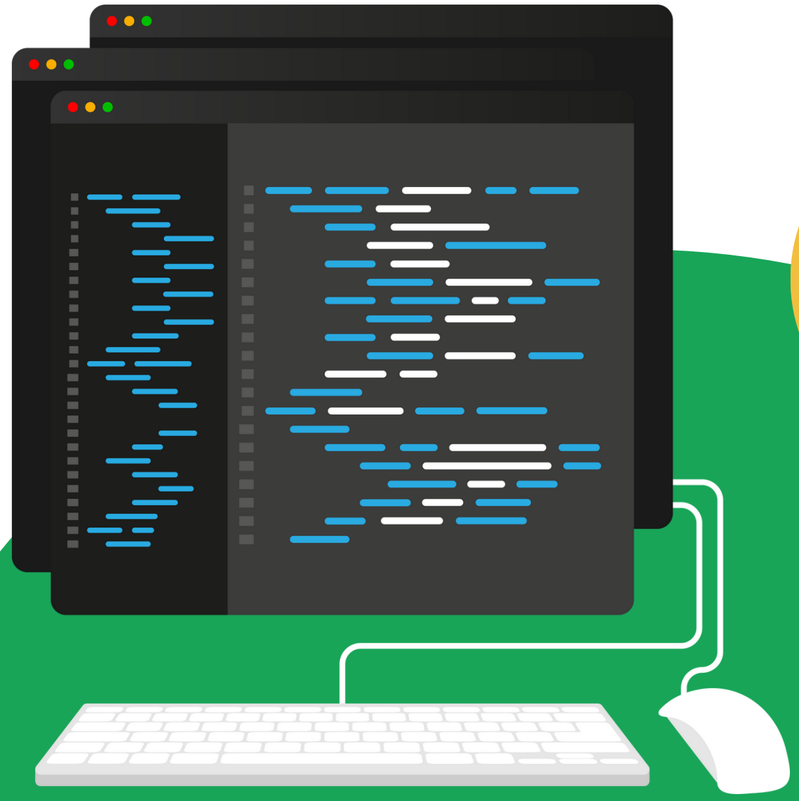
COMPUTER PROGRAMMING 1

LECTURE

PREPARED BY: MS. JHONA VALLESTERO



GOOD MORNING!
WELCOME TO OUR CLASS

PRAYER

CHECKING OF ATTENDANCE

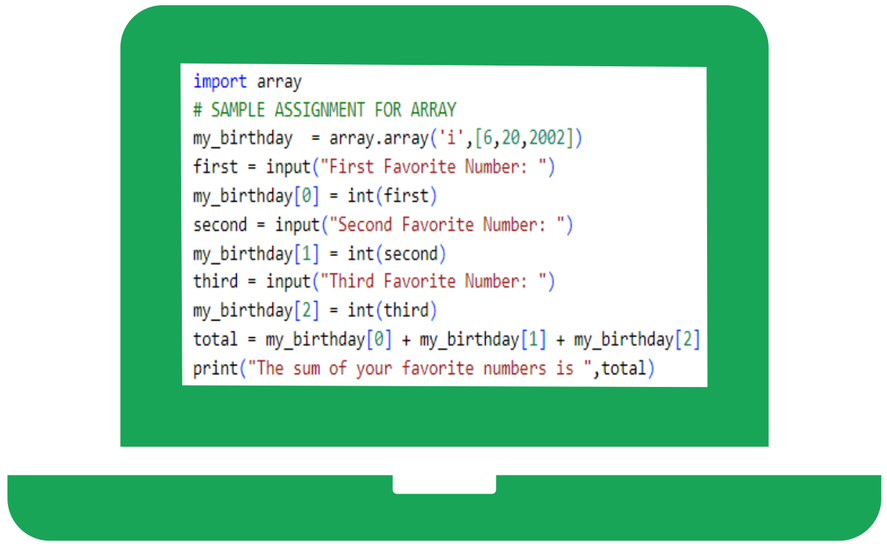
ENERGIZER

LET US HAVE A REVIEW

LET US HAVE A REVIEW

1. What is the list and tuples?
2. What is the difference of list and tuples?

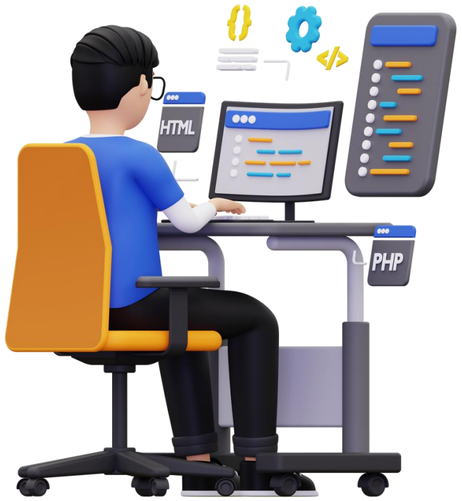
PYTHON: DICTIONARIES AND ARRAYS



Objectives

At the end of the lesson, the students should be able to:

1. Understand the use of python dictionaries and arrays;
2. Differentiate the python list and python arrays;
3. Read a program with dictionaries and arrays; and
4. Create a program using dictionaries and arrays.



Introduction

In Python, dictionaries and arrays are both types of data structures used to store and manipulate collections of data, but they have different characteristics and uses.

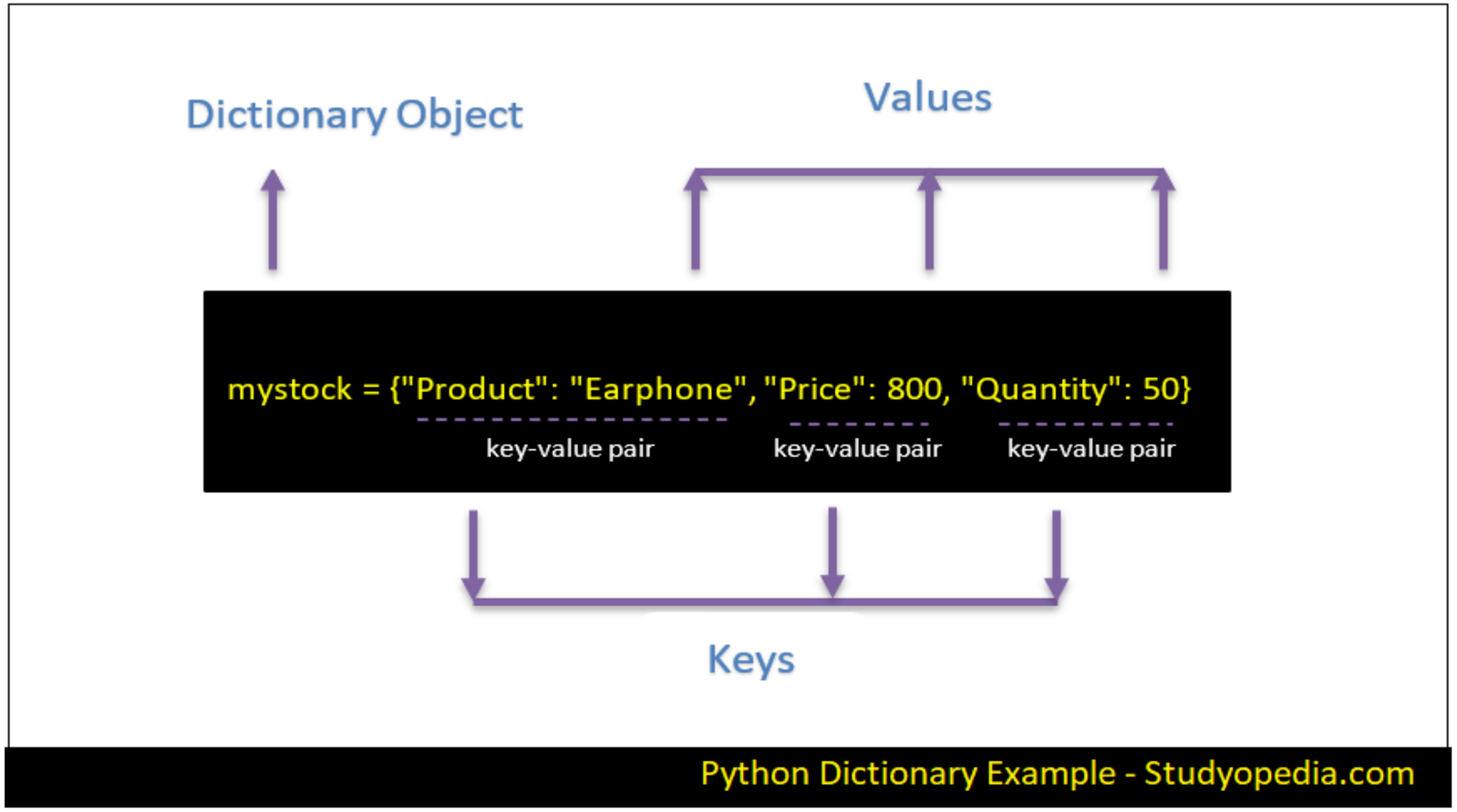
Python Dictionaries

Python Dictionaries

- are used to store data values in key : value pairs
- is a collection which is ordered*, changeable and do not allow duplicates

As of Python version 3.7, dictionaries are ordered. In Python 3.6 and earlier, dictionaries are unordered.

Components of a Dictionaries



Each key is separated from its value by a colon (:), the items are separated by commas, and the whole thing is enclosed in curly braces. An empty dictionary without any items is written with just two curly braces, like this: {}.

Python Dictionaries Operators

d1 = {'a': 2, 'b': 4, 'c': 30}

d2 = {'a1': 20, 'b1': 40, 'c1': 60}

Operator	Description	Example 
dict[key]	Extract/assign the value mapped with key	print (d1['b']) retrieves 4 d1['b'] = 'Z' assigns new value to key 'b'
dict1|dict2	Union of two dictionary objects, retuning new object	d3=d1|d2 ; print (d3) {'a': 2, 'b': 4, 'c': 30, 'a1': 20, 'b1': 40, 'c1': 60}
dict1|=dict2	Augmented dictionary union operator	d1|=d2; print (d1) {'a': 2, 'b': 4, 'c': 30, 'a1': 20, 'b1': 40, 'c1': 60}

HOW TO.....

CREATE

READ/ACCESS

UPDATE

DELETING

ADD ITEMS

CREATE

 # Creating an empty Dictionary

student = {}

student = dict()

 # Creating a Dictionary with dict() method

student = dict({'Name':'Jhona','Course':'BTVTEd-CP'})

 # Dictionary from List of Tuples

student = dict([('Name','Jhona'),('Course','BTVTEd-CP')])

 # Dictionary from Keyword Arguments

student = dict('Name'='Jhona','Course'='BTVTEd-CP')

READ/ACCESS

UPDATE

DELETING

ADDITEMS

CREATE

UPDATE

DELETING

ADDITEMS

READ/ACCESS

- referring to **its key name**, inside square brackets

  x = student['Name']

- **get()**

  x = student.get('Name')

- **key()** - return a list of all the keys in the dictionary.

  x = student.keys()

- **value()** - return a list of all the values in the dictionary.

  x = student.values()

- **item()** - return each item in a dictionary, as tuples in a list.

  x = student.items()

CREATE

READ/ACCESS

DELETING

ADDITEMS

UPDATE

- **referring to its key name**

  student['Name'] = 'Justin'

- **update**() - update the dictionary with the items from the given argument. The argument must be a dictionary, or an iterable object with key:value pairs.

  student.update({'Name':'Justin'})

CREATE READ/ACCESS UPDATE ADDITEMS

DELETING

- **pop()** - removes the item with the specified key name.

  student.pop('Name')

  student.pop() # removes the last index

- **del keyword**

  -> **removes** the item with the specified **key name**.

  del student['Name']

  -> can also **delete** the dictionary **completely**.

  del student

- **clear()** - empties the dictionary.

  student.clear()

CREATE

READ/ACCESS

UPDATE

DELETING

ADD ITEMS

- *using* ***update() method*** - update the dictionary with the items from a given argument. If the item does not exist, the item will be added.

  student.update({'Age':21})

  student.update([('Age',21),('Gender','Female')])

  student.update('Age'=21),'Gender'='Female')

- using a **new index key** and assigning a value to it

  student['Age'] = 21

IMPORTANT POINTS TO REMEMBER

- ❑ More than one entry per key not allowed.
- ❑ Keys must be immutable.

PYTHON DICTIONARIES FUNCTIONS AND METHODS

Dictionary Functions	
`cmp(dict1, dict2)`	Compares elements of both dict.
`len(dict)`	Gives the total length of the dictionary. This would be equal to the number of items in the dictionary.
`str(dict)`	Produces a printable string representation of a dictionary.
`type(variable)`	Returns the type of the passed variable. If passed variable is dictionary, then it would return a dictionary type.

FUNCTIONS

Dictionary Methods	
`clear()`	Removes all elements of dictionary dict
`copy()`	Returns a shallow copy of dictionary dict
`fromkeys()`	Create a new dictionary with keys from seq and values set to value.
`get(key, default=None)`	For key key, returns value or default if key not in dictionary
`has_key(key)`	Returns true if key in dictionary dict, false otherwise

METHODS

items()	Returns a list of dict's (key, value) tuple pairs
keys()	Returns list of dictionary dict's keys
setdefault(key, default=None)	Similar to get(), but will set dict[key]=default if key is not already in dict
update(dict2)	Adds dictionary dict2's key-values pairs to dict
values()	Returns list of dictionary dict's values

METHODS

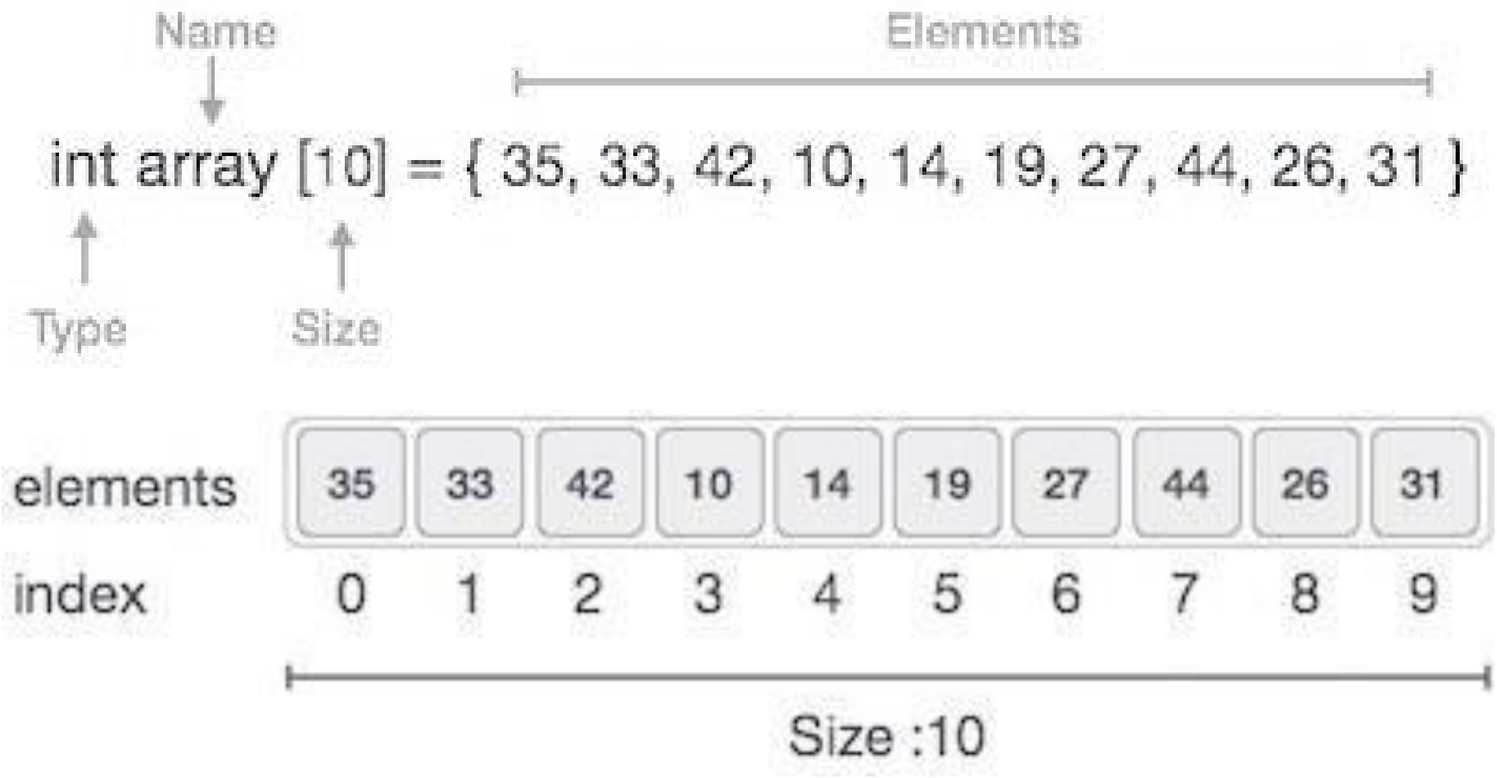
Python Arrays

Python Arrays

- Arrays are containers which are able to store more than one item at the same time.
- Arrays are an ordered collection of elements with every value being of the same data type.
- Python's standard library has array module. The array class in it allows you to construct an array of three basic types, integer, float and Unicode characters.

Arrays can be declared in various ways in different languages. Below is an illustration.

JAVA



- Index starts with 0.
- Array length is 10, which means it can store 10 elements.
- Each element can be accessed via its index.

Arrays can be declared in various ways in different languages. Below is an illustration.

PYTHON

```
import array
obj_name = array.array(typecode,[initializer])
```

- **typecode** – are the codes that are used to define the type of value the array will hold.
- **initializer** – array initialized from the optional value, which must be a list, a bytes-like object, or iterable over elements of the appropriate type.

Arrays can be declared in various ways in different languages. Below is an illustration.

TYPECODE

Typecode	Value
b	Represents signed **integer** of size 1 byte
B	Represents unsigned **integer** of size 1 byte
u	Represents unicode **character** of size 2 byte
i	Represents signed **integer** of size 2 bytes
I	Represents unsigned **integer** of size 2 bytes
f	Represents **floating** point of size 4 bytes
d	Represents **floating** point of size 8 bytes

HOW TO.....

INSERT

DELETE

SEARCH/ACCESS

UPDATE

INSERT

Adds an element at the given index

- Using **insert() method**

  num.insert(1,4) # insert(index,value/element)

- Using **append() method**

  num.append(8) # append(element to be added)

DELETE SEARCH/ACCESS UPDATE

DELETE

Deletes an element at the given index

- Using **pop() method**

```
num.pop(2) # pop(index)
```

- Using **remove() method**

```
num.remove(2) # remove(element)
```

- Using **clear() method**

```
name.clear() # will return an error
```

INSERT

DELETE

SEARCH/ACCESS

Searches an element using the given index or by the value

- Using **index() method** – will return the index

  name.index(4) # index(value/element)

- Get the value **using index number**

  x = num[1]

UPDATE

INSERT

DELETE

SEARCH/ACCESS

UPDATE

We simply reassign a new value to the desired index we want to update.

num[2] = 8

PYTHON
ARRAY METHODS

Method	Description
append()	Adds an element at the end of the list
clear()	Removes all the elements from the list
copy()	Returns a copy of the list
count()	Returns the number of elements with the specified value
extend()	Add the elements of a list (or any iterable), to the end of the current list
index()	Returns the index of the first element with the specified value
insert()	Adds an element at the specified position
pop()	Removes the element at the specified position
remove()	Removes the first item with the specified value
reverse()	Reverses the order of the list
sort()	Sorts the list

ARRAY METHODS

REFERENCES

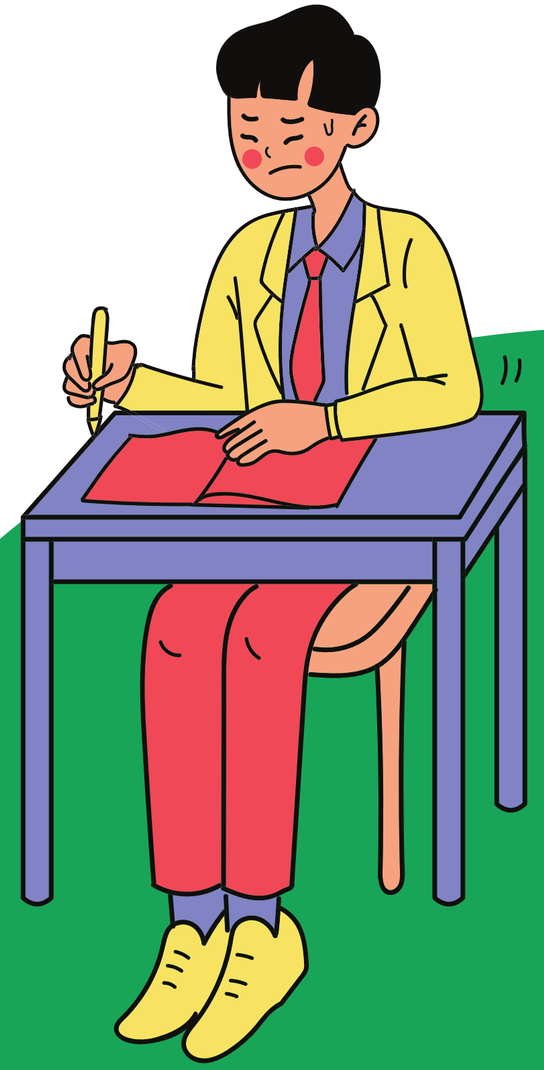
https://www.tutorialspoint.com/python/python_dictionary.htm

https://www.w3schools.com/python/python_dictionaries.asp

https://www.tutorialspoint.com/python/python_arrays.htm

https://www.w3schools.com/python/python_arrays.asp

QUIZ TIME



THANK YOU FOR LISTENING