**Om Pawaskar**

**S22-96**

**PY LAB**

**DICTIONARY IN PYTHON**

Example = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

print(Example["brand"])

Ford

#Duplicate values will overwrite existing values:

Example= {

"brand": "Ford",

"model": "Mustang",

"year": 1964,

"year": 2020

} print(Example) print(len(Example)) print(type(Example))

{'brand': 'Ford', 'model': 'Mustang', 'year': 2020}

3

<class 'dict'>

Example = dict(name = "Nikhilesh", age = 47, country = "India") print(Example)

Example = dict(name = "SHLOK", age = 14, country = "India") print(Example)

{'name': 'Nikhilesh', 'age': 47, 'country': 'India'} {'name': 'SHLOK', 'age': 14, 'country': 'India'}

Example = {

"brand": "Ford",

"model": "Mustang",

"year": 1964 } x = Example["model"]

print(x)

#There is also a method called get() that will give you the same result: x =

Example.get("model") print(x)

#Add a new item to the original dictionary, and see that the keys list gets updated as well:

car = {

"brand": "Ford",

"model": "Mustang",

"year": 1964 } x = car.keys()

print(x) #before the change car["color"] = "white" print(x) #after the change

car = {"brand": "Ford","model": "Mustang","year":

1964} x = car.values() print(x) #before the change car["year"] = 2020 print(x) #after the change

#Add a new item to the original dictionary, and see that the values list gets updated as well: car = { "brand": "Ford","model": "Mustang","year": 1964} x = car.values()

print(x) #before the change car["color"] = "red" print(x) #after the change

#The items() method will return each item in a dictionary, as tuples in a list. x = Example.items() print(x)

#Make a change in the original dictionary, and see that the items list gets updated as well: car = {"brand": "Ford","model": "Mustang","year": 1964} x = car.items()

print(x) #before the change car["year"] = 2020 print(x) #after the change

#Add a new item to the original dictionary, and see that the items list gets updated as well: car = {"brand": "Ford","model": "Mustang","year": 1964} x = car.items()

print(x) #before the change car["color"] = "red" print(x) #after the change

#Check if "model" is present in the dictionary:

Example = {"brand": "Ford","model": "Mustang","year": 1964} if "model" in Example: print("Yes, 'model' is one of the keys in the Example dictionary")

Mustang Mustang

dict\_keys(['brand', 'model', 'year']) dict\_keys(['brand', 'model', 'year']) dict\_keys(['brand', 'model', 'year', 'color']) dict\_values(['Ford', 'Mustang', 1964]) dict\_values(['Ford', 'Mustang', 2020]) dict\_values(['Ford', 'Mustang', 1964])

dict\_values(['Ford', 'Mustang', 1964, 'red'])

dict\_items([('brand', 'Ford'), ('model', 'Mustang'), ('year', 1964)]) dict\_items([('brand', 'Ford'), ('model', 'Mustang'), ('year', 1964)]) dict\_items([('brand', 'Ford'), ('model', 'Mustang'), ('year', 2020)]) dict\_items([('brand', 'Ford'), ('model', 'Mustang'), ('year', 1964)]) dict\_items([('brand', 'Ford'), ('model', 'Mustang'), ('year', 1964), ('color', 'red')])

Yes, 'model' is one of the keys in the Example dictionary

#Change Values

#Change the "year" to 2018:

Example = { "brand": "Ford", "model": "Mustang", "year": 1964} print (Example) Example["year"] = 2018 print(Example)

#update Method

#Update the "year" of the car by using the update() method: Example = {"brand": "Ford","model": "Mustang","year": 1964} print(Example)

Example.update({"year": 2020}) print(Example)

{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}

{'brand': 'Ford', 'model': 'Mustang', 'year': 2018} {'brand': 'Ford', 'model': 'Mustang', 'year': 1964}

{'brand': 'Ford', 'model': 'Mustang', 'year': 2020}

#Remove Items

#The pop() method removes the item with the specified key name:

Example = {"brand": "Ford","model": "Mustang","year": 1964} Example.pop("model") print(Example)

{'brand': 'Ford', 'year': 1964}

#The popitem() method removes the last inserted item

Example = {"brand": "Ford","model": "Mustang","year": 1964} Example.popitem() print(Example)

{'brand': 'Ford', 'model': 'Mustang'}

#The del keyword removes the item with the specified key name: Example = {"brand": "Ford","model": "Mustang","year": 1964} del Example["model"]

print(Example)

{'brand': 'Ford', 'year': 1964}

#The clear() method empties the dictionary:

Example = {"brand": "Ford","model": "Mustang","year": 1964} Example.clear()

print(Example)

{}

#Loop through

#Print all key names in the dictionary, one by one: Example = {"brand": "Ford","model": "Mustang","year": 1964} for x in Example: print(x)

#Print all values in the dictionary, one by one print("PART TWO")

Example = {"brand": "Ford","model": "Mustang","year": 1964} for x in Example:

print(Example[x])

#You can also use the values() method to return values of a dictionary:

print("PART THREE")

for x in Example.values(): print(x)

#You can use the keys() method to return the keys of a dictionary:

print("PART FOUR")

for x in Example.keys(): print(x)

#Loop through both keys and values, by using the items() method:

print("PART FIVE")

Example = {"brand": "Ford","model": "Mustang","year": 1964} for x, y in Example.items(): print(x, y)

brand model year

PART TWO

Ford

Mustang

1964

PART THREE Ford

Mustang

1964 PART FOUR brand model year PART FIVE brand Ford model Mustang year 1964

# Copy a Dictionary

Example = {"brand": "Ford","model": "Mustang","year": 1964} print(Example)

mydict = Example.copy() print(mydict)

#Make a copy of a dictionary with the dict() function: Example = {"brand": "Ford","model": "Mustang","year": 1964} mydict = dict(Example)

print(mydict)

{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}

{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}

{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}

# Nested Dictionaries

myfamily = {"child1" : { "name" : "Jay","year" : 2004},"child2" : {"name" : "Titan","year" : 2007}, "child3" : {"name" : "Leo","year" : 2011 print(myfamily) print(myfamily["child2"]["name"])

{'child1': {'name': 'Jay', 'year': 2004}, 'child2': {'name': 'Titan',

'year': 2007}, 'child3': {'name': 'Leo', 'year': 2011}} Titan

# Dictionary Methods

#Remove all elements from the car list: car = {"brand": "Ford","model": "Mustang","year": 1964} print(car) car.clear() print(car) print("SECOND PART")

#Copy the car dictionary:

car = {"brand": "Ford","model": "Mustang","year": 1964}

x = car.copy()

print(x) print("THIRD PART")

car = {"brand": "Ford","model": "Mustang","year": 1964}

x = car.get("model") print(x) print("PART FIVE") #Return the dictionary's key-value pairs: car = {"brand": "Ford","model": "Mustang","year": 1964}

x = car.items()

print(x)

print("PART SIX")

#When an item in the dictionary changes value, the view object also gets updated: car = {"brand": "Ford","model": "Mustang","year": 1964}

x = car.items()

car["year"] = 2018 print(x) print("PART SEVEN")

car {"brand": "Ford" "model": "Mustang" "year": 1964}

car = { brand : Ford , model : Mustang , year : 1964} x = car.setdefault("color", "white")

print(x)

{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}

{}

SECOND PART

{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}

THIRD PART

Mustang PART FIVE dict\_items([('brand', 'Ford'), ('model', 'Mustang'), ('year', 1964)]) PART SIX dict\_items([('brand', 'Ford'), ('model', 'Mustang'), ('year', 2018)]) PART SEVEN white