# 第7讲 自定义函数和模式替换

## 7－1 自定义函数

所有的输入的实体都是符号表达式；

所有的操作都是调用变换规则（Transformation rules）对表达式求值；

表达式求值过程就是将表达式从一种表示形式变换为另一种表示形式的过程。

在符号计算环境下并不强调函数与命令的区别。

例：求和函数Sum和绘图Plot等都可看成一个变换规则。

xx = 123　　对表达式xx定义了一条变换规则

### 1. 模式　x_　　_blank

本核心语言的一个独特的优势，其强大的、简洁明了的、 高可读性的符号语言

例1： 定义函数 f (x) = x - 1.

```
f[x_] := x - 1   (*或 f[x_]=x -1*)

{f[10], f[a]}

s = {{1, 2}, {3, 4}}; f[s]
```

例2：变量是矩阵

```
g[x] = 1 + 2 x

g[10] + g[x]   (* 系统只认识g[x]，g[10]无定义 *)
```

例3：同名函数不同模式

```
f[x_, y_] := x + y
f[x_, x_] := Sin[x] + Cos[x]
{f[11], f[3, 3], f[3, 5]}
```

例4：看看f的所有的函数定义

```
? f
```

| 模式定 义 | 说 明 |
|---|---|
| f[x_] | 定义时命名为 x的任意表达式， |
| f[x_, y_] | 命名为 x，y的任意表达式 |
| f[x_, x_] | 命名为 两个相同的任意表达式 |
| x^n_ | x 的任意幂次，幂次为n |
| x_^n_ | 任意表达式的任意幂次 |
| {a_, b_} | 含有两个表达式的表 |

### 2. := 与 = 延时赋给与立即赋给

*lhs* := *rhs*　　*rhs*保留未计算的形式

例5：观察延时赋给与立即赋给的区别

```
x = 1; fa[x_] := 2 + x

ga[x_] = 2 + x
```

```
{fa[5], ga[5]}
```

例6：计算数据表list的算术平均值.

```
mean[list_] := Apply[Plus, list] / Length[list]
mean[{1, 3, 5, 7, 9}]
```

例7：计算方阵的算术平均值

```
aver[m_] := (n = Length[m]; Sum[m[[i, j]], {i, 1, n}, {j, 1, n}] / n^2)
aver[{{1, 2}, {3, 10}}]
```

### 3. x_, x__, x___

x__ 一个或多个变量， x___ 零个、一个或多个变量

例8：观察定义和调用时变量的数量

```
h[x__] := Plus[x]
h[2, 3, 4]
h[{2, 3, 4}]
```

### 4. 模式类型说明

| 模式 | 说 明 | 模式 | 说 明 |
|---|---|---|---|
| x_ | 任何表达式， | x_Integer | 任何整数 |
| x_Real | 任何近似实数， | x_Complex | 任何复数 |
| x_List | 任何表 | x_Matrix | 变量是矩阵 |
| x_h | 任何头为h的表达式 | | |

例9：定义变量为整型

```
ua[x_Integer] := (x - 1) (x + 1)
{ua[a + b], ua[3], ua[1.1]}
```

对于更复杂的模式，例如，模式是y的多项式，元素为数值的矩阵等限定条件。则用

```
pattern?test
```

问号表示对模式测试是否满足test的逻辑条件。仅当测试函数的值为真时才调用函数。

例10：定义变量为 数值类型

```
uv[x_?NumericQ] := x + 2
{uv[1.2], uv[3], uv[4 / 5], uv[2 + 3 I], uv[xyz]}
```

| 测试函数 | 是否为 |
| --- | --- |
| NumberQ[expr] | 数 |
| NumericQ[expr] | 数值类型 |
| IntegerQ[expr] | 整数 |
| OddQ[expr] / EvenQ[expr] | 奇数 / 偶数 |
| Positive[x] / Negative[x] | 正数 / 负数 |
| NonPositive[x] / NonPositive[x] | 非正数 / 非负数 |
| MatrixQ[expr] | 矩阵 |
| MemberQ[list, form] | list中是否有form的元素 |
| TrueQ[expr] | 逻辑值为真 |
| PolynomialQ[expr, {x1, x2, ..}] | expr是否为x1, x2, ... 的多项式 |
| FreeQ[expr, form] | expr 中是否没有与form匹配的部分 |
| MatchQ[expr, form] | 模式 form 是否与 expr 匹配 |

例11：模式匹配吗？

```
{MatchQ[a^b, _ + _], MatchQ[a^b, _^_], MatchQ[a^b, _], MatchQ[a^b, __]}
```