http://pythontutor.com/visualize.html*#mode=edit*

**Comments:**

[**python**]

# comment

"""

multi

line

comment

"""

[**end**]

**Print and Strings:**

[**python**]

print('Hello')

print('World')

print('Hello') ; print('World')

print('Hello', 'World') #auto space

print('Hello', end=' '); print('World')

print('Hello', 'World', sep=', ')

[**end**]

*- Concat:*

'Alice' + 'Bob'

*- Multiple:*

'Alice' \* 5

*- Length:*

len(myStr)

*- to int:*

myint = int(myStr)

*- to str:*

str(myInt)

*- to float:*

float(myStr)

*- to lowercase:*

myStr.lower()

myStr.upper()

myStr.upper().lower()

myStr.isupper() *#has no lowercase and at least one upper*

myStr.islower() *#has no uppercase and at least one lower*

*- is:*

isalpha() = if the strings consists only of letters and is not blank

isalnum() = if the string consists only of letters and numbers and is not blank

isdecimal() = if the string consists only of numeric characters and is not blank

isspace() = if the string consists only of spaces, tabs, and new lines and is not blank

istitle() = if the string consists only of words that begin with an uppercase letter followed by only

*- startswith/endswith:*

'Hello world'.startswith('Hello')

*- can use the list functions:*

*- escape characters:*

\'

\"

\t

\n

\\

*- raw string:*

print(r'That is \t awesome')

That is \t awesome

*- multiline strings:*

print('''

this

is

a

whole

string''')

*- rjust/ljust/center:*

move text right, left, or center

*- removing whitespace:*

myStr.strip() *#return a new string without any whitesapce characters at the beginning or end*

myStr.lstrip()

myStr.rstrip()

*- copy/paste:*

import pyperclip

pyperclip.copy('test')

pyperclip.paste()

*<========================================*>

**Math Operators:**

[**python**]

\*\* #Exponent

% #Modulus

// #Floor division

/

\*

-

+

[**end**]

*<========================================*>

**Logical Operators:**

[**python**]

and

or

not

[**end**]

*<========================================*>

**Condition Statements:**

[**python**]

if condition :

code

elif condition :

code

else :

code

[**end**]

! -1 is not a false statement

*<========================================*>

**Loop Statements:**

[**python**]

#while

while condition :

code

[**end**]

*-* break = break Loop

*-* continue = jump back to start of loop

[**python**]

#do while

while True:

print(i)

i = i + 1

if(i > 3):

break

[**end**]

[**python**]

#for

for i in range(5): # up to but not including 5

print('Jimmy Five Times ' + str(i))

range(12,16) #12,13,14,15

range(0,10,2) #0,2,4,8

#for lists

for n in numbers:

print(n)

[**end**]

range(start num, up to num, increment by num)

*<========================================*>

**Standard Library:**

[**python**]

import random, sys, os, math

random.randint(1,10)

from random import \* #from random import everything

randint(1,10)

sys.exit()

[**end**]

! help(randint)

*<========================================*>

**Lists, tuples, Dictionaries:**

**Lists:**

*-* (Arrays)

*-* Passes by reference when copied and when passed to function

[**python**]

spam = [['hello', 'world'], 3.1415, True, None, 42]

spam[0] #['hello', 'world']

spam[-1] #42

spam[1:3] #[3.1415, True]

spam[1:-1] #[3.1415, True, None]

spam[1:] #[3.1415, True, None, 42]

spam[:2] #[['hello', 'world'], 3.1415]

len(spam) #5

[**end**]

*- Concat:*

list1 + list2

*- Delete and Remove:*

del spam[**0**] *#can also delete variables*

spam.remove('hello')

*- Dynamic:*

[**python**]

catNames = []

while True:

name = input()

catNames = catNames + [name]

[**end**]

*- In/Not In:*

'chris' in ['hello', 'world'] *#false*

'chris' not in ['hello', 'world'] *#true*

*- Multiple Assignment:*

[**python**]

cat = ['fat', 'orange']

size = cat[0]

color = cat[1]

#SAME AS

size, color = cat

#Use to swap variables nv nvqsggv

a, b = 'Hello', 'World'

a, b = b, a

[**end**]

*- Index:*

spam.index('Hello') *# returns 0*

*- Append:*

spam.append('World')

*- Insert:*

spam.insert(1, 'to the') *#[Hello, to the, World]*

*- Sort:*

spam.sort()

*-* cant use on lists with different types

spam.sort(key=str.lower)

*-* sort by letters instead of by ascii (capitols sorted first and then lowercase)

*- Reverse:*

spam.sort(reverse=True)

*- Copy:*

import copy

spam = [...]

cheese = copy.copy(spam)

*-* if the list has a list inside use deepcopy

cheese = copy.deepcopy(spam)

*- join/split:*

', '.join(myCat) *#string of list joined by ,*

myCat.split('\n') *#list of entries divided at every ,*

**Tuples:**

*-* immutable data types (immutable = unchangeable)

eggs = ('hello', 42, 0.5)

eggs[**0**]

test = ('hello',)

*- Convert:*

tuple(['cat', 'dog', 45])

list(('cat', 'dog', 5))

list('hello') *#['h','e','l','l','o']*

**Dictionary:**

(associated array)

myCat = {'size': 'fat', 'color': 'gray', 'disposition': 'loud'}

print(mycat['size']) *# 'fat'*

*-* order doesnt matter for determining equality

*- Access Info:*

myCat.values() *#immutable list returned*

myCat.keys() *#immutable list returned*

myCat.items() *#immutable list returned*

*- For loop:*

for k, v in spam.items():

print('Key: ' + k + ' Value: ' + str(v))

*- In/Out:*

'color' in myCat.keys() *#True*

'gray' in myCat.values() *#True*

*- Get:*

myCat.get('name', 'cat doesn't have a name')

1st param: key to get value from

2nd param: value to return if key isnt found

*- Setdefault:*

myCat.setdefault('name', 'sam')

Checks if name key exists and if not create it and set it to sam

Returns the key's value

*- Pretty Printing:*

import pprint

pprint.pprint(myCat) *#prints*

pprint.pformat(myCat) *#returns a string*

Prints the contents of a dictionary better

*<========================================*>

**Functions:**

[**python**]

def hello():

print('Hello')

hello()

hello()

[**end**]

*-* non return functions return None (null)

[**python**]

spam

spam == None #True

[**end**]

*<========================================*>

**Keyword Arguments:**

*-* Optional arguments to pass to the function call

*-* ex) print('Hello', 'World', sep=' ')

[**python**]

def quadratic(a=0, b=0, c=0): #a, b, c have default values of 0

x1 = -b / (2\*a)

x2 = sqrt(b\*\*2 - 4\*a\*c) / (2\*a)

return (x1 + x2), (x1 - x2)

quadratic(31, 93, 62) # order matters when passed positionally

quadratic(a=31, c=62, b=93) # order doesn't matter when passed by name

[**end**]

*-* Capture all positional arguments @ once)

[**python**]

def product(\*numbers, initial=1):

total = initial

for n in numbers:

total \*= n

return total

[**end**]

*-* Required keyword arguments)

[**python**]

def join(\*iterables, joiner): # iterables catches all positionally passed args so "joiner" must be a keyword arg that needs to be specified since it doesnt have a default val

if not iterables:

return

yield from iterables[0]

for iterable in iterables[1:]:

yield joiner

yield from iterable

[**end**]

*-* ONLY keyword arguments)

*-* \* as a parameter without anything after it

*-* ANY keyword arguments)

[**python**]

def format\_attributes(\*\*attributes):

"""Return a string of comma-separated key-value pairs."""

return ", ".join(

f"{param}: {value}"

for param, value in attributes.items()

)

format\_attributes(name="Trey", website="http://treyhunner.com", color="purple")

[**end**]

*-* Dictionary as keyword arguments)

[**python**]

items = {'name': "Trey", 'website': "http://treyhunner.com", 'color': "purple"}

format\_attributes(\*\*items)

[**end**]

*<========================================*>

**Global:**

*-* modify a global variable in local scope with global keyword

[**python**]

def spam():

global eggs

eggs = 'spam'

eggs = 'global'

spam()

print(eggs) #prints spam

[**end**]

*<========================================*>

**Exceptions:**

[**python**]

try:

print(42/0)

print(42/2) #never ran

except ZeroDivisionError:

print('cant divide by 0')

[**end**]

*-* Once jumped to except doesnt return to try

*-* raise exceptions

*-* traceback errors

import traceback

*-* assert

assert podBayDoorStatus == 'open', 'The pod bay doors need to be "open".'

if the assert fails the code fails

*<========================================*>

**Files:**

import os

*- Join file path:*

os.path.join('usr', 'bin', 'spam')

*- Get current working directory:*

os.getcwd()

*- Change current working directory:*

os.chdir

*- Go back a directory:*

..

*- File size:*

os.path.getsize(path)

*- Get files in dir:*

os.listdir(path)

*- Check if it exists:*

os.path.exists(path)

*- Check file exists:*

os.path.isfile(path)

*- Check dir exists:*

os.path.isdir(path)

*- Open/read/write/close file:*

myFile = open(path, mode = r)

myContents = myFile.read()

readlines() *# list of string values from file*

myFile.close()

*#write*

open(path, w) *# overrides original file contents*

*# creates a new file if it doesnt exist*

write('test')

myFile.close()

*#append*

open(path, a) *# append to the end of the file*

*-* copy,move,rename,delete:

import shutil

*- safe delete:*

import send2trash

*<========================================*>

**Third-Party Modules:**

*-* done with pip

Windows)

In the python directory, under scripts folder

./pip.exe install pyperclip *# allows you to copy and paste stuff*

*- Command Line arguments:*

import sys

sys.argv[**1**]

*- Save variables to hard drive:*

import shelve

myFile = shelve.open('mydata')

cats = ['zophie', 'pooka', 'Simon']

myFile['cats'] = cats

myFile.close()