Make and Manipulate Matrices

Conclusion

Summary

Summary: Make and Manipulate Matrices

Manually Enter Arrays

Create a Row Vector

Use square brackets and separate the values using a comma or a space.

```
a = [10 15 20 25]

a =
     10   15   20   25
```

```
a = [10, 15, 20, 25]

a =
     10   15   20   25
```

Create a Column Vector

Use square brackets and separate the values using a semicolon.

```
b = [2; 3; 5; 7]

b =
     2
     3
     5
     7
```

Transpose a Vector

Use the transpose operator.

```
c = b'

c =
     2   3   5   7
```

Create a Matrix

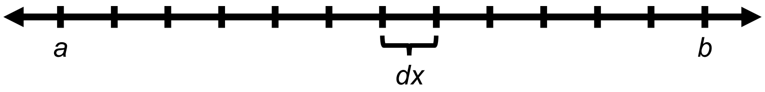
Use square brackets and enter values row-by-row. Separate values in a row using a comma or a space, and use a semicolon to start a new row.

```
A = [1 3 5; 2 4 6]

A =
     1   3   5
     2   4   6
```

Create Evenly Spaced Vectors

Given the Starting Value, Ending Value, and Interval



Use the colon operator to separate the starting value, the interval, and the ending value.

```
a = 3:2:7

a =
     3   5   7
```

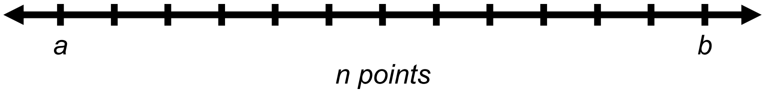
When the Interval Is 1

Use the colon operator to separate the starting and the ending value.

```
b = 3:7

b =
     3   4   5   6   7
```

Given the Starting Value, Ending Value, and Number of Elements



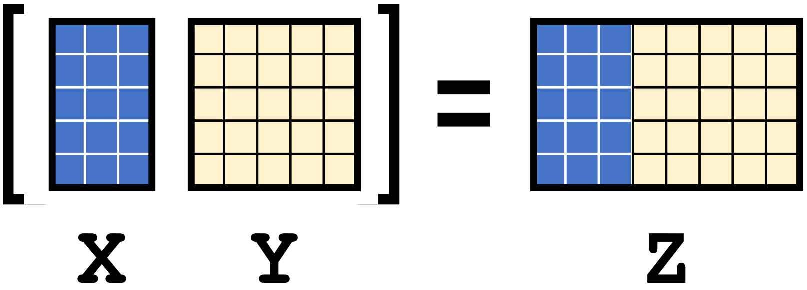
Use the function `linspace` when you know the number of elements in the vector.

```
c = linspace(3.2,8.1,5)

c =
     3.2   4.42   5.65   6.87   8.1
```

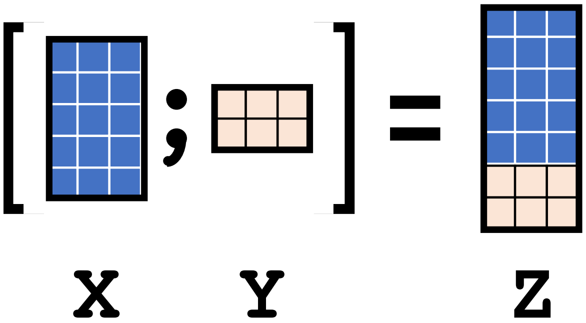
Concatenate Arrays

Horizontal Concatenation



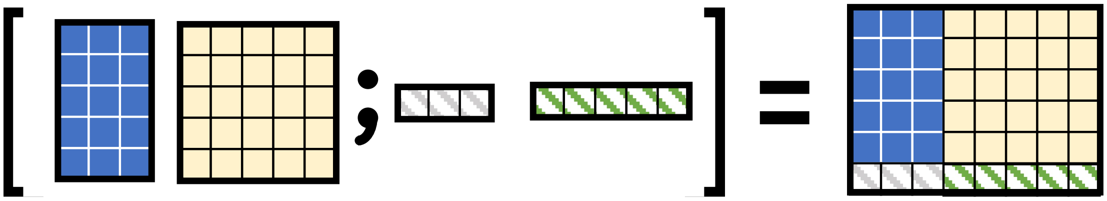
Separate elements by using a comma (,) or space ( ).

Vertical Concatenation



Separate elements by using a semicolon (;).

Combined Concatenation



Create each row by separating elements with a comma (,) or space ( ), and then separate the rows with a semicolon (;).

Array Creation Functions

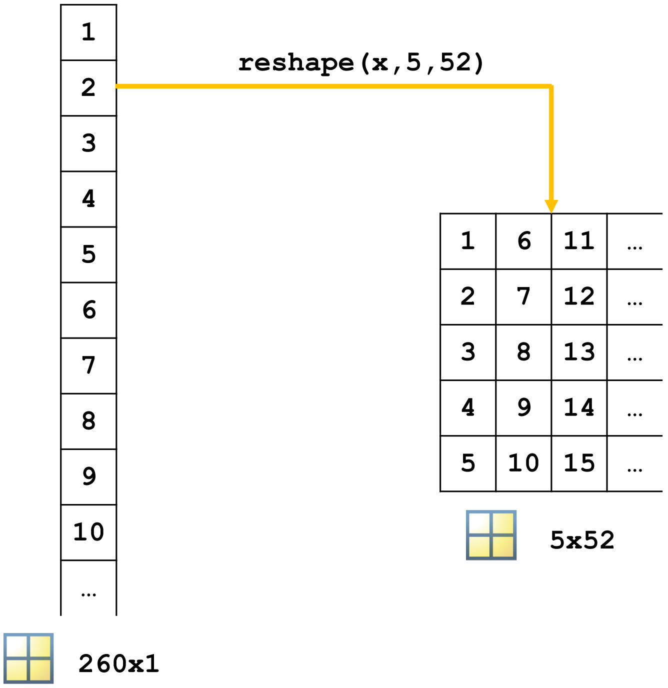
You can use several functions to create arrays.

Most of these functions support the calling syntaxes shown below.

Calling Syntax	Output
fun(*m*,*n*)	m-by-n
fun(*n*)	n-by-n

Reshape Arrays

You can reshape a column into a matrix.



1	6	11	...
2	7	12	...
3	8	13	...
4	9	14	...
5	10	15	...

Create a vector of random numbers to reshape.

```
x = rand(260,1);
```

Specify the dimensions for the new array.

```
y = reshape(x,5,52);
```

For convenience, you can also leave one of the dimensions blank when you call `reshape` and the function calculates that dimension automatically.

```
y = reshape(x,5,[]);
```

You can create a vector from a matrix by using the colon operator.

```
y = x(:);
```

Indexing

$$\begin{array}{c|c|}
\cline{2-2}
1 & 2.3 \\
\cline{2-2}
2 & 1.5 \\
\cline{2-2}
3 & 1.3 \\
\cline{2-2}
4 & 0.9 \\
\cline{2-2}
5 & 1.3 \\
\cline{2-2}
\end{array}$$

v

Extract one element from a vector.	`v(2)`
	`    1.5`

Extract the last element from a vector.	`v(end)`
	`    1.3`

Extract multiple elements from a vector.	`v([1 end-2:end])`
	`    2.3`
	`    1.3`
	`    0.9`
	`    1.3`

To extract elements from a matrix, provide two indices, the row and column numbers.

$$\begin{array}{c|c|c|c|c|}
 & 1 & 2 & 3 & 4 \\
\hline
1 & 1.5 & 1.1 & 2.6 & 0.9 \\
\hline
2 & 1.5 & 2.4 & 1.7 & 1.4 \\
\hline
3 & 2.5 & 1.6 & 1.9 & 0.7 \\
\hline
4 & 2.4 & 1.1 & 1.8 & 2.5 \\
\hline
5 & 1.9 & 2.8 & 0.6 & 0.6 \\
\hline
\end{array}$$

M

Extract one element from a matrix.

```
M(2,3)

    1.7
```

Extract an entire column. Here, it is the last one.

```
M(:,end)

    0.9
    1.4
    0.7
    2.5
    0.6
```

Extract multiple elements from a matrix.

```
M([1 end],2)

    1.1
    2.8
```

Change Elements in Arrays

Change one element in a vector.

```
v(2) = 0

    2.3
    0
    1.3
    0.9
    1.3
```

Change multiple elements in a vector to the same value.

```
v(1:3) = 0

    0
    0
    0
    0.9
    1.3
```

Change multiple elements in a vector to different values.

```
v(1:3) = [3 5 7]

    3
    5
    7
    0.9
    1.3
```

Remove elements from a vector.	`v(1:3) = []`
	`0.9`
	`1.3`

Assign a nonexistent value. The vector is padded with zeros.	`v(5) = 42`
	`0.9`
	`1.3`
	`0`
	`0`
	`42`

Changing elements in matrices works the same way as with vectors, but you must specify both rows and columns.