Dictionaries

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

A dictionary is a collection which is ordered, changeable and do not allow duplicates.

In [1]:
```
d={}
print(type(d))
```
```
<class 'dict'>
```

In [40]:
```
d1={"name":"mamatha","stream":"VLSI-DV"}
print(d1)
print(type(d1))
```
```
{'name': 'mamatha', 'stream': 'VLSI-DV'}
<class 'dict'>
```

In [3]:
```
d={}
d["name"]="mamatha"
d["class"]="python"
d["time"]=9.301
print(d)
```
```
{'name': 'mamatha', 'class': 'python', 'time': 9.301}
```

In [4]:
```
d3=dict({})
print(type(d3))
```
```
<class 'dict'>
```

In [5]:
```
d3=dict(course="python",students=5)
print(d3)
```
```
{'course': 'python', 'students': 5}
```

In [6]:
```
d3=dict(Name="Sachin",Age=40)
print(d3)
```
```
{'Name': 'Sachin', 'Age': 40}
```

In [7]:
```
a=dict([['name','Sachin'],['age',40]])
print(a)
```
```
{'name': 'Sachin', 'age': 40}
```

In [39]:
```
a=dict([['name','Sachin'],['age',45]])
print(a)
```
```
{'name': 'Sachin', 'age': 45}
```

In [41]:
```
a=dict([['name','Sachin'],['age',40]])
print(a)
```
```
{'name': 'Sachin', 'age': 40}
```

In [9]:
```
# duplicaints are not allowed!
# duplicate values will overwrite existing values:
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964,
  "year": 2020
}
print(thisdict)
print(len(thisdict))
```
```
{'brand': 'Ford', 'model': 'Mustang', 'year': 2020}
3
```

In [10]:
```
# accessing the elements
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
x = thisdict["model"] # 1st method
print(x)
y = thisdict.get("model") #2nd method using get()
print(y)
```
```
Mustang
Mustang
```

In [11]:
```
#GET KEYS:
  # The keys() method will return a list of all the keys in the dictionary.
  thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
  }
  x = thisdict.keys()
  print(x)

dict_keys(['brand', 'model', 'year'])
```

In [16]:
```
car = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
  }
x = car.keys()
print(x) #before the change
car["color"] = "white"
print(x) #after the change

dict_keys(['brand', 'model', 'year'])
dict_keys(['brand', 'model', 'year', 'color'])
```

In [17]:
```
# get value:
  # The values() method will return a list of all the values in the dictionary.
  thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
  }
  x = thisdict.values()
  print(x)

dict_values(['Ford', 'Mustang', 1964])
```

In [18]:
```
car = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
  }
x = car.values()
print(x) #before the change
car["year"] = 2020
print(x) #after the change

dict_values(['Ford', 'Mustang', 1964])
dict_values(['Ford', 'Mustang', 2020])
```

In [19]:
```
# GET ITEMS
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
x = thisdict.items()
print(x)

dict_items([('brand', 'Ford'), ('model', 'Mustang'), ('year', 1964)])
```

In [21]:
```
# check the key value exist or not
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
if "model" in thisdict:
  print("Yes, 'model' is one of the keys in the thisdict dictionary")

Yes, 'model' is one of the keys in the thisdict dictionary
```

In [22]:
```
# change the variable
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict["year"] = 2018
print(thisdict)

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

In [23]:
```
# update:
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict.update({"year": 2020})
print(thisdict)

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

In [24]:
```
#add the elements:
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict["color"] = "red"
print(thisdict)

{'brand': 'Ford', 'model': 'Mustang', 'year': 1964, 'color': 'red'}
```

In [25]:
```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict.update({"color": "red"})
print(thisdict)

{'brand': 'Ford', 'model': 'Mustang', 'year': 1964, 'color': 'red'}
```

In [26]:
```
#pop
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict.pop("model")
print(thisdict)

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

In [45]:
```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict.popitem()
#The popitem() method removes the last inserted key-value pair
print(thisdict)

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

In [28]:
```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
del thisdict["model"]
print(thisdict)

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

In [29]:
```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
del thisdict
print(thisdict) #this will cause an error because "thisdict" no longer exists.
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[29], line 7
      1 thisdict = {
      2   "brand": "Ford",
      3   "model": "Mustang",
      4   "year": 1964
      5 }
      6 del thisdict
----> 7 print(thisdict)

NameError: name 'thisdict' is not defined
```

In [30]:
```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict.clear()
print(thisdict)
```
```
{}
```

In [32]:
```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
for x in thisdict:
  print(x)
print()
for x in thisdict:
  print(thisdict[x])

brand
model
year

Ford
Mustang
1964
```

In [35]:
```
# add a key-value pair to the dict
key=int(input("Enter the key (int) to be added:"))
value=int(input("Enter the value for the key to be added:"))
d={}
d.update({key:value})
print("Updated dictionary is:")
print(d)

Updated dictionary is:
{5: 10}
```

In [36]:
```
# python program to check if a given key exists in a dic or not:
dic={2:4,3:9,4:16,5:25,6:36}
print(dic)
key=int(input("enter the key:"))
if key in dic:
    print("yes, it corresponding value is",dic[key])
else:
    print("No")

yes, it corresponding value is 4
```

In [37]:
```
# python program to concatenate two dic into one.
dic1={1:2,4:3,9:4,16:5}
dic2={5:6,9:36,7:49,8:64}
dic1.update(dic2)
print(dic1)

{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}
```

In [38]:
```
# n,n^2
n=int(input("enter the value :"))
d={}
for x in range(1,n+1):
    d[x]=x*x
print(d)

{1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
```

In [39]:
```
# Python Program to Sum All the Items in a Dictionary
d={"a":100,"b":25,"c":36,"d":41}
print(sum(d.values()))

200
```

In [40]:
```
d={"a":100,"b":25,"c":36,"d":41}
sum=0
for i in d:
    sum=sum+d[i]
print(sum)

200
```

In [42]:
```
# multiplications
d={"a":100,"b":25,"c":36,"d":41}
mul=1
for i in d:
    mul=mul*d[i]
print(mul)

3150000
```

In [43]:
```
keys=[]
values=[]
n=int(input("Enter number of elements for dictionary:"))
print("For keys:")
for x in range(0,n):
    keys.append(int(input("Enter element" + str( x+1) + ":")))
    keys.append(element)
    print("For values:")
for x in range(0,n):
    element=int(input("Enter element" + str( x+1) + ":"))
    values.append(element)
d=dict(zip(keys,values))
print("The dictionary is:")
print(d)

For keys:
For values:
For values:
For values:
For values:
For values:
The dictionary is:
{1: 12, 2: 24, 3: 36, 4: 48, 5: 60}
```

In [44]:
```
#program:
d={'a':1,'b':2,'c':3}
key=input("Enter key to check:")
if key in d.keys():
    print("Key is present and value of the key is:")
    print(d[key])
else:
    print("Key isn't present!")

Key is present and value of the key is:
2
```

In [45]:
```
d1={'A':1,'B':2}
d2={'C':3}
d1.update(d2)
print("Concatenated dictionary is:")
print(d1)

Concatenated dictionary is:
{'A': 1, 'B': 2, 'C': 3}
```

In [46]:
```
#Python Program to Remove the Given Key from a Dictionary
d={'a':1,'b':2,'c':3,'d':4}
print("Initial dictionary")
print(d1e1)
key=input("Enter the key to delete(a-d):")
if key in d1e1:
    del d1e[key]
else:
    print("Key not found!")
    exit(0)
print("Updated dictionary")
print(d1e1)

Initial dictionary
{'a': 1, 'b': 2, 'c': 3, 'd': 4}
Enter the key to delete(a-d):1
Key not found!
```

In [20]:
```
courses_info = {
    "CS101": ["Intro to Computer Science", "Credits": 3, "Instructor": "Dr. Smith"},
    "MATH201": {"Title": "Calculus II", "Credits": 4, "Instructor": "Dr. Johnson"},
    "ENG102": {"Title": "English Literature", "Credits": 3, "Instructor": "Dr. Brown"}
}

# Print course titles and their instructors
for course_code, info in courses_info.items():
    print(f"{info['Title']} is taught by {info['Instructor']}")
```
```
Intro to Computer Science is taught by Dr. Smith
Calculus II is taught by Dr. Johnson
English Literature is taught by Dr. Brown
```

In [35]:
```
weather_forecast = {
    "Monday": {"Temperature": 75, "Condition": "Sunny"},
    "Tuesday": {"Temperature": 72, "Condition": "Partly Cloudy"},
    "Wednesday": {"Temperature": 70, "Condition": "Rainy"}
}

# Print weather forecast for the week
for day, forecast in weather_forecast.items():
    print(f"{day}: {forecast['Condition']}, {forecast['Temperature']}F")
```
```
Monday: Sunny, 75F
Tuesday: Partly Cloudy, 72F
Wednesday: Rainy, 70F
```

In [34]:
```
student_grades = {
    "Alice": {"Math": 85, "Science": 92, "English": 88},
    "Bob": {"Math": 78, "Science": 74, "English": 81},
    "Charlie": {"Math": 95, "Science": 89, "English": 94}
}

# Calculate the average grade for each student
average_grades = {}
for student, grades in student_grades.items():
    total = sum(grades.values())
    count = len(grades)
    average_grades[student] = total / count

print("Average grades:", average_grades)
```
```
Average grades: {'Alice': 88.33333333333333, 'Bob': 77.66666666666667, 'Charlie': 92.66666666666667}
```

In [27]:
```
attendance_record = {
    "Monday": {"Alice", "Bob", 22, "Charlie": 18},
    "Tuesday": {"Alice": 20, "Bob": 21, "Charlie": 19},
    "March": {"Alice": 22, "Bob": 20, "Charlie": 19}
}

# Calculate total attendance for each employee over three months
total_attendance = {}
for month, records in attendance_record.items():
    for employee, days in records.items():
        if employee not in total_attendance:
            total_attendance[employee] = 0
        total_attendance[employee] += days

print("Total attendance by employee:", total_attendance)

Total attendance by employee: {'Alice': 61, 'Bob': 63, 'Charlie': 57}
```

In [33]:
```
team_performance = {
    "Team1": {"Wins": 10, "Losses": 5, "Draws": 2},
    "Team2": {"Wins": 8, "Losses": 6, "Draws": 3},
    "Team3": {"Wins": 12, "Losses": 3, "Draws": 2}
}

# Calculate the total number of matches played by each team
total_matches = {}
for team, record in team_performance.items():
    total = record["Wins"] + record["Losses"] + record["Draws"]
    total_matches[team] = total

print("Total matches played by team:", total_matches)

Total matches played by team: {'Team1': 17, 'Team2': 17, 'Team3': 17}
```

In [31]:
```
seating_arrangement = {
    "Row1": ["Alice", "Bob", "Charlie"],
    "Row2": ["David", "Eve", "Frank"],
    "Row3": ["Grace", "Heidi", "Ivan"]
}

# Print the students in each row
for row, students in seating_arrangement.items():
    print("Students in {row}: {', '.join(students)}")
```

```
Students in Row1: Alice, Bob, Charlie
Students in Row2: David, Eve, Frank
Students in Row3: Grace, Heidi, Ivan
```

In [ ]: