

In[]:= **u = {u1, u2, u3}; v = {v1, v2, v3};**
u.v

Out[]:= **u1 v1 + u2 v2 + u3 v3**

In[]:= **u * v**

Out[]:= **{u1 v1, u2 v2, u3 v3}**

In[]:= **? Dot**

Out[]:=

Symbol i

a.b.c or **Dot[a, b, c]** gives products of vectors, matrices, and tensors.

▼

In[]:= **Dot[u, v]**

Out[]:= **u1 v1 + u2 v2 + u3 v3**

In[]:= **? Cross**

Out[]:=

Symbol i

Cross[a, b] gives the vector cross product of *a* and *b*.

▼

In[]:= **Cross[u, v]**

Out[]:= **{-u3 v2 + u2 v3, u3 v1 - u1 v3, -u2 v1 + u1 v2}**

In[]:= **Cross[v, u]**

Out[]:= **{u3 v2 - u2 v3, -u3 v1 + u1 v3, u2 v1 - u1 v2}**

In[]:= **? Range**

Out[]:=

Symbol i

Range[i_{max}] generates the list {1, 2, ..., i_{max}}.

Range[i_{min}, i_{max}] generates the list {i_{min}, ..., i_{max}}.

Range[i_{min}, i_{max}, di] uses step *di*.

▼

In[]:= **a = Range[5]**

Out[]:= **{1, 2, 3, 4, 5}**

In[]:= **Range[7, 15]**

Out[]:= **{7, 8, 9, 10, 11, 12, 13, 14, 15}**

```
In[ ]:= Range[3, 9, 3]
```

```
Out[ ]:= {3, 6, 9}
```

```
In[ ]:= ? Table
```

```
Out[ ]:=
```

Symbol i

`Table[expr, n]` generates a list of n copies of `expr`.

`Table[expr, {i, imax}]` generates a list of the values of `expr` when i runs from 1 to i_{max} .

`Table[expr, {i, imin, imax}]` starts with $i = i_{min}$.

`Table[expr, {i, imin, imax, di}]` uses steps di .

`Table[expr, {i, {i1, i2, ...}}]` uses the successive values i_1, i_2, \dots .

`Table[expr, {i, imin, imax}, {j, jmin, jmax}, ...]` gives a nested list. The list associated with i is outermost.

▼

```
In[ ]:= Table[2 a, {3}]
```

```
Out[ ]:= {{2, 4, 6, 8, 10}, {2, 4, 6, 8, 10}, {2, 4, 6, 8, 10}}
```

```
In[ ]:= Table[x^2 + 1, {x, 3}]
```

```
Out[ ]:= {2, 5, 10}
```

```
In[ ]:= Table[x^2 + 1, {x, 4, 7}]
```

```
Out[ ]:= {17, 26, 37, 50}
```

```
In[ ]:= Table[x^2 + 1, {x, 4, 8, .2}]
```

```
Out[ ]:= {17., 18.64, 20.36, 22.16, 24.04, 26., 28.04, 30.16, 32.36, 34.64,
37., 39.44, 41.96, 44.56, 47.24, 50., 52.84, 55.76, 58.76, 61.84, 65.}
```

```
In[ ]:= Table[i + j^2, {i, 2, 6, 1.5}, {j, 7, 12, 0.1}]
```

```
Out[ ]:= {{51., 52.41, 53.84, 55.29, 56.76, 58.25, 59.76, 61.29, 62.84, 64.41, 66., 67.61, 69.24, 70.89,
72.56, 74.25, 75.96, 77.69, 79.44, 81.21, 83., 84.81, 86.64, 88.49, 90.36, 92.25, 94.16,
96.09, 98.04, 100.01, 102., 104.01, 106.04, 108.09, 110.16, 112.25, 114.36, 116.49, 118.64,
120.81, 123., 125.21, 127.44, 129.69, 131.96, 134.25, 136.56, 138.89, 141.24, 143.61, 146.},
{52.5, 53.91, 55.34, 56.79, 58.26, 59.75, 61.26, 62.79, 64.34, 65.91, 67.5,
69.11, 70.74, 72.39, 74.06, 75.75, 77.46, 79.19, 80.94, 82.71, 84.5, 86.31,
88.14, 89.99, 91.86, 93.75, 95.66, 97.59, 99.54, 101.51, 103.5, 105.51,
107.54, 109.59, 111.66, 113.75, 115.86, 117.99, 120.14, 122.31, 124.5, 126.71,
128.94, 131.19, 133.46, 135.75, 138.06, 140.39, 142.74, 145.11, 147.5},
{54., 55.41, 56.84, 58.29, 59.76, 61.25, 62.76, 64.29, 65.84, 67.41, 69., 70.61, 72.24, 73.89,
75.56, 77.25, 78.96, 80.69, 82.44, 84.21, 86., 87.81, 89.64, 91.49, 93.36, 95.25, 97.16,
99.09, 101.04, 103.01, 105., 107.01, 109.04, 111.09, 113.16, 115.25, 117.36, 119.49, 121.64,
123.81, 126., 128.21, 130.44, 132.69, 134.96, 137.25, 139.56, 141.89, 144.24, 146.61, 149.}}
```

```
In[ ]:= f[r_, c_] := r^2 + c + 2;
Table[f[i, j], {j, 2, 8, 2}, {i, 1, 5}]
```

```
Out[ ]:= {{5, 8, 13, 20, 29}, {7, 10, 15, 22, 31}, {9, 12, 17, 24, 33}, {11, 14, 19, 26, 35}}
```

```
In[ ]:= ? Array
```

Symbol i

`Array[f, n]` generates a list of length n , with elements $f[i]$.

`Array[f, n, r]` generates a list using the index origin r .

`Array[f, n, {a, b}]` generates a list using n values from a to b .

`Array[f, {n1, n2, ...}]` generates an $n_1 \times n_2 \times \dots$ array of nested lists, with elements $f[i_1, i_2, \dots]$.

`Array[f, {n1, n2, ...}, {r1, r2, ...}]` generates a list using the index origins r_i (default 1).

`Array[f, {n1, n2, ...}, {{a1, b1}, {a2, b2}, ...}]` generates a list using n_i values from a_i to b_i .

`Array[f, dims, origin, h]` uses head h , rather than List, for each level of the array.

▼

```
In[ ]:= ff[x_] := x^2 + 1;
Array[ff, 3]
```

```
Out[ ]:= {2, 5, 10}
```

```
In[ ]:= Array[f, {2, 3}]
```

```
Out[ ]:= {{4, 5, 6}, {7, 8, 9}}
```

```
In[ ]:= Array[ff, 11, 0]
```

```
Out[ ]:= {1, 2, 5, 10, 17, 26, 37, 50, 65, 82, 101}
```