# >>> Basic Datatypes

BASIC DATATYPES

# >>> String

a = "Hello"

print(a)

a = """Lorem ipsum dolor sit amet,

consectetur adipiscing elit,

sed do eiusmod tempor incididunt

ut labore et dolore magna aliqua."""

print(a)

a = "Hello, World!"

print(a[1])

for x in "banana":

print(x)

a = "Hello, World!"

print(len(a))

txt = "The best things in life are free!"

print("free" in txt)

txt = "The best things in life are free!"

if "free" in txt:

print("Yes, 'free' is present.")

txt = "The best things in life are free!"

print("expensive" not in txt)

txt = "The best things in life are free!"

if "expensive" not in txt:

print("No, 'expensive' is NOT present.")

txt = "welcome to the jungle"

x = txt.split(" ")

print(x)

txt = "hello, my name is Peter, I am 26 years old"

x = txt.split(", ")

print(x)

a = "Hello"

b = "World"

c = a + b

print(c)

# >>> int, float, complex

x = 1 # int

y = 2.8 # float

z = 1j # complex

# int

x = 1

y = 35656222554887711

z = -3255522

print(type(x))

print(type(y))

print(type(z))

# float

x = 1.10

y = 1.0

z = -35.59

print(type(x))

print(type(y))

print(type(z))

x = 35e3

y = 12E4

z = -87.7e100

print(type(x))

print(type(y))

print(type(z))

# complex

x = 3+5j

y = 5j

z = -5j

print(type(x))

print(type(y))

print(type(z))

# >>> boolean

a=True

b=False

print(a,b)

print(bool(False))

print(bool(None))

print(bool(0))

print(bool(""))

print(bool(()))

print(bool([]))

print(bool({}))

print(bool("abc"))

print(bool(123))

print(bool(["apple", "cherry", "banana"]))

# >>> Lists

LIST DATATYPE

# Lists are used to store multiple items in a single variable.

# List items are ordered, changeable, and allow duplicate values.

mylist = ["apple", "banana", "cherry"]

print(mylist)

# Ordered

# When we say that lists are ordered, it means that the items have a defined order, and that order will not change.

# If you add new items to a list, the new items will be placed at the end of the list.

mylist.append("Mango")

print(mylist)

# Lists allow duplicate values

mylist.append("Mango")

print(mylist)

# Check length of a list

print(len(mylist))

# List items can be of any type

# homogeneous

list1 = ["apple", "banana", "cherry"]

list2 = [1, 5, 7, 9, 3]

list3 = [True, False, False]

print(list1,list2,list3)

# heterogeneous

list1 = ["abc", 34, True, 40, "male"]

print(list1)

# >>> Type of list

mylist = ["apple", "banana", "cherry"]

print(type(mylist))

# >>> Accessing values in list

thislist = ["apple", "banana", "cherry"]

print(thislist[1])

thislist = ["apple", "banana", "cherry"]

print(thislist[-1])

thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]

print(thislist[2:5])

thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]

print(thislist[:4])

thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]

print(thislist[2:])

thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]

print(thislist[-4:-1])

thislist = ["apple", "banana", "cherry"]

if "apple" in thislist:

print("Yes, 'apple' is in the fruits list")

# >>> Change list item

thislist = ["apple", "banana", "cherry"]

thislist[1] = "blackcurrant"

print(thislist)

thislist = ["apple", "banana", "cherry", "orange", "kiwi", "mango"]

thislist[1:3] = ["blackcurrant", "watermelon"]

print(thislist)

thislist = ["apple", "banana", "cherry"]

thislist[1:2] = ["blackcurrant", "watermelon"]

print(thislist)

thislist = ["apple", "banana", "cherry"]

thislist[1:3] = ["watermelon"]

print(thislist)

thislist = ["apple", "banana", "cherry"]

thislist.insert(2, "watermelon")

print(thislist)

# >>> Add item to list

thislist = ["apple", "banana", "cherry"]

thislist.append("orange")

print(thislist)

thislist = ["apple", "banana", "cherry"]

thislist.insert(1, "orange")

print(thislist)

thislist = ["apple", "banana", "cherry"]

tropical = ["mango", "pineapple", "papaya"]

thislist.extend(tropical)

print(thislist)

# >>> Remove item from list

thislist = ["apple", "banana", "cherry"]

thislist.remove("banana")

print(thislist)

thislist = ["apple", "banana", "cherry"]

thislist.pop(1)

print(thislist)

thislist = ["apple", "banana", "cherry"]

thislist.pop()

print(thislist)

thislist = ["apple", "banana", "cherry"]

del thislist[0]

print(thislist)

thislist = ["apple", "banana", "cherry"]

del thislist

thislist = ["apple", "banana", "cherry"]

thislist.clear()

print(thislist)

# >>> Loop a list

thislist = ["apple", "banana", "cherry"]

for x in thislist:

print(x)

# >>> Sort list

thislist = ["orange", "mango", "kiwi", "pineapple", "banana"]

thislist.sort()

print(thislist)

thislist = [100, 50, 65, 82, 23]

thislist.sort()

print(thislist)

thislist = ["orange", "mango", "kiwi", "pineapple", "banana"]

thislist.sort(reverse = True)

print(thislist)

thislist = [100, 50, 65, 82, 23]

thislist.sort(reverse = True)

print(thislist)

# >>> Tuple

TUPLE DATATYPE

# Tuples are used to store multiple items in a single variable.

# A tuple is a collection which is ordered and unchangeable.

thistuple = ("apple", "banana", "cherry")

print(thistuple)

# Tuples allow duplicate values:

thistuple = ("apple", "banana", "cherry", "apple", "cherry")

print(thistuple)

# Length of a tuple

thistuple = ("apple", "banana", "cherry")

print(len(thistuple))

tuple1 = ("apple", "banana", "cherry")

tuple2 = (1, 5, 7, 9, 3)

tuple3 = (True, False, False)

print(tuple1,tuple2,tuple3)

tuple1 = ("abc", 34, True, 40, "male")

print(tuple1)

mytuple = ("apple", "banana", "cherry")

print(type(mytuple))

# >>> access tuple

thistuple = ("apple", "banana", "cherry")

print(thistuple[1])

thistuple = ("apple", "banana", "cherry")

print(thistuple[-1])

thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")

print(thistuple[2:5])

thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")

print(thistuple[:4])

thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")

print(thistuple[2:])

thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")

print(thistuple[-4:-1])

thistuple = ("apple", "banana", "cherry")

if "apple" in thistuple:

print("Yes, 'apple' is in the fruits tuple")

# >>> Changing tuple

# Once a tuple is created, you cannot change its values.

# Tuples are unchangeable, or immutable as it also is called.

# But there is a workaround.

# You can convert the tuple into a list, change the list, and convert the list back into a tuple.

x = ("apple", "banana", "cherry")

y = list(x)

y[1] = "kiwi"

x = tuple(y)

print(x)

# >>> unpack tuple

fruits = ("apple", "banana", "cherry")

(green, yellow, red) = fruits

print(green)

print(yellow)

print(red)

# >>> loop tuple

thistuple = ("apple", "banana", "cherry")

for x in thistuple:

print(x)

# >>> Set

SET DATATYPE

# Sets are used to store multiple items in a single variable.

# A set is a collection which is unordered, unchangeable\*, and unindexed.

myset = {"apple", "banana", "cherry"}

thisset = {"apple", "banana", "cherry"}

print(thisset)

thisset = {"apple", "banana", "cherry", "apple"}

print(thisset)

thisset = {"apple", "banana", "cherry"}

print(len(thisset))

set1 = {"apple", "banana", "cherry"}

set2 = {1, 5, 7, 9, 3}

set3 = {True, False, False}

print(set1,set2,set3)

set1 = {"abc", 34, True, 40, "male"}

print(set1)

thisset = set(("apple", "banana", "cherry")) # note the double round-brackets

print(thisset)

thisset = {"apple", "banana", "cherry"}

for x in thisset:

print(x)

# >>> Add element

thisset = {"apple", "banana", "cherry"}

thisset.add("orange")

print(thisset)

thisset = {"apple", "banana", "cherry"}

tropical = {"pineapple", "mango", "papaya"}

thisset.update(tropical)

print(thisset)

thisset = {"apple", "banana", "cherry"}

mylist = ["kiwi", "orange"]

thisset.update(mylist)

print(thisset)

# >>> Remove element

thisset = {"apple", "banana", "cherry"}

thisset.remove("banana")

print(thisset)

thisset = {"apple", "banana", "cherry"}

thisset.discard("banana")

print(thisset)

thisset = {"apple", "banana", "cherry"}

x = thisset.pop()

print(x)

print(thisset)

thisset = {"apple", "banana", "cherry"}

thisset.clear()

print(thisset)

thisset = {"apple", "banana", "cherry"}

del thisset

print(thisset)

# >>> Dictionary

DICTIONARY DATATYPE

# Dictionaries are used to store data values in key:value pairs.

thisdict = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

print(thisdict)

thisdict = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

print(thisdict["brand"])

# Duplicate values will overwrite existing values:

thisdict = {

"brand": "Ford",

"model": "Mustang",

"year": 1964,

"year": 2020

}

print(thisdict)

# length of a list

print(len(thisdict))

# The values in dictionary items can be of any data type:

thisdict ={

"brand": "Ford",

"electric": False,

"year": 1964,

"colors": ["red", "white", "blue"]

}

thisdict ={

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

print(type(thisdict))

# >>> Access items

thisdict ={

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

x = thisdict["model"]

x = thisdict.get("model")

x = thisdict.keys()

print(x)

# >>> Change value

car = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

x = car.keys()

print(x) #before the change

car["color"] = "white"

print(x) #after the change

# >>> Get all values

x = thisdict.values()

car = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

x = car.values()

print(x) #before the change

car["year"] = 2020

print(x) #after the change

# >>> check if value exist in dictionary

thisdict = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

if "model" in thisdict:

print("Yes, 'model' is one of the keys in the thisdict dictionary")

# >>> Change items

thisdict = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

thisdict.update({"year": 2020})

# add new item

thisdict = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

thisdict.update({"color": "red"})

# >>> Remove item from dictionary

thisdict = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

thisdict.pop("model")

print(thisdict)

thisdict = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

del thisdict["model"]

print(thisdict)

thisdict = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

thisdict.clear()

print(thisdict)

# >>> Loop through dictionary

for x in thisdict:

print(thisdict[x])

for x in thisdict.values():

print(x)

for x in thisdict.keys():

print(x)

for x, y in thisdict.items():

print(x, y)

# >>> Nested Dictionary

myfamily = {

"child1" : {

"name" : "Emil",

"year" : 2004

},

"child2" : {

"name" : "Tobias",

"year" : 2007

},

"child3" : {

"name" : "Linus",

"year" : 2011

}

}

print(myfamily["child1"])

# >>> Range

DICTIONARY DATATYPE

# The range() function returns a sequence of numbers,

# starting from 0 by default, and increments by 1 (by default),

# and stops before a specified number.

x = range(6)

for n in x:

print(n)

x = list(range(6))

for n in x:

print(n)

x = range(3, 6)

for n in x:

print(n)

x = range(3, 20, 2)

for n in x:

print(n)