In [52]:
```
#  1   split this stri
```

In [2]:
```
#1 split this string

s="hi there sam!"
h=s.split()
print(h);
```

```
['hi', 'there', 'sam!']
```

In [ ]:
```
# 2  use. format() to print the following string
```

In [40]:
```
      #the  diameter of earth is 1274 2 kilometers
```

In [41]:
```
planet="earth"
diameter=12742
print('the diameter of {} is {} killometers.'. format(planet, diameter));
```

```
the diameter of earth is 12742 killometers.
```

In [11]:
```
#3 in this nested  dictionary grab the word "hello".


d={'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}
d={'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}
print(d['k1'][3]["tricky"][3]['target'][3])
```

```
hello
```

In [84]:
```
#4a)array of 10 zeros
import numpy as np
array=np.zeros(10)*2
print("an array of 10 zero")
print(array)
```

```
an array of 10 zero
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

In [87]:
```
#4b) array(10)*5
import numpy as np
array=np.ones(10)*5
print("an array of 10 five")
print(array)
```

```
an array of 10 five
[5. 5. 5. 5. 5. 5. 5. 5. 5. 5.]
```

In [59]:
```
#5)arrray of all the even integer from 20 to 35")
import numpy as np
array=np.arange(20,35)
print("array of all the even integers from 20 to 35")
print(array)
```

```
array of all the even integers from 20 to 35
[20 21 22 23 24 25 26 27 28 29 30 31 32 33 34]
```

In [70]:
```
#6)3x3 matrix
import numpy as np
array_2D=np.identity(3)
print('3*3 matrix:')
print(array_2D)
```

Loading [MathJax]/extensions/Safe.js

```
3*3 matrix:
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```

In [38]:
```
#7)concatenation
import numpy as np
a=np.array([1,2,3])
b=np.array ([4,5,6])
c=a+b,a*b,a-b
print(c)
```

```
(array([5, 7, 9]), array([ 4, 10, 18]), array([-3, -3, -3]))
```

In [28]:
```
#8  Creating  Dataframe from Lists

# Import pandas library
import pandas as pd

# initialize list elements
data = [10,20,30,40,50,60]

# Create the pandas DataFrame with column name is provided explicitly
df = pd.DataFrame(data, columns=['Numbers'])

# print dataframe.
df
```

Out[28]:

	Numbers
0	10
1	20
2	30
3	40
4	50
5	60

In [74]:
```
#9 generate the series of dates from 1st 2023,1,1. to 2023-10-2
import timedelta,data
def daterange(date1,date2);
for n if in_rang(int((date2-date).days)+1):
    yield date1+timedelta(n)
   start_dt=date(2023-1-1)
    end_dt=date=date(2023-10-2)
    for dt in daterange(start_dt,end_dt);
    print9dt.strftime("yy-%m-%d")
```

```
  File <tokenize>:6
    start_dt=date(2023-1-1)
    ^
IndentationError: unindent does not match any outer indentation level
```

In [ ]:

In [19]:
```
#10)create 2d list to data frame
import pandas as pd

# List1
lst=[[1,'aaa',22],[2,'bbb',25],[3,'ccc',24]]
```

Loading [MathJax]/extensions/Safe.js

```
# creating df object with columns specified
df=pd.DataFrame(lst, columns =['FName', 'LName', 'Age'], dtype =float )
print(df)
```

```
   FName LName   Age
0    1.0   aaa  22.0
1    2.0   bbb  25.0
2    3.0   ccc  24.0
```

```
C:\Users\GANESH\AppData\Local\Temp\ipykernel_6748\3116688088.py:8: FutureWarning: Could
not cast to float64, falling back to object. This behavior is deprecated. In a future ve
rsion, when a dtype is passed to 'DataFrame', either all columns will be cast to that dt
ype, or a TypeError will be raised.
  df=pd.DataFrame(lst, columns =['FName', 'LName', 'Age'], dtype =float )
```

In [ ]:

Loading [MathJax]/extensions/Safe.js