

修改表达式的传入参数

目标:

将 $s[\{a,s,d\}]$ 变成 $s[\{a,s,d,x\}]$

```
In[ ]:= t = s[{a, s, d}]
Out[ ]:=
s[{a, s, d}]

In[ ]:=
Block[{t}, t /. s[r_] -> s[r ~Append~ x]]
|块 |追加

Append: Append[r, x] 中的位置 1 处应该是非原子表达式.

Out[ ]:=
s[{a, s, d}]
```

上面的方式该如何修改

使用Part和Append

```
In[ ]:= s[{a, s, d}, g][[0]]
s[{a, s, d}, g][[1]]
s[{a, s, d}, g][[2]]
s[{a, s, d}, g][[3]]

Out[ ]:=
s

Out[ ]:=
{a, s, d}

Out[ ]:=
g

Part: s[{a, s, d}, g] 的部分 3 不存在.

Out[ ]:=
s[{a, s, d}, g][[3]]
```

Part 总是对 FullForm 形式的表达式进行运算:

```
In[ ]:= (x / y) // FullForm
|完全格式

Out[ ]//FullForm=
Times[x, Power[y, -1]]

In[ ]:= (x / y)[[2]]
Out[ ]:=
1
-
y
```

```
In[ ]:= Block[{t}, t[[0]][t[[1]]~Append~x]]
      块      追加
```

... Part: 部分指定 t[[1]] 比对象深度更长.

... Part: 表达式 x 不能作为部分指定使用.

... Symbol: Symbol[t[[1, x]]] 中位置 1 处应该是字符串.

... Part: 表达式 x 不能作为部分指定使用.

... Symbol: Symbol[s[{a, s, d}][[1, x]]] 中位置 1 处应该是字符串.

```
Out[ ]:=
Symbol[s[{a, s, d}][[1, x]]]
```

```
In[ ]:= Block[{t}, t[[1]]]
      块
```

... Part: 部分指定 t[[1]] 比对象深度更长.

```
Out[ ]:=
{a, s, d}
```

2. AppendTo

```
In[ ]:= Block[{t}, AppendTo[t[[1]], x]; t]
      块      附加
```

... Part: 部分指定 t[[1]] 比对象深度更长.

... AppendTo: t[[1]] 不是一个含有值的变量, 因此它的值不能改变.

```
Out[ ]:=
s[{a, s, d}]
```

```
In[ ]:= t
```

```
Out[ ]:=
s[{a, s, d}]
```