

## MATLAB – Matrix

A matrix is a two-dimensional array of numbers.

In MATLAB, you create a matrix by entering elements in each row as comma or space delimited numbers and using semicolons to mark the end of each row.

For example, let us create a 4-by-5 matrix **a**

```
a = [ 1 2 3 4 5; 2 3 4 5 6; 3 4 5 6 7; 4 5 6 7 8]
```

MATLAB will execute the above statement and return the following result –

```
a =  
    1    2    3    4    5  
    2    3    4    5    6  
    3    4    5    6    7  
    4    5    6    7    8
```

## Referencing the Elements of a Matrix

To reference an element in the  $m^{\text{th}}$  row and  $n^{\text{th}}$  column, of a matrix **mx**, we write –

```
mx(m, n);
```

For example, to refer to the element in the 2nd row and 5th column, of the matrix **a**, as created in the last section, we type –

```
a = [ 1 2 3 4 5; 2 3 4 5 6; 3 4 5 6 7; 4 5 6 7 8];  
a(2,5)
```

MATLAB will execute the above statement and return the following result –

```
ans = 6
```

To reference all the elements in the  $m^{\text{th}}$  column we type  $A(:,m)$ .

Let us create a column vector **v**, from the elements of the 4<sup>th</sup> row of the matrix **a**:

```
a = [ 1 2 3 4 5; 2 3 4 5 6; 3 4 5 6 7; 4 5 6 7 8];  
v = a(:,4)
```

MATLAB will execute the above statement and return the following result –

```
v =
```

```
4  
5  
6  
7
```

You can also select the elements in the  $m^{\text{th}}$  through  $n^{\text{th}}$  columns, for this we write –

```
a(:,m:n)
```

Let us create a smaller matrix taking the elements from the second and third columns –

```
a = [ 1 2 3 4 5; 2 3 4 5 6; 3 4 5 6 7; 4 5 6 7 8];
```

```
a(:, 2:3)
```

MATLAB will execute the above statement and return the following result –

```
ans =
```

```
2    3  
3    4  
4    5  
5    6
```

In the same way, you can create a sub-matrix taking a sub-part of a matrix.

```
a = [ 1 2 3 4 5; 2 3 4 5 6; 3 4 5 6 7; 4 5 6 7 8];
```

```
a(:, 2:3)
```

MATLAB will execute the above statement and return the following result –

```
ans =
```

```
2    3  
3    4  
4    5  
5    6
```

In the same way, you can create a sub-matrix taking a sub-part of a matrix.

For example, let us create a sub-matrix say taking the inner subpart of **a**:

```
3    4    5  
4    5    6
```

To do this, write –

```
a = [ 1 2 3 4 5; 2 3 4 5 6; 3 4 5 6 7; 4 5 6 7 8];  
sa = a(2:3,2:4)
```

MATLAB will execute the above statement and return the following result –

```
sa =  
    3    4    5  
    4    5    6
```

## Deleting a Row or a Column in a Matrix

You can delete an entire row or column of a matrix by assigning an empty set of square braces [] to that row or column. Basically, [] denotes an empty array.

For example, let us delete the fourth row of **a** –

```
a = [ 1 2 3 4 5; 2 3 4 5 6; 3 4 5 6 7; 4 5 6 7 8];  
a( 4 , : ) = []
```

MATLAB will execute the above statement and return the following result –

```
a =  
    1    2    3    4    5  
    2    3    4    5    6  
    3    4    5    6    7
```

Next, let us delete the fifth column of **a** –

```
a = [ 1 2 3 4 5; 2 3 4 5 6; 3 4 5 6 7; 4 5 6 7 8];  
a(:, 5)=[]
```

MATLAB will execute the above statement and return the following result –

```
a =  
    1    2    3    4  
    2    3    4    5  
    3    4    5    6  
    4    5    6    7
```

## Example

In this example, let us create a 3-by-3 matrix *m*, then we will copy the second and third rows of this matrix twice to create a 4-by-3 matrix.

Create a script file with the following code –

```
a = [ 1 2 3 ; 4 5 6; 7 8 9];  
new_mat = a([2,3,2,3],:)
```

When you run the file, it displays the following result –

```
new_mat =  
    4    5    6  
    7    8    9  
    4    5    6  
    7    8    9
```