

Function 类说明文档。

简介: *Function* 类是 *MathBox* 中的基础类型。主要用于储存用户输入的字符函数表达式, 其可进行表达式的拼接和函数求值运算。

基本用法:

1. 赋值运算。对于赋值运算在声明变量时可以有两种方式。一种是对已存在的函数变量进行表达式运算来求, 此时新的函数的自变量继承于参与运算的函数的自变量。另一种是从声明自变量开始, 然后输入自变量构成的表达式来形成函数。

#示例

```
Function f(x,y)=x+y;
```

```
Function f2=f+f;
```

```
Function f3=f+f2+f*f2
```

```
f3=Cos(f3+Sin(f3+f2))+Pow(f+f2,3.21+f3);
```

#输出

```
>>New Function
```

```
@ f=x+y
```

```
>>New Function
```

```
@ f2=x+y+x+y
```

```
>>New Function
```

```
@ f3=x+y+x+y+x+y+(x+y)*(x+y+x+y)
```

```
>>Rewrite Function
```

```
@f3=Cos(x+y+x+y+x+y+(x+y)*(x+y+x+y)+Sin(x+y+x+y+x+y+(x+y)*(x+y+x+y)+x+y+x+y))+Pow(x+y+(x+y+x+y),3.21+x+y+x+y+x+y+(x+y)*(x+y+x+y))
```

2. 表达式运算。对于已经初始化的 *Function* 变量，可直接输入表达式进行运算，结果保存在 *Result* 里面。

#示例

```
Function f1(x,y,z)=x*y*z+x+y;
```

```
Function f2(x,i)=Cos(x+i)+i;
```

```
f1+f2+Cos(f1+f2)+Sin(f1+f2)*Atan(f1*f2+1.321*f1);
```

#输出

```
>>New Function
```

```
@ f1=x*y*z+x+y
```

```
>>New Function
```

```
@ f2=Cos(x+i)+i
```

```
>>The result is
```

```
x*y*z+x+y+Cos(x+i)+i+Cos(x*y*z+x+y+Cos(x+i)+i)+Sin(x*y*z+x+y+Cos(x+i)+i)*Atan((x*y*z+x+y)*(Cos(x+i)+i)+1.321*(x*y*z+x+y))
```

3. 求值运算。对于已经赋值的 *Function* 变量可以直接进行赋值运算求出数值并可以将结果保存在一个 *Number* 类的变量里，默认保存在 *Result* 变量里。

#示例

```
Function f1(x,y,z)=x*y*z+y
```

```
Number save;
```

```
f1(1,2,3);
```

```
f1(1,2,3,save);
```

#输出

>>The function's value is 9

>>The function's value is 9

>>Rewrite Number

@ save=9

4. 用于各需要函数表达式和函数变量的过程当中。例如 *Pro_RandFunction* 过程是用来产生随机函数的数值过程，其有两个参量一个是一个大于 0 的 *Number* 表达式表示自变量的个数，另一个是已存在的 *Function* 变量用于保存随机函数的值。

#示例

Function f(x)=x;

Number a=3;

Pro_RandFunction(a+1,f);

#输出

>>*New Function*

@ f=x

>>*New Number*

@ a=10

>>Rewrite Function

@

f=+522.593-672.981/x1-x2/x0/x0/x2+x2/367.787+x0+246.102+x3

-x2/x1*271.295*x1+Atan(x3/174.104/x0-331.727)/x2+x3+x1*939.

096+262.251+704.647