#BASICS OF PYTHON

def quicksort(arr):

if len(arr) <= 1:

return arr

pivot = arr[len(arr) // 2]

left = [x for x in arr if x < pivot]

middle = [x for x in arr if x == pivot]

right = [x for x in arr if x > pivot]

return quicksort(left) + middle + quicksort(right)

print(quicksort([3,6,8,10,1,2,1]))

#Basic data types

#Numbers

x = 3

print(x, type(x))

print(x + 1) # Addition

print(x - 1) # Subtraction

print(x \* 2) # Multiplication

print(x \*\* 2) # Exponentiation

x += 1

print(x)

x \*= 2

print(x)

y = 2.5

print(type(y))

print(y, y + 1, y \* 2, y \*\* 2)

#Booleans

t, f = True, False

print(type(t))

print(t and f) # Logical AND;

print(t or f) # Logical OR;

print(not t) # Logical NOT;

print(t != f) # Logical XOR;

#Strings

hello = 'hello' # String literals can use single quotes

world = "world" # or double quotes; it does not matter

print(hello, len(hello))

hw = hello + ' ' + world # String concatenation

print(hw)

hw12 = '{} {} {}'.format(hello, world, 12) # string formatting

print(hw12)

s = "hello"

print(s.capitalize()) # Capitalize a string

print(s.upper()) # Convert a string to uppercase; prints "HELLO"

print(s.rjust(7)) # Right-justify a string, padding with spaces

print(s.center(7)) # Center a string, padding with spaces

print(s.replace('l', '(ell)')) # Replace all instances of one substring with anoth

print(' world '.strip()) # Strip leading and trailing whitespace

#Containers

#lists

xs = [3, 1, 2] # Create a list

print(xs, xs[2])

print(xs[-1]) # Negative indices count from the end of the list; prints "2"xs[2] = 'foo' # Lists can contain elements of different types

print(xs)

xs.append('bar') # Add a new element to the end of the list

print(xs)

x = xs.pop() # Remove and return the last element of the list

print(x, xs)

#Slicing

nums = list(range(5)) # range is a built-in function that creates a list of inte

print(nums) # Prints "[0, 1, 2, 3, 4]"

print(nums[2:4]) # Get a slice from index 2 to 4 (exclusive); prints "[2, 3]"

print(nums[2:]) # Get a slice from index 2 to the end; prints "[2, 3, 4]"

print(nums[:2]) # Get a slice from the start to index 2 (exclusive); prints "[0

print(nums[:]) # Get a slice of the whole list; prints ["0, 1, 2, 3, 4]"

print(nums[:-1]) # Slice indices can be negative; prints ["0, 1, 2, 3]"

nums[2:4] = [8, 9] # Assign a new sublist to a slice

print(nums) # Prints "[0, 1, 8, 9, 4]"

#Loops

animals = ['cat', 'dog', 'monkey']

for animal in animals:

print(animal)

animals = ['cat', 'dog', 'monkey']

for idx, animal in enumerate(animals):

print('#{}: {}'.format(idx + 1, animal))

#List Comprehensions

nums = [0, 1, 2, 3, 4]

squares = []

for x in nums:

squares.append(x \*\* 2)

print(squares)

#above code can be written as simpler using a list comprehension:

nums = [0, 1, 2, 3, 4]

squares = [x \*\* 2 for x in nums]

print(squares)

nums = [0, 1, 2, 3, 4]

even\_squares = [x \*\* 2 for x in nums if x % 2 == 0]

print(even\_squares)

#Dictionaries

d = {'cat': 'cute', 'dog': 'furry'} # Create a new dictionary with some data

print(d['cat']) # Get an entry from a dictionary; prints "cute"

print('cat' in d) # Check if a dictionary has a given key; prints "True"

d['fish'] = 'wet' # Set an entry in a dictionary

print(d['fish']) # Prints "wet"

print(d.get('monkey', 'N/A')) # Get an element with a default; prints "N/A"

print(d.get('fish', 'N/A')) # Get an element with a default; prints "wet"

del d['fish'] # Remove an element from a dictionary

print(d.get('fish', 'N/A')) # "fish" is no longer a key; prints "N/A"

d = {'person': 2, 'cat': 4, 'spider': 8}

for animal, legs in d.items():

print('A {} has {} legs'.format(animal, legs))

#Sets

animals = {'cat', 'dog'}

print('cat' in animals) # Check if an element is in a set; prints "True"

print('fish' in animals) # prints "False"

animals.add('fish') # Add an element to a set

print('fish' in animals)

print(len(animals)) # Number of elements in a set;

animals.add('cat') # Adding an element that is already in the set does nothin

print(len(animals))

animals.remove('cat') # Remove an element from a set

print(len(animals))

animals = {'cat', 'dog', 'fish'}

for idx, animal in enumerate(animals):

print('#{}: {}'.format(idx + 1, animal))

from math import sqrt

print({int(sqrt(x)) for x in range(30)})

#Tuples

d = {(x, x + 1): x for x in range(10)} # Create a dictionary with tuple keys

t = (5, 6) # Create a tuple

print(type(t))

print(d[t])

print(d[(1, 2)])

#Functions

def sign(x):

if x > 0:

return 'positive'

elif x < 0:

return 'negative'

else:

return 'zero'

for x in [-1, 0, 1]:

print(sign(x))

def hello(name, loud=False):

if loud:

print('HELLO, {}'.format(name.upper()))

else:

print('Hello, {}!'.format(name))

hello('Bob')

hello('Fred', loud=True)

#Classes

class Greeter:

# Constructor

def \_\_init\_\_(self, name):

self.name = name # Create an instance variable

# Instance method

def greet(self, loud=False):

if loud:

print('HELLO, {}'.format(self.name.upper()))

else:

print('Hello, {}!'.format(self.name))

g = Greeter('Fred') # Construct an instance of the Greeter class

g.greet() # Call an instance method; prints "Hello, Fred"

g.greet(loud=True) # Call an instance method; prints "HELLO, FRED!"