

基础语法

表达式

ReplaceAll (/.) "(全部替换)"
MapAll (//@) "(映射全部)"
Map (/@) "(映射)"
Postfix "(后缀)"
Prefix "(前缀)"
Infix "(中缀)"
Function (&,->,⟶) "(纯函数)"

```
In[ ]:=  
{a, b, c, d} /. b -> x
```

```
Out[ ]:=  
{a, x, c, d}
```

```
In[ ]:=  
Map[f, {a, b, c, d, e}]  
|映射
```

```
Out[ ]:=  
{f[a], f[b], f[c], f[d], f[e]}
```

```
In[ ]:=  
f /@ {a, b, c, d, e}
```

```
Out[ ]:=  
{f[a], f[b], f[c], f[d], f[e]}
```

```
In[ ]:=  
f ~Map~ {a, b, c, d, e}  
|映射
```

```
Out[ ]:=  
{f[a], f[b], f[c], f[d], f[e]}
```

```
In[ ]:=  
Infix[f[f, {a, b, c, d, e}], "⇒"]  
|中缀
```

```
Out[ ]:=  
f ⇒ {a, b, c, d, e}
```

```
In[ ]:=  
f → {a, b, c, d, e}
```

```
Out[ ]:=  
f → {a, b, c, d, e}
```

f # & /@

```
Out[ ]:=  
f
```