Cheatography

Python basics Cheat Sheet

by DarioPittera (aggialavura) via cheatography.com/83764/cs/19817/

OPERATIONS

esponential	**
remainer	%

Parenthesis have priority

STRINGS

quotes	'
double quotes	"
wrap quotes	"'"

PRINT

```
# Normal print
print(var)
# Print formatting
print('My number is: {}, and my name is: {}'
.format(v1,v2))
# Print formatting 2
# (with order)
print('My number is: {one}, and my name is: {two}'.format(one=-var1,two=var2))
```

VARIABLES NAMES RULES

. cannot start with numbers
. cannot start with special char
. underscore to chain_words

VARIABLE TYPES

Lists	it uses **[ ]**
create	list = ['a','b','c'] or list = [1,2,3]
append	list.append('d')
nested	nest = [1,2,3,[4,5,['target']]]
- indexing	nest[3][2][0] = ''target'
Tuples	uses **( )** cannot change
create	t = (1,2,3)
Dictionaries	it uses **{ }** has keys and values
create	d = { 'key1':'item1', 'key2':'item2'}
indexing	d['key1'] = 'item1'
indexing 2	d[key1][0:3] = 'ite'
Sets *	it uses **{ }** has unique values
create	s = {1,2,2,2,2,3,3,3,4} = 1,2,3,4
append	s.add(5)
Booleans	True, False

sets differs from dictionary because has no keys and has unique values

INDEXING

```
s = 'hello'
# Indexing starts at 0
s[0]= 'h'
# the last value is excluded
s[0:2] = 'he'
```

LOOPS

if	if 1 < 2: print('Yep!')

if else	if 1 > 2:
	  print('first')
	else:
	  print('last')

if elif	if 1 == 2:
	  print('first')
	elif 3 == 3:
	  print('middle')
	else:
	  print('Last')

for	seq = [1,2,3,4,5] for item in seq: print(item)

while	i = 1
	while i < 5:
	  print('i is: {}'.format(i))
	  i = i+1

* **elif**: will execute only the first true condition

By **DarioPittera** (aggialavura)

cheatography.com/aggialavura/
www.dariopittera.com

Not published yet.
Last updated 24th June, 2019.

Cheatography

Python basics Cheat Sheet

by DarioPittera (aggialavura) via cheatography.com/83764/cs/19817/

FUNCTIONS

range(num)	create range from 0 to x
list(range(num))	convert range to list
list comprehension	out = [var**2 for var in x]
functions *	def fun(param1='default'): print('Hello ' + param1)
return	return a value
print	print a value
lambda expressions *	lambda var: var*2
map *	map(function,seq)
filter *	filter(function,seq)

In functions param1 defines the default parameter if there is no input.
lambda: take what: give what.
map: it applies a function to something. map(what, to_what). Also, the **list** is necessary to have the right output.
apply works on a row / column basis of a DataFrame, **applymap** works element-wise on a DataFrame, and **map** works element-wise on a Series.
filter: takes out only the results of a certain condition

METHODS

string.lower()	'aaaa'
string.upper()	'AAAA'
string.split() *	'ciao' 'sono' 'io'
s.split('symbol')	split using a symbol
dict.keys()	dict_keys(['key1', 'key2'])
dict.items()	dict_items([('key1', 'item1')])
list.pop(n)	take out the Ith element
var in [list]	check in a list - bool

split() supports also **indexing** after: str.split()[0]

By **DarioPittera** (aggialavura)

cheatography.com/aggialavura/
www.dariopittera.com

Not published yet.
Last updated 24th June, 2019.