Python Basics Cheat Sheet

by user111 via cheatography.com/37570/cs/11802/

Basic of Basic

comment	#comment
print	print()
assignment sign	=
equal sign	==
case sensitive	YES
indenting matters	YES
help(x) or ?x	INFO on x
IF condition	if elif else
empty_list = []	initiate an empty list
!ls	list files in wd

Math Operators

basic 4	+ - * /
exponentiation	**
modulo	%
floor division	//

ZERO Indexing

x[0]	1st element of x
x[-1]	last element of x
x[a:b]	from position "a" to "b-1"
x[a:]	from position "a" to the last
x[:b]	from position "b-1" to the first
x[::-1]	read backward 1 step back

Basic functions

Type & conversion	type() str() int() float() bool()
	len()
define function	def function_name(parameter):
output	return new_value
docstrings	""" foo """ after function declared

LIST

list = [a, b, c]	compound data type
add to list	foo + list
remove from list	del(list[x])
copy list (explicit)	list() or list[:]

List & Dict comprehension

new_nums = [num + 1 for num in nums]

new_pairs = [(num1, num2) for num1 in range1 for num2 in range2]

new_dict = {num: num+1 for num in range(x)}

Most common packages

Pandas	Dataframe
Numpy	Array
Matplotlib	Visualisation
Scikit-learn	Machine learning

STRING methods

find	<str>.find(<str>, <number>)
	1st find position; -1 if not found starting from <number>
case	upper() lower()
count	count("foo")

LIST methods

append	list.append(foo)
index	list.index(foo)
count	list.count(foo)
remove (1st match)	list.remove(foo)
reverse (order)	list.reverse()

Dataframe methods

df.columns	df.head(), df.tail()
df.shape, df.info()	df['foo'].value_counts(dropna=FALSE)
df.describe()	summary statistic of numeric data

Iterating

list	for index, item in enumerate(list_name)
string	for chr in "name"
dictionary	for key, val in dict_name.items()
Numpy array	for val in numpy.nditer(array_name)
Pandas series	for label, row in pandas_name.iterrows()

Iterating (cont)

data	for chunk in pd.read_csv('data.csv', chunksize = 1000)

iterate through rows in pandas is not efficient. avoid if possible.

i

an iterable is an object that can return an iterator, while an iterator is an object that keeps state and produces the next value when you call next() on it

iterable -> iter() -> next()

print(*x) to print all elements

Quirks

Different types of value

> string*number is OK but string+number is error

Multiple assignment

> a, b = 3, 4 the value of a is 3 and the value of b is 4

tuple vs. list vs. dictionary

> ( ) vs. [ ] vs. {}, immutable vs. mutable vs. immutable key

declare non-local var

> in local functions

declare global var

> in enclosing functions

generator

> not stored in memory like list, iterable

BEWARE

edge case: s[0] causes ERROR if s is empty

list is ordered vs dictionary is un-ordered

Functions on function

map()	applies function over an object
reduce()	filter()

Used together with lambda

By **user111**

cheatography.com/user111/

Not published yet.

Last updated 18th May, 2017.