- Creating a contact

In [13]:
```
contact = {}
def contactapp(name,number):
    if name not in contact:
        contact[name] = number
        print(name," contact is added")
    else:
        #contact[name] = number
        print(name," is already exists")
    return contact
contactapp("alekhya",9876543210)
contactapp("chandana",8796543217)
contactapp("chandana",89297642324)
contactapp("archana",9765421567)
```

```
alekhya  contact is added
chandana  contact is added
chandana  is already exists
archana  contact is added
```

Out[13]:
```
{'alekhya': 9876543210, 'chandana': 8796543217, 'archana': 9765421567}
```

- search for the name ,if name is exists then print name and number,otherwise print does not exists
- update the number in your contacts

In [5]:
```
def searchcontact(name):
    if name in contact:
        print(name,":",contact[name])
    else:
        print("contact does not exsists")
searchcontact("alekhya")
searchcontact("hemanth")
```

```
alekhya : 9876543210
contact does not exsists
```

In [2]:
```
s = {"alekhya":3456789}
s["alekhya"]
```

Out[2]:
```
3456789
```

In [9]:
```
C={"chandu" :47,"rani":48}
S=input("enter key to search")
if S in C:
      print ("yes it is exist", C.get(S))
else:
      print("contact doesn't exist")
```

```
enter key to searchalekhya
contact doesn't exist
```

- update your contact in contact application

In [16]:
```
def updatecontact(newcontact):
    contact.update(newcontact)
    print("contact is updated")
newcontact = {"hemanth":9076542134,"aihika":8976543210}
updatecontact(newcontact)
```

```
contact is updated
```

In [17]:
```
contact
```

Out[17]:
```
{'alekhya': 9876543210,
 'chandana': 8796543217,
 'archana': 9765421567,
 'hemanth': 9076542134,
 'aihika': 8976543210}
```

In [ ]:

Sets

- set is a collection of unique items without any order
- create a set
  - set()
  - symbol {} curly braces

In [18]:
```
set1 = set()
set2 = {1,2,3}

print(type(set1))
print(type(set2))
```

```
<class 'set'>
<class 'set'>
```

In [19]:
```
a = {1,2,3,41,2,3,4}
print(a)
```

```
{1, 2, 3, 4, 41}
```

- methods in sets

In [20]:
```
print(dir(set))
```

```
['__and__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge_
_', '__getattribute__', '__gt__', '__hash__', '__iand__', '__init__', '__init_subclass__', '__ior__', '__isub_
_', '__iter__', '__ixor__', '__le__', '__len__', '__lt__', '__ne__', '__new__', '__or__', '__rand__', '__reduc
e__', '__reduce_ex__', '__repr__', '__ror__', '__rsub__', '__rxor__', '__setattr__', '__sizeof__', '__str__',
'__sub__', '__subclasshook__', '__xor__', 'add', 'clear', 'copy', 'difference', 'difference_update', 'discar
d', 'intersection', 'intersection_update', 'isdisjoint', 'issubset', 'issuperset', 'pop', 'remove', 'symmetric
_difference', 'symmetric_difference_update', 'union', 'update']
```

- add(item)

In [21]:
```
names = {"alekhya","aihika","chandana"}
```

In [22]:
```
names
```

Out[22]:
```
{'aihika', 'alekhya', 'chandana'}
```

In [23]:
```
names.add("archana")
```

In [24]:
```
names
```

Out[24]:
```
{'aihika', 'alekhya', 'archana', 'chandana'}
```

In [30]:
```
names.add("archana")
```

In [31]:
```
names
```

Out[31]:
```
{'aihika', 'alekhya', 'archana', 'chandana'}
```

- remove()

In [32]:
```
names.remove("archana")
```

In [28]:
```
names
```

Out[28]:
```
{'aihika', 'alekhya', 'chandana'}
```

In [33]:
```
names.remove("archana")
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-33-1eae3f59d82a> in <module>
----> 1 names.remove("archana")

KeyError: 'archana'
```

- discard(item)

In [34]:
```
names
```

Out[34]:
```
{'aihika', 'alekhya', 'chandana'}
```

In [37]:
```
names.discard("chandana")
```

In [36]:
```
names
```

Out[36]:
```
{'aihika', 'alekhya'}
```

```
In [38]: names.discard("chandana")
```

```
In [39]: names
```

```
Out[39]: {'aihika', 'alekhya'}
```

```
In [41]: help(set.discard)
```

```
Help on method_descriptor:

discard(...)
    Remove an element from a set if it is a member.

    If the element is not a member, do nothing.
```

```
In [42]: help(set.remove)
```

```
Help on method_descriptor:

remove(...)
    Remove an element from a set; it must be a member.

    If the element is not a member, raise a KeyError.
```

- clear()

```
In [43]: names.clear()
```

```
In [44]: names
```

```
Out[44]: set()
```

- copy()

```
In [45]: names = {"a","b","c","d"}
         names
```

```
Out[45]: {'a', 'b', 'c', 'd'}
```

```
In [46]: namescopy = names.copy()
```

```
In [47]: namescopy
```

```
Out[47]: {'a', 'b', 'c', 'd'}
```

- intersection()

```
In [48]: s1 = {1,2,3,4,5}
         s2 = {4,2,1,6,7,78}
         s1.intersection(s2)
```

```
Out[48]: {1, 2, 4}
```

- union()

```
In [49]: s1
```

```
Out[49]: {1, 2, 3, 4, 5}
```

```
In [50]: s2
```

```
Out[50]: {1, 2, 4, 6, 7, 78}
```

```
In [51]: s1.union(s2)
```

```
Out[51]: {1, 2, 3, 4, 5, 6, 7, 78}
```

```
In [52]: s1
```

```
Out[52]: {1, 2, 3, 4, 5}
```

```
In [53]: s2
```

```
Out[53]: {1, 2, 4, 6, 7, 78}
```

- intersection_update()

```
In [54]: s1.intersection_update(s2)
```

```
In [55]: s1
```

```
Out[55]: {1, 2, 4}
```

```
In [56]: s2.intersection_update(s1)
```

```
In [57]: s2
```

```
Out[57]: {1, 2, 4}
```

```
In [58]: s1,s2
```

```
Out[58]: ({1, 2, 4}, {1, 2, 4})
```

- symmetric_difference()

```
In [59]: s1 = {1,2,3,4,5,6,7}
         s2 = {3,2,1,4,9,23,45}
```

```
In [60]: s1.symmetric_difference(s2)
```

```
Out[60]: {5, 6, 7, 9, 23, 45}
```

```
In [61]: s1
```

```
Out[61]: {1, 2, 3, 4, 5, 6, 7}
```

```
In [62]: s2
```

```
Out[62]: {1, 2, 3, 4, 9, 23, 45}
```

- symmetric_difference_update

```
In [63]: s2.symmetric_difference_update(s1)
```

```
In [64]: s2
```

```
Out[64]: {5, 6, 7, 9, 23, 45}
```

```
In [65]: s1.symmetric_difference_update(s2)
```

```
In [66]: s1
```

```
Out[66]: {1, 2, 3, 4, 9, 23, 45}
```

- find the common charcters between two persons

```
In [67]: person1 = "alekhya"
         person2 = "archana"
```

```
In [68]: person1.intersection(person2)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-68-6ef4c64b166b> in <module>
----> 1 person1.intersection(person2)

AttributeError: 'str' object has no attribute 'intersection'
```

```
In [69]: s1 = set(person1)
```

```
In [70]: s1
```

```
Out[70]: {'a', 'e', 'h', 'k', 'l', 'y'}
```

```
In [71]: print(type(s1))
```

```
<class 'set'>
```

```
In [72]: print(type(person1))

         <class 'str'>

In [73]: s2 = set(person2)

In [74]: s2

Out[74]: {'a', 'c', 'h', 'n', 'r'}

In [75]: type(s2)

Out[75]: set

In [76]: s1.intersection(s2)

Out[76]: {'a', 'h'}

In [77]: s1.union(s2)

Out[77]: {'a', 'c', 'e', 'h', 'k', 'l', 'n', 'r', 'y'}

In [78]: s1.difference(s2)

Out[78]: {'e', 'k', 'l', 'y'}

In [79]: s2.difference(s1)

Out[79]: {'c', 'n', 'r'}
```

Packages and Modules

```
In [80]: help("modules")
```

```

Please wait a moment while I gather a list of all available modules...


C:\Users\Alekhya\Anaconda3\lib\site-packages\IPython\kernel\__init__.py:13: ShimWarning: The `IPython.kernel
` package has been deprecated since IPython 4.0.You should import from ipykernel or jupyter_client instead.
  "You should import from ipykernel or jupyter_client instead.", ShimWarning)
WARNING: AstropyDeprecationWarning: astropy.utils.compat.futures is now deprecated - use concurrent.futures
instead [astropy.utils.compat.futures]
WARNING: The conda.compat module is deprecated and will be removed in a future release.
C:\Users\Alekhya\Anaconda3\lib\site-packages\dask\config.py:168: YAMLLoadWarning: calling yaml.load() withou
t Loader=... is deprecated, as the default Loader is unsafe. Please read https://msg.pyyaml.org/load (http
s://msg.pyyaml.org/load) for full details.
  data = yaml.load(f.read()) or {}
C:\Users\Alekhya\Anaconda3\lib\site-packages\distributed\config.py:20: YAMLLoadWarning: calling yaml.load()
 without Loader=... is deprecated, as the default Loader is unsafe. Please read https://msg.pyyaml.org/load
 (https://msg.pyyaml.org/load) for full details.
  defaults = yaml.load(f)
C:\Users\Alekhya\Anaconda3\lib\site-packages\nltk\twitter\__init__.py:22: UserWarning: The twython library h
as not been installed. Some functionality from the twitter package will not be available.
```

```
In [82]: help("math")
```

```
Help on built-in module math:

NAME
    math

DESCRIPTION
    This module is always available.  It provides access to the
    mathematical functions defined by the C standard.

FUNCTIONS
    acos(x, /)
        Return the arc cosine (measured in radians) of x.

    acosh(x, /)
        Return the inverse hyperbolic cosine of x.

    asin(x, /)
        Return the arc sine (measured in radians) of x.
```

```
In [86]: import math
```

In [87]:

```
print(dir(math))
```

```
['__doc__', '__loader__', '__name__', '__package__', '__spec__', 'acos', 'acosh', 'asin', 'asinh', 'atan', 'at
an2', 'atanh', 'ceil', 'copysign', 'cos', 'cosh', 'degrees', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs', 'fact
orial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf', 'isclose', 'isfinite', 'isinf', 'isn
an', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'log2', 'modf', 'nan', 'pi', 'pow', 'radians', 'remainder',
'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'tau', 'trunc']
```

In [89]:

```
math.sqrt(4)
```

Out[89]:

```
2.0
```

In [90]:

```
sqrt(4)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-90-317e033d29d5> in <module>
----> 1 sqrt(4)

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

In [91]:

```
import webbrowser
import time
```

In [92]:

```
print(dir(webbrowser))
```

```
['BackgroundBrowser', 'BaseBrowser', 'Chrome', 'Chromium', 'Elinks', 'Error', 'Galeon', 'GenericBrowser', 'Gra
il', 'Konqueror', 'Mozilla', 'Netscape', 'Opera', 'UnixBrowser', 'WindowsDefault', '__all__', '__builtins__',
'__cached__', '__doc__', '__file__', '__loader__', '__name__', '__package__', '__spec__', '_browsers', '_loc
k', '_os_preferred_browser', '_synthesize', '_tryorder', 'get', 'main', 'open', 'open_new', 'open_new_tab', 'o
s', 'register', 'register_X_browsers', 'register_standard_browsers', 'shlex', 'shutil', 'subprocess', 'sys',
'threading']
```

In [93]:

```
print(dir(time))
```

```
['_STRUCT_TM_ITEMS', '__doc__', '__loader__', '__name__', '__package__', '__spec__', 'altzone', 'asctime', 'cl
ock', 'ctime', 'daylight', 'get_clock_info', 'gmtime', 'localtime', 'mktime', 'monotonic', 'monotonic_ns', 'pe
rf_counter', 'perf_counter_ns', 'process_time', 'process_time_ns', 'sleep', 'strftime', 'strptime', 'struct_ti
me', 'thread_time', 'thread_time_ns', 'time', 'time_ns', 'timezone', 'tzname']
```

In [96]:

```
web = ["www.gmail.com","www.youtube.com","www.amazon.in"]
for url in web:
    webbrowser.open(url)
    time.sleep(15)
```

In [97]:

```
help(time.sleep)
```

```
Help on built-in function sleep in module time:

sleep(...)
    sleep(seconds)

    Delay execution for a given number of seconds.  The argument may be
    a floating point number for subsecond precision.
```

- creating own modules

In [1]:

```
import mymodule
```

In [2]:

```
mymodule.evenodd(8)
```

```
even
```

In [3]:

```
mymodule.evenodd(45)
```

```
odd
```

In [4]:

```
mymodule.add(2,3)
```

```
5
```

In [5]:

```
mymodule.add(23,45)
```

```
68
```

In [6]:

```
mymodule.hello("alekhya")
```

```
Hello alekhya
```

In [7]:
```
pip list
```

```
Package                            Version
---------------------------------- --------
alabaster                          0.7.12
anaconda-client                    1.7.2
anaconda-navigator                 1.9.7
anaconda-project                   0.8.2
asn1crypto                         0.24.0
astroid                            2.2.5
astropy                            3.1.2
atomicwrites                       1.3.0
attrs                              19.1.0
Babel                              2.6.0
backcall                           0.1.0
backports.os                       0.1.1
backports.shutil-get-terminal-size 1.0.0
beautifulsoup4                     4.7.1
bitarray                           0.8.3
bkcharts                           0.2
bleach                             3.1.0
```

In [8]:
```
import seaborn
```

In [9]:
```
print(dir(seaborn))
```

```
['FacetGrid', 'JointGrid', 'PairGrid', '__builtins__', '__cached__', '__doc__', '__file__', '__loader__', '__n
ame__', '__package__', '__path__', '__spec__', '__version__', '_orig_rc_params', 'algorithms', 'axes_style',
'axisgrid', 'barplot', 'blend_palette', 'boxenplot', 'boxplot', 'categorical', 'catplot', 'choose_colorbrewer_
palette', 'choose_cubehelix_palette', 'choose_dark_palette', 'choose_diverging_palette', 'choose_light_palett
e', 'clustermap', 'cm', 'color_palette', 'colors', 'countplot', 'crayon_palette', 'crayons', 'cubehelix_palett
e', 'dark_palette', 'desaturate', 'despine', 'distplot', 'distributions', 'diverging_palette', 'dogplot', 'ext
ernal', 'factorplot', 'get_dataset_names', 'heatmap', 'hls_palette', 'husl_palette', 'jointplot', 'kdeplot',
'light_palette', 'lineplot', 'lmplot', 'load_dataset', 'lvplot', 'matrix', 'miscplot', 'mpl', 'mpl_palette',
'pairplot', 'palettes', 'palplot', 'plotting_context', 'pointplot', 'rcmod', 'regplot', 'regression', 'relatio
nal', 'relplot', 'reset_defaults', 'reset_orig', 'residplot', 'rugplot', 'saturate', 'scatterplot', 'set', 'se
t_color_codes', 'set_context', 'set_hls_values', 'set_palette', 'set_style', 'stripplot', 'swarmplot', 'timese
ries', 'tsplot', 'utils', 'violinplot', 'widgets', 'xkcd_palette', 'xkcd_rgb']
```

In [10]:
```
import pandas
```

In [11]:

```
help(pandas)
```

```
Help on package pandas:

NAME
    pandas

DESCRIPTION
    pandas - a powerful data analysis and manipulation library for Python
    =====================================================================

    **pandas** is a Python package providing fast, flexible, and expressive data
    structures designed to make working with "relational" or "labeled" data both
    easy and intuitive. It aims to be the fundamental high-level building block for
    doing practical, **real world** data analysis in Python. Additionally, it has
    the broader goal of becoming **the most powerful and flexible open source data
    analysis / manipulation tool available in any language**. It is already well on
    its way toward this goal.

    Main Features
    -------------
    Here are just a few of the things that pandas does well:

      - Easy handling of missing data in floating point as well as non-floating
        point data.
      - Size mutability: columns can be inserted and deleted from DataFrame and
        higher dimensional objects
      - Automatic and explicit data alignment: objects can be explicitly aligned
        to a set of labels, or the user can simply ignore the labels and let
        `Series`, `DataFrame`, etc. automatically align the data for you in
        computations.
      - Powerful, flexible group by functionality to perform split-apply-combine
        operations on data sets, for both aggregating and transforming data.
      - Make it easy to convert ragged, differently-indexed data in other Python
        and NumPy data structures into DataFrame objects.
      - Intelligent label-based slicing, fancy indexing, and subsetting of large
        data sets.
      - Intuitive merging and joining data sets.
      - Flexible reshaping and pivoting of data sets.
      - Hierarchical labeling of axes (possible to have multiple labels per tick).
      - Robust IO tools for loading data from flat files (CSV and delimited),
        Excel files, databases, and saving/loading data from the ultrafast HDF5
        format.
      - Time series-specific functionality: date range generation and frequency
        conversion, moving window statistics, moving window linear regressions,
        date shifting and lagging, etc.

PACKAGE CONTENTS
    _libs (package)
    _version
    api (package)
    arrays (package)
    compat (package)
    conftest
    core (package)
    errors (package)
    io (package)
    plotting (package)
    testing
    tests (package)
    tseries (package)
    util (package)

SUBMODULES
    _hashtable
    _lib
    _tslib
    offsets

DATA
    IndexSlice = <pandas.core.indexing._IndexSlice object>
    NaT = NaT
    __docformat__ = 'restructuredtext'
    __git_version__ = 'cb00deb94500205fcb27a33cc1d0df79a9727f8b'
    describe_option = <pandas.core.config.CallableDynamicDoc object>
    get_option = <pandas.core.config.CallableDynamicDoc object>
    options = <pandas.core.config.DictWrapper object>
    reset_option = <pandas.core.config.CallableDynamicDoc object>
    set_option = <pandas.core.config.CallableDynamicDoc object>

VERSION
    0.24.2

FILE
    c:\users\alekhya\anaconda3\lib\site-packages\pandas\__init__.py
```

```
In [12]: help(seaborn)
```

```
Help on package seaborn:

NAME
    seaborn - # Capture the original matplotlib rcParams

PACKAGE CONTENTS
    algorithms
    apionly
    axisgrid
    categorical
    cm
    colors (package)
    conftest
    distributions
    external (package)
    linearmodels
    matrix
    miscplot
    palettes
    rcmod
    regression
    relational
    tests (package)
    timeseries
    utils
    widgets

DATA
    crayons = {'Almond': '#EFDECD', 'Antique Brass': '#CD9575', 'Apricot':...
    xkcd_rgb = {'acid green': '#8ffe09', 'adobe': '#bd6c48', 'algae': '#54...

VERSION
    0.9.0

FILE
    c:\users\alekhya\anaconda3\lib\site-packages\seaborn\__init__.py
```

```
In [1]: from mypackage import module
```

```
In [2]: module.isprime(5)
```

```
prime number
```

```
In [3]: module.isprime(12)
```

```
not a prime number
```

```
In [4]: from mypackage import mod
```

```
In [5]: mod.names("alekhya")
```

```
alekhya
```

```
In [6]: module.factorial(5)
```

```
120
```

```
In [ ]:
```