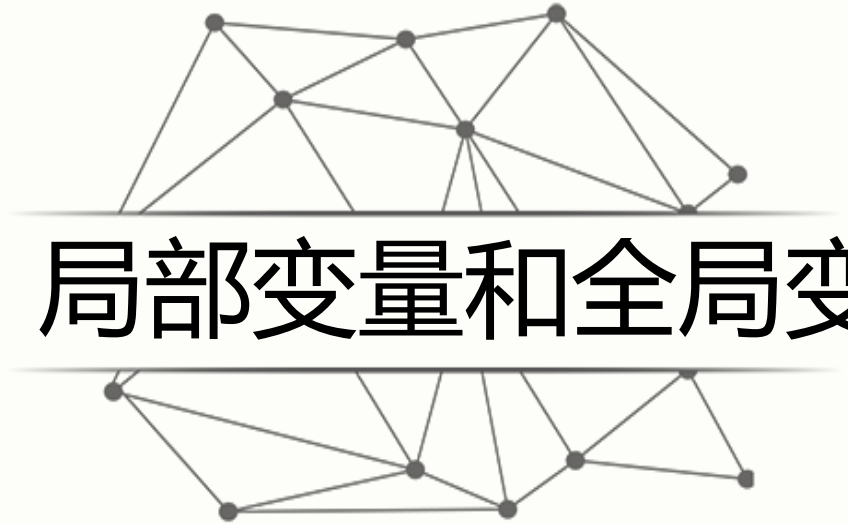
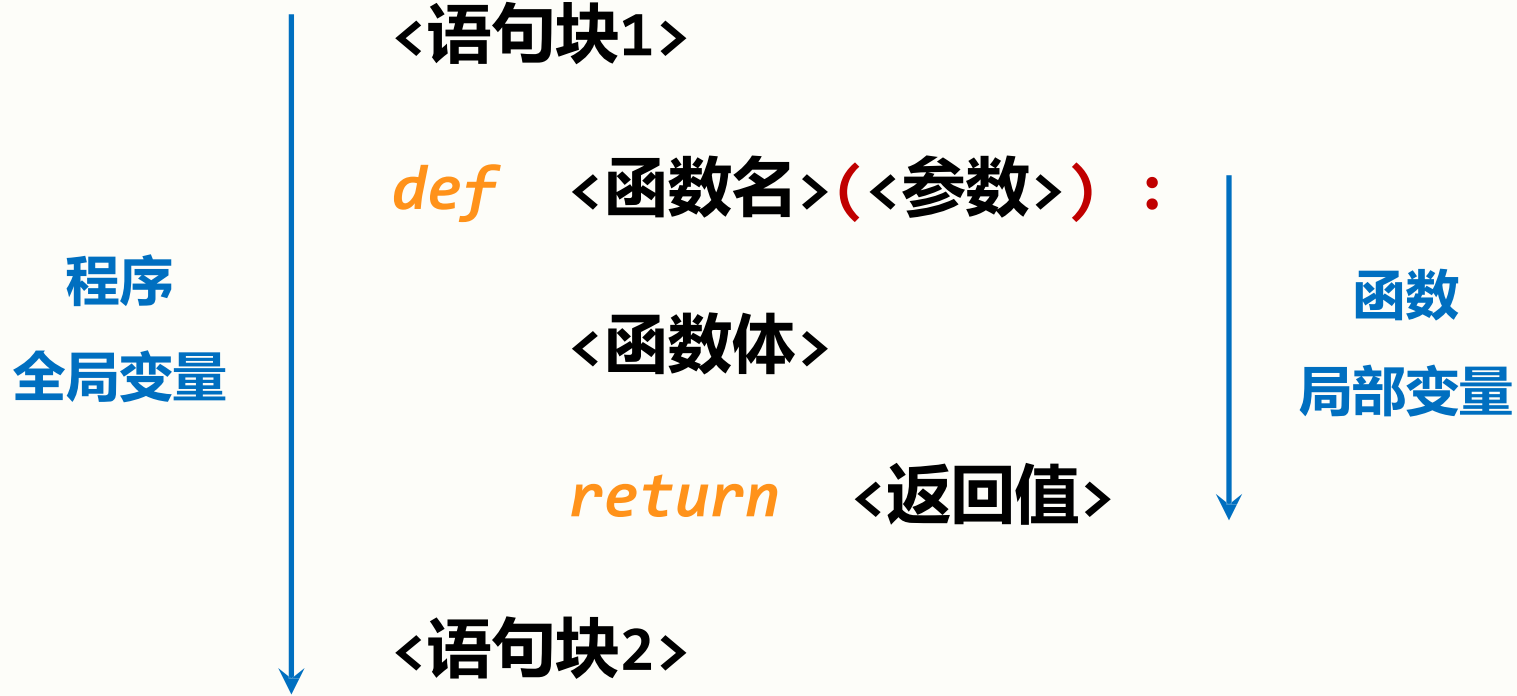


6.3 局部变量和全局变量

局部变量和全局变量



局部变量和全局变量

```
n, s = 10, 100
def fact(n) :
    s = 1
    for i in range(1, n+1):
        s *= i
    return s
print(fact(n), s)
```

n和s是全局变量

fact()函数中的n和s是局部变量

n和s是全局变量

运行结果

```
>>>
3628800 100
```

局部变量和全局变量

规则1: 局部变量和全局变量是不同变量

- 局部变量是函数内部的占位符，与全局变量可能重名但不同
- 函数运算结束后，局部变量被释放
- 可以使用*global*保留字在函数内部使用全局变量

局部变量和全局变量

```
n, s = 10, 100
def fact(n) :
    s = 1
    for i in range(1, n+1):
        s *= i
    return s
print(fact(n), s)
```

fact()函数中s是局部变量

与全局变量s不同

此处局部变量s是3628800

此处全局变量s是100

运行结果

```
>>>
3628800 100
```

局部变量和全局变量

```
n, s = 10, 100
def fact(n) :
    global s
    for i in range(1, n+1):
        s *= i
    return s
print(fact(n), s)
```

fact()函数中使用global保留字声明

此处s是全局变量s

此处s指全局变量s

此处全局变量s被函数修改

运行结果

```
>>>
362880000 362880000
```

局部变量和全局变量

规则2: 局部变量为序列数据类型且未创建，等同于全局变量

```
ls = ["F", "f"]
def func(a) :
    ls.append(a)
    return
func("C")
print(ls)
```

通过使用[]真实创建了一个全局变量列表ls

此处ls是列表类型，未真实创建

则等同于全局变量

全局变量ls被修改

运行结果

```
>>>
['F', 'f', 'C']
```

局部变量和全局变量

```
ls = ["F", "f"]
def func(a) :
    ls = []
    ls.append(a)
    return
func("C")
print(ls)
```

通过使用[]真实创建了一个全局变量列表ls

此处ls是列表类型，真实创建

ls是局部变量

局部变量ls被修改

运行结果

```
>>>
['F', 'f']
```

局部变量和全局变量

使用规则

- 基本数据类型，无论是否重名，局部变量与全局变量不同
- 可以通过global保留字在函数内部声明全局变量
- 序列数据类型，如果局部变量未真实创建，则是全局变量