extend()	list.extend(list2)
count() returns the number of items*	list.count() / list.count('value')
index() *finds the item and return its index*	list.index('value')
min() / max() / sum()	list.min() list.max() list.sum()

LISTS (cont)

enumerate(index, value)	for i, v in enumerate(list): /n "{i} : {v}'
for loop	for x in list:
through index	for x in range(len(list)):
while loop	while x <= len(list): /n i+=1
list compreehension	[print[x] for x in list]

TUPLES

Tuples	tuple = ("value1", int, bool ,"value1")
are unordered, unchangeable, allow duplicates	tuple = (item1,)
Lenght	len(tuple)
Access *[start index included:end index not]*	tuple[1] / tuple[-1], tuple[2:5], tuple[:5]
Check if Item exists	if "item" in tuple:
Change items *tuples are unchangeable*	list = list(tuple)
convert tuple to list and back to tuple	tuple = tuple(list)
Concatenate *add tuple to a tuple*	tuple1 + tuple2 / tuple1 =+ tuple2
Lists inside a tuple *are changeable*	tuple = (["value1","value2"], item2, item3)
count() *returns the number of items*	tuple.count() / tuple.count('value')
index() *finds the item and return its index*	tuple.index('value')
min() / max() / sum()	tuple.min() tuple.max() tuple.sum()
enumerate(index, value)	for i, v in enumerate(tuple): /n "{i} : {v}'
for loop	for x in tuple:
through index	for x in range(len(tuple)):
while loop	while x <= len(tuple): /n i+=1
list compreehension	[print[x] for x in tuple]

DICTIONARIES

Dictionaries	dict = {"key":"value", "key2":"value2"}
*ordered, changeable, do **not** allow duplicates*	dict={"key1": bool,"key2":int, "key3":[list]}
dict cannot have same keys	
Lenght	len(dict)
Access *get the value of the "key"*	value1 = dict["key1"]
dict.get("key","return if not found")	value4 = dict.get("key4", "Not found")
List of Keys	x = dict.keys()
Check if key exists	if "keys" in dict:
if values exists	if "value1" in dict.values()
Change values of a key	dict["key"] = value
using update()	dict.update({'key':'value'})
Add	dict["key"] = value
also can use update()	dict.update({'key':'value'})
Remove *.pop or popitem(removes the last key inserted))*	dict.pop("key") / dict.popitem()
using del	del dict("key")
del can delete the dictionary completely	del dict
clear *empties the dictionary*	dict.clear() or dict = {}
Copy	dict = dict2 / dict2 = dict.copy() / dict2 = dict(dict)
items()	dict.items()
keys()	dict.keys()
values()	dict.values()
for loop	for keys in dict:
keys	for keys in dict.keys():
values	for values in dict.values():
keys and values	for keys, values in dict.items()