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]:
```
# a dictionary is a key value pair
height = {
    "Faith":160,
    "Tim":180,
    "Tom":175
}

#extract the data
height["Faith"]
```

Out[1]:
```
160
```

In [ ]:
```
## Exercise On Dictionaries
```

In [35]:
```
# age in years, height in cm, weight in kg
age = [25, 25, 27, 29, 30, 23, 25, 28, 29, 30, 30, 27]
height = [160, 165, 170, 160, 165, 160, 168, 155, 167, 169, 169, 175]
weight = [55, 60, 58, 58, 69, 49, 57, 60, 55, 61, 61, 65]

friends = {
    "Nancy":[25,160, 55],
    "Steve":[25,165, 60],
    "Mike":[27,170, 58],
    "Nancy":[29,160, 58],
    "Billy":[30,165, 69],
    "Max":[23,160, 49],
    "Jane":[25,168, 57],
    "Dustin":[28,155, 60],
    "Chrissy":[29,167, 55],
    "Jonathan":[30,169, 61],
    "Jonathan":[30,169, 61],
    "Lucas":[27,175, 65],

}


friends["Nancy"]
```

Out[35]:
```
[29, 160, 58]
```

In [36]:
```
#get number of items
print(len(friends))
```

```
10
```

In [37]:
```
#check type if it is in dictionary form
print(type(friends))
```

```
<class 'dict'>
```

In [38]:
```
#get the list of keys
print(friends.keys()) #or use x = friends.keys() and the print x
```

```
dict_keys(['Nancy', 'Steve', 'Mike', 'Billy', 'Max', 'Jane', 'Dustin', 'Chrissy',
'Jonathan', 'Lucas'])
```

In [39]:
```
#change current value

 #print
print('Original:')
print(friends["Lucas"])


#car["color"] = "red"
```

```
Original:
[27, 175, 65]
```

In [40]:
```
#change age

friends.update({"Lucas":[27,175,71]})
print('After update:')
print(friends["Lucas"])
```

```
After update:
[27, 175, 71]
```

In [41]:
```
#change using loops

for key, value in friends.items():
    if key == 'Billy':
        friends[key] = [30,165, 67]

print(friends["Billy"])
```

```
[30, 165, 67]
```

Pandas

In [51]:
```
# pandas -  these are tables in python and they are important for data analysis

#import the pandas library
import pandas as pd

#initialize an empty panda
x_df = pd.DataFrame()
print(x_df) #has no columns
```

```
Empty DataFrame
Columns: []
Index: []
```

In [ ]:
```

```

In [52]:
```
#create columns

x_df = pd.DataFrame(columns=['Name', 'Age', 'Birth City', 'Gender'])
print(x_df)
```

```
Empty DataFrame
Columns: [Name, Age, Birth City, Gender]
Index: []
```

In [60]:
```
#dataframe with columns and indecies

df = pd.DataFrame(
```

```
    columns=['Age', 'Birth City', 'Gender'],
    index=['Jane', 'Melissa', 'John', 'Matt'])
print(df)
```

```
         Age Birth City Gender
Jane     NaN        NaN    NaN
Melissa  NaN        NaN    NaN
John     NaN        NaN    NaN
Matt     NaN        NaN    NaN
```

In [54]:
```
#check if dataframe is still empty

print(df.empty)
```

```
False
```

In [ ]:

In [61]:
```
#use the LOC() fuction to add data to the dataframe

df.loc['Jane',:] = [23, 'London', 'F']
print(df)
```

```
         Age Birth City Gender
Jane      23     London      F
Melissa  NaN        NaN    NaN
John     NaN        NaN    NaN
Matt     NaN        NaN    NaN
```

In [ ]:

Pandas - Extracting Files

In [43]:
```
dataframe  = pd.read_csv("https://gist.githubusercontent.com/curran/a08a1080b88344
# read a csv file thats stored on github
dataframe.head()# show the top 5 rows in the data
```

Out[43]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

In [45]:
```
dataframe.tail()# show the botton 5 rows
```

Out[45]:

	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

In [46]:
```
dataframe.describe()# create a summary of the columsn in the data
```

Out[46]:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [49]:
```
 # extract one column with 10 rows

dataframe["sepal_length"].head(10)
```

Out[49]:
```
0    5.1
1    4.9
2    4.7
3    4.6
4    5.0
5    5.4
6    4.6
7    5.0
8    4.4
9    4.9
Name: sepal_length, dtype: float64
```

In [50]:
```
dataframe[["sepal_length","sepal_width"]].head()# extract two or more columns
```

Out[50]:

	sepal_length	sepal_width
0	5.1	3.5
1	4.9	3.0
2	4.7	3.2
3	4.6	3.1
4	5.0	3.6