```
In[1]:= ?Sum[i, {i, 1, 99, 2}]
        求和

Out[1]= Information[2500, LongForm → False]

In[2]:= Sum[2 n - 1, {n, 1, 50}]
        求和

Out[2]= 2500

In[3]:= Product[2 n - 1, {n, 1, 50}]
        乘积

Out[3]= 2 725 392 139 750 729 502 980 713 245 400 918 633 290 796 330 545 803 413 734 328 823 443 106 201 \
         171 875

In[4]:= Sum[n^2, {n, 1, 50}]
        求和

Out[4]= 42 925

In[5]:= Product[n^2, {n, 1, 50}]
        乘积

Out[5]= 925 017 065 282 507 919 013 470 723 235 883 682 349 486 807 421 901 987 706 139 271 018 810 570 \
         717 360 434 442 383 213 140 448 215 302 144 000 000 000 000 000 000 000 000

In[6]:= Sum[1 / 2 n, {n, 1, 50}]
        求和
```

Out[6]= $\frac{1275}{2}$

```
In[7]:= Product[1 / 2 n, {n, 1, 50}]
        乘积
```

Out[7]= $\frac{216105129892080882169214875191192738017616943359375}{8}$

```
In[8]:= ?PrimeQ
```

Out[8]=

Symbol

PrimeQ[*expr*] 如果 *n* 是一个素数，产生 True，其它情况下产生 False.

In[10]:=
```
?PrimePi
```

Out[10]=

Symbol

PrimePi[x] 给出小于等于 x 的素数 π(x) 的数目.

In[11]:=
```
?Prime
```

Out[11]=

Symbol

Prime[n] 给出第 n 个质数 p_n.

In[12]:=
```
Times @@ Prime[Range[PrimePi[100]]]
乘        素数  范围  素数Pi的数目
```

Out[12]=
```
2 305 567 963 945 518 424 753 102 147 331 756 070
```

In[15]:=
```
Total[Prime[Range[PrimePi[100]]]]
总计  素数  范围  素数Pi的数目
```

Out[15]=
```
1060
```

In[16]:=
```
Prime[Range[PrimePi[100]]]
素数  范围  素数Pi的数目
```

Out[16]=
```
{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31,
 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97}
```

In[17]:=
```
(*随机删除list中的三个元素*)
```

In[18]:=
```
list = Range[1, 10]
       范围
```

Out[18]=
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

In[21]:=
```
newlist = Complement[list, RandomSample[list, 3]]
          补集              伪随机采样
```

Out[21]=
```
{3, 4, 5, 6, 8, 9, 10}
```

In[22]:=
```
?Table
```

Out[22]=

Symbol

Table[*expr*, *n*] 产生 *expr* 的 *n* 个拷贝的列表.

Table[*expr*, {*i*, i_{max}}] 产生 *i* 从1到 i_{max} 的 *expr* 值的列表.

Table[*expr*, {*i*, i_{min}, i_{max}}] 从 $i = i_{min}$ 开始.

Table[*expr*, {*i*, i_{min}, i_{max}, *di*}] 使用步长 *di*.

Table[*expr*, {*i*, {i_1, i_2, …}}] 使用连续值 i_1、 i_2、 ….

Table[*expr*, {*i*, i_{min}, i_{max}}, {*j*, j_{min}, j_{max}}, …] 给出一个嵌套列表. 和 *i* 相关联的列表是最外的列表.

In[23]:=
```
?If
```

Out[23]=

Symbol

If[*condition*, *t*, *f*] 如果 *condition* 计算为 True 则给出 *t*，若计算为 False 则给出 *f*.

If[*condition*, *t*, *f*, *u*] 如果 *condition* 既不计算为 True 也不计算为 False 则给出 *u*.

In[26]:=
```
Table[If[i < j, Nothing, 10 * i + j], {i, 1, 4}, {j, 1, 4}]
表格    如果          无（会自动被删除）
```

Out[26]=
```
{{11}, {21, 22}, {31, 32, 33}, {41, 42, 43, 44}}
```

In[27]:=
```
?RandomReal
```

Out[27]=

Symbol

RandomReal[] 给出一个在 0 到 1 范围内的伪随机实数.

RandomReal[{x_{min}, x_{max}}] 给出一个在 x_{min} 到 x_{max} 范围内的伪随机实数.

RandomReal[x_{max}] 给出一个在 0 到 x_{max} 范围内的伪随机实数.

RandomReal[*range*, *n*] 给出 *n* 个伪随机实数组成的列表.

RandomReal[*range*, {n_1, n_2, …}] 给出伪随机实数组成的 $n_1 \times n_2 \times \ldots$ 数组.

In[28]:=
```
matrix = RandomReal[{5.2, 9.7}, {4, 4}]
               伪随机实数
```

Out[28]=
```
{{9.65558, 7.57773, 6.99457, 9.4824}, {5.49424, 6.39368, 7.98169, 6.3261},
 {6.44011, 7.72489, 8.99503, 5.47703}, {7.64163, 7.72405, 5.36274, 6.96835}}
```

In[29]:=
```
Flatten[matrix]
压平
```

Out[29]=
```
{9.65558, 7.57773, 6.99457, 9.4824, 5.49424, 6.39368, 7.98169, 6.3261,
 6.44011, 7.72489, 8.99503, 5.47703, 7.64163, 7.72405, 5.36274, 6.96835}
```

In[30]:=
```
Min[Flatten[matrix]]
…  压平
```

Out[30]=
```
5.36274
```

In[31]:=
```
Max[Flatten[matrix]]
…  压平
```

Out[31]=
```
9.65558
```

```
(*定义圆盘参数*)circle1 = {{x1, y1}, r1};
circle2 = {{x2, y2}, r2};
circle3 = {{x3, y3}, r3};

(*定义要检查的点*)
point = {x, y};

(*计算到每个圆盘中心的距离*)
distance1 = Sqrt[(point[[1]] - circle1[[1, 1]])^2 + (point[[2]] - circle1[[1, 2]])^2];
            平方根
distance2 = Sqrt[(point[[1]] - circle2[[1, 1]])^2 + (point[[2]] - circle2[[1, 2]])^2];
            平方根
distance3 = Sqrt[(point[[1]] - circle3[[1, 1]])^2 + (point[[2]] - circle3[[1, 2]])^2];
            平方根

(*判断点是否在所有圆盘内*)
coveredByAllCircles =
  distance1 ≤ circle1[[2]] && distance2 ≤ circle2[[2]] && distance3 ≤ circle3[[2]];

(*输出结果*)
coveredByAllCircles
```

Out[40]=

$$\sqrt{(x - x1)^2 + (y - y1)^2} \le r1 \,\&\&\, \sqrt{(x - x2)^2 + (y - y2)^2} \le r2 \,\&\&\, \sqrt{(x - x3)^2 + (y - y3)^2} \le r3$$

In[41]:=
```
? Expand
```

Out[41]=

Symbol

Expand[*expr*] 用来展开 *expr* 中的乘积和正整数幂.

Expand[*expr*, *patt*] 用来使得不具有模式 *patt* 的 *expr* 任意部分不被展开.

In[42]:=
```
Expand[(x + 1) (x^2 - 2 x + 3)]
```
展开

Out[42]=

$3 + x - x^2 + x^3$

In[43]:=
```
expr = (x - 2) (x^2 + 2 x + 4) + (x + 5) (x^2 - 5 x + 25)
```

Out[43]=

$(5 + x) \left(25 - 5 x + x^2\right) + (-2 + x) \left(4 + 2 x + x^2\right)$

In[44]:=
```
Simplify[expr] /. x → -4
```
化简

Out[44]=

-11

In[45]:=
```
Simplify[expr]
```
化简

Out[45]=

$117 + 2 x^3$

In[46]:=
```
? Factor
```

Out[46]=

Symbol

Factor[*poly*] 在整数上对一个多项式分解因式.

Factor[*poly*, Modulus → *p*] 以素数 *p* 为模对多项式分解因式.

Factor[*poly*, Extension → {a_1, a_2, …}]

对一个多项式分解因式，而这个多项式的系数允许是代数数 a_i 的有理组合.

In[47]:=
```
Factor[x^5 - x^3]
```
因式分解

Out[47]=

$(-1+x)\ x^3\ (1+x)$

In[48]:=
```
Factor[16 - x^4]
```
因式分解

Out[48]=

$-\left((-2+x)\ (2+x)\ \left(4+x^2\right)\right)$

In[49]:=
```
Factor[(x + y)^2 - 10 (x + y) + 25]
```
因式分解

Out[49]=

$(-5+x+y)^2$

```
expr = 3 * a * x + 4 * b * y + 4 * a * y + 3 * b * x;
Factor[expr]
```
因式分解

Out[51]=

$(a+b)\ (3\,x+4\,y)$

In[57]:=

$$\text{expr} = \frac{1}{x+1} - \frac{x+3}{x\hat{}2-1} * \frac{x\hat{}2-2x+1}{x\hat{}2+4x+3};$$

In[53]:=
```
FullSimplify[expr]
```
完全简化

Out[53]=

$$\frac{2}{(1+x)^2}$$

In[56]:=

$$\text{expr} = \frac{a}{(a-b)\ (a-c)} + \frac{b}{(b-c)\ (b-a)} + \frac{c}{(c-a)\ (c-b)};$$

In[55]:=
```
FullSimplify[expr]
```
完全简化

Out[55]=

0

In[62]:=
```
Solve[a*b*x^2 + (a^4 + b^4) x + a^3*b^3 == 0, x, Assumptions → ab ≠ 0]
```
解方程 假设

Out[62]=

$$\left\{\left\{x \to -\frac{a^3}{b}\right\}, \left\{x \to -\frac{b^3}{a}\right\}\right\}$$

In[69]:=
```
eqns = {x^2 + 2*x*y + y^2 == 9, (x - y)^2 - 3 (x - y) - 10 == 0};
Solve[eqns, {x, y}]
```
解方程

Out[70]=

$$\left\{\left\{x \to -\frac{5}{2}, y \to -\frac{1}{2}\right\}, \left\{x \to \frac{1}{2}, y \to \frac{5}{2}\right\}, \{x \to 1, y \to -4\}, \{x \to 4, y \to -1\}\right\}$$

In[71]:=
```
Sum[k^5, {k, m, n}]
```
求和

Out[71]=

$$-\frac{1}{12}(-1 + m - n)(m + n)\left(m + m^2 - 4 m^3 + 2 m^4 - n - 2 m n + 2 m^2 n + n^2 - 2 m n^2 + 2 m^2 n^2 + 4 n^3 + 2 n^4\right)$$

In[77]:=
```
RSolve[{b[n + 1] == (2*b[n] + 3)/(b[n] + 4), b[0] == 0}, b[n], n]
```
解递推方程

Out[77]=

$$\left\{\left\{b[n] \to \frac{3\left((-1)^n - \left(-\frac{1}{5}\right)^n\right)}{3(-1)^n + \left(-\frac{1}{5}\right)^n}\right\}\right\}$$

In[76]:=
```
?RSolve
```

Out[76]=

Symbol

RSolve[*eqn*, *a*[*n*], *n*] 解递推方程，求 *a*[*n*].

RSolve[{eqn_1, eqn_2, …}, {$a_1[n]$, $a_2[n]$, …}, *n*] 求解递推方程组.

RSolve[*eqn*, *a*[n_1, n_2, …], {n_1, n_2, …}] 求解部分递推方程.