

Python - Matrix

Matrix is a special case of two dimensional array where each data element is of strictly same size. So every matrix is also a two dimensional array but not vice versa.

Matrices are very important data structures for many mathematical and scientific calculations. As we have already discussed two dimensional array data structure in the previous chapter we will be focusing on data structure operations specific to matrices in this chapter.

We also be using the numpy package for matrix data manipulation.

Matrix Example

Consider the case of recording temprature for 1 week measured in the morning, mid-day, evening and mid-night. It can be presented as a 7X5 matrix using an array and the reshape method available in numpy.

```
from numpy import *
a = array([[ 'Mon',18,20,22,17],[ 'Tue',11,18,21,18],
           [ 'Wed',15,21,20,19],[ 'Thu',11,20,22,21],
           [ 'Fri',18,17,23,22],[ 'Sat',12,22,20,18],
           [ 'Sun',13,15,19,16]])
m = reshape(a,(7,5))
print(m)
```

Output

The above data can be represented as a two dimensional array as below –

```
[  
    ['Mon' '18' '20' '22' '17'],  
    ['Tue' '11' '18' '21' '18'],  
    ['Wed' '15' '21' '20' '19'],  
    ['Thu' '11' '20' '22' '21'],  
    ['Fri' '18' '17' '23' '22'],  
    ['Sat' '12' '22' '20' '18'],  
    ['Sun' '13' '15' '19' '16']  
]
```

Accessing Values

The data elements in a matrix can be accessed by using the indexes. The access method is same as the way data is accessed in Two dimensional array.

Example

```
from numpy import *  
  
m = array([[ 'Mon', 18, 20, 22, 17], [ 'Tue', 11, 18, 21, 18],  
           [ 'Wed', 15, 21, 20, 19], [ 'Thu', 11, 20, 22, 21],  
           [ 'Fri', 18, 17, 23, 22], [ 'Sat', 12, 22, 20, 18],  
           [ 'Sun', 13, 15, 19, 16]])  
  
# Print data for Wednesday  
print(m[2])
```

```
# Print data for friday evening
print(m[4][3])
```

Output

When the above code is executed, it produces the following result –

```
[ 'Wed', 15, 21, 20, 19]
23
```

Adding a row

Use the below mentioned code to add a row in a matrix.

Example

```
from numpy import *
m = array([[ 'Mon', 18, 20, 22, 17], [ 'Tue', 11, 18, 21, 18],
           [ 'Wed', 15, 21, 20, 19], [ 'Thu', 11, 20, 22, 21],
           [ 'Fri', 18, 17, 23, 22], [ 'Sat', 12, 22, 20, 18],
           [ 'Sun', 13, 15, 19, 16]])
m_r = append(m, [[ 'Avg', 12, 15, 13, 11]], 0)

print(m_r)
```

Output

When the above code is executed, it produces the following result –

```
[
 [ 'Mon' '18' '20' '22' '17']
```

```
[ 'Tue' '11' '18' '21' '18']
[ 'Wed' '15' '21' '20' '19']
[ 'Thu' '11' '20' '22' '21']
[ 'Fri' '18' '17' '23' '22']
[ 'Sat' '12' '22' '20' '18']
[ 'Sun' '13' '15' '19' '16']
[ 'Avg' '12' '15' '13' '11']
]
```

Adding a column

We can add column to a matrix using the `insert()` method. here we have to mention the index where we want to add the column and a array containing the new values of the columns added.In the below example we add t a new column at the fifth position from the beginning.

Example

```
from numpy import *
m = array([[ 'Mon',18,20,22,17],[ 'Tue',11,18,21,18],
[ 'Wed',15,21,20,19],[ 'Thu',11,20,22,21],
[ 'Fri',18,17,23,22],[ 'Sat',12,22,20,18],
[ 'Sun',13,15,19,16]])
m_c = insert(m,[5],[[1],[2],[3],[4],[5],[6],[7]],1)
print(m_c)
```

Output

When the above code is executed, it produces the following result –

```
[ 
[ 'Mon' '18' '20' '22' '17' '1']
```

```
[ 'Tue' '11' '18' '21' '18' '2' ]
[ 'Wed' '15' '21' '20' '19' '3' ]
[ 'Thu' '11' '20' '22' '21' '4' ]
[ 'Fri' '18' '17' '23' '22' '5' ]
[ 'Sat' '12' '22' '20' '18' '6' ]
[ 'Sun' '13' '15' '19' '16' '7' ]
]
```

Delete a row

We can delete a row from a matrix using the `delete()` method. We have to specify the index of the row and also the axis value which is 0 for a row and 1 for a column.

Example

```
from numpy import *
m = array([[ 'Mon', 18, 20, 22, 17], [ 'Tue', 11, 18, 21, 18],
           [ 'Wed', 15, 21, 20, 19], [ 'Thu', 11, 20, 22, 21],
           [ 'Fri', 18, 17, 23, 22], [ 'Sat', 12, 22, 20, 18],
           [ 'Sun', 13, 15, 19, 16]])
m = delete(m,[2],0)
print(m)
```

Output

When the above code is executed, it produces the following result –

```
[ 
  [ 'Mon' '18' '20' '22' '17' ]
  [ 'Tue' '11' '18' '21' '18' ]
  [ 'Thu' '11' '20' '22' '21' ]
```

```
[ 'Fri' '18' '17' '23' '22']
[ 'Sat' '12' '22' '20' '18']
[ 'Sun' '13' '15' '19' '16']
]
```

Delete a column

We can delete a column from a matrix using the `delete()` method. We have to specify the index of the column and also the axis value which is 0 for a row and 1 for a column.

Example

```
from numpy import *
m = array([[ 'Mon',18,20,22,17],[ 'Tue',11,18,21,18],
           [ 'Wed',15,21,20,19],[ 'Thu',11,20,22,21],
           [ 'Fri',18,17,23,22],[ 'Sat',12,22,20,18],
           [ 'Sun',13,15,19,16]])
m = delete(m,s_[2],1)

print(m)
```

Output

When the above code is executed, it produces the following result –

```
[ 
  [ 'Mon' '18' '22' '17']
  [ 'Tue' '11' '21' '18']
  [ 'Wed' '15' '20' '19']
  [ 'Thu' '11' '22' '21']
  [ 'Fri' '18' '23' '22']
  [ 'Sat' '12' '20' '18']
```

```
[ 'Sun' '13' '19' '16']  
]
```

Update a row

To update the values in the row of a matrix we simply re-assign the values at the index of the row. In the below example all the values for thrusday's data is marked as zero. The index for this row is 3.

Example

```
from numpy import *  
m = array([[ 'Mon', 18, 20, 22, 17], [ 'Tue', 11, 18, 21, 18],  
          [ 'Wed', 15, 21, 20, 19], [ 'Thu', 11, 20, 22, 21],  
          [ 'Fri', 18, 17, 23, 22], [ 'Sat', 12, 22, 20, 18],  
          [ 'Sun', 13, 15, 19, 16]])  
m[3] = [ 'Thu', 0, 0, 0, 0]  
  
print(m)
```

Output

When the above code is executed, it produces the following result –

```
[  
 [ 'Mon' '18' '20' '22' '17']  
 [ 'Tue' '11' '18' '21' '18']  
 [ 'Wed' '15' '21' '20' '19']  
 [ 'Thu' '0' '0' '0' '0']  
 [ 'Fri' '18' '17' '23' '22']  
 [ 'Sat' '12' '22' '20' '18']
```

```
[ 'Sun' '13' '15' '19' '16']  
]
```

 **Print Page**