

An aerial photograph of the Hong Kong skyline at sunset, featuring numerous skyscrapers and the Victoria Harbour. A semi-transparent monitor icon is overlaid on the right side of the image, displaying the Python logo (a blue and yellow interlocking snake) against the city background.

# Python语言基础与应用

Python语言概览 / 计算和控制流

陈斌 北京大学 [gischen@pku.edu.cn](mailto:gischen@pku.edu.cn)

# 计算和控制流

- › 计算与流程
- › 运算语句
- › 控制流语句
- › 定义语句

# 计算与流程

- › 对现实世界处理和过程的抽象
- › 各种类型的数据对象，可以通过各种运算组织成复杂的表达式

```
>>> 12 * 34.5 + 23.4
437.4
>>> ('abc' + '123') * 3
'abc123abc123abc123'
>>>
>>> import math
>>> math.sqrt(12)
3.4641016151377544
```



# 运算语句

› 将表达式赋值给变量进行引用

› 赋值语句用来实现处理与暂存

- 表达式计算

```
>>> n = 12 * 34
```

```
>>> n
```

```
408
```

- 函数调用

```
>>> p2 = math.sqrt(2)
```

```
>>> p2
```

```
1.4142135623730951
```

- 赋值

```
>>> pfg = math.sqrt
```

```
>>> pfg
```

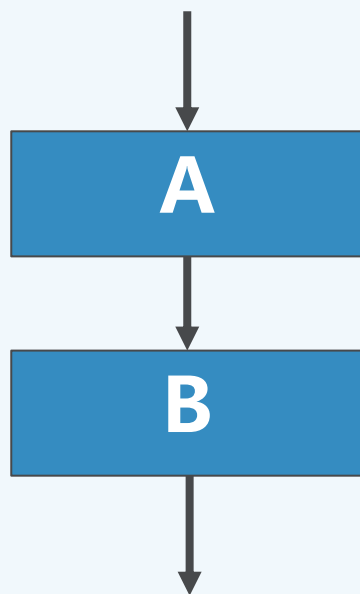
```
<built-in function sqrt>
```

```
>>> pfg(2)
```

```
1.4142135623730951
```

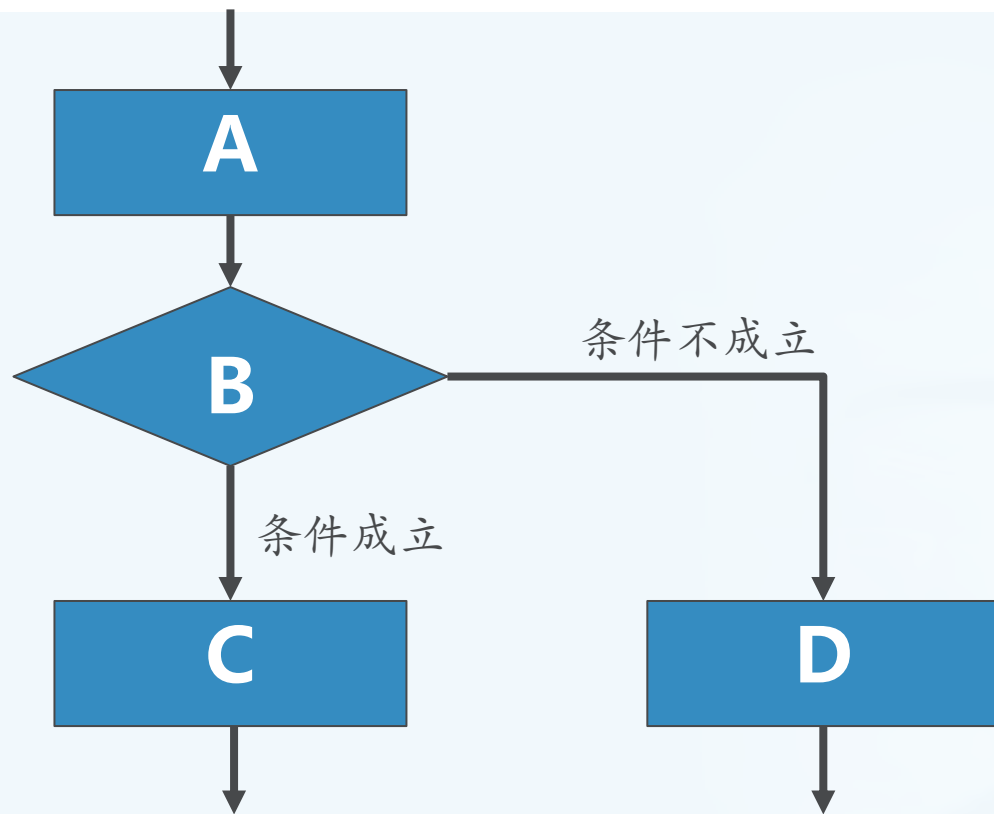
# 控制流语句

## › 控制流语句用来组织语句描述过程



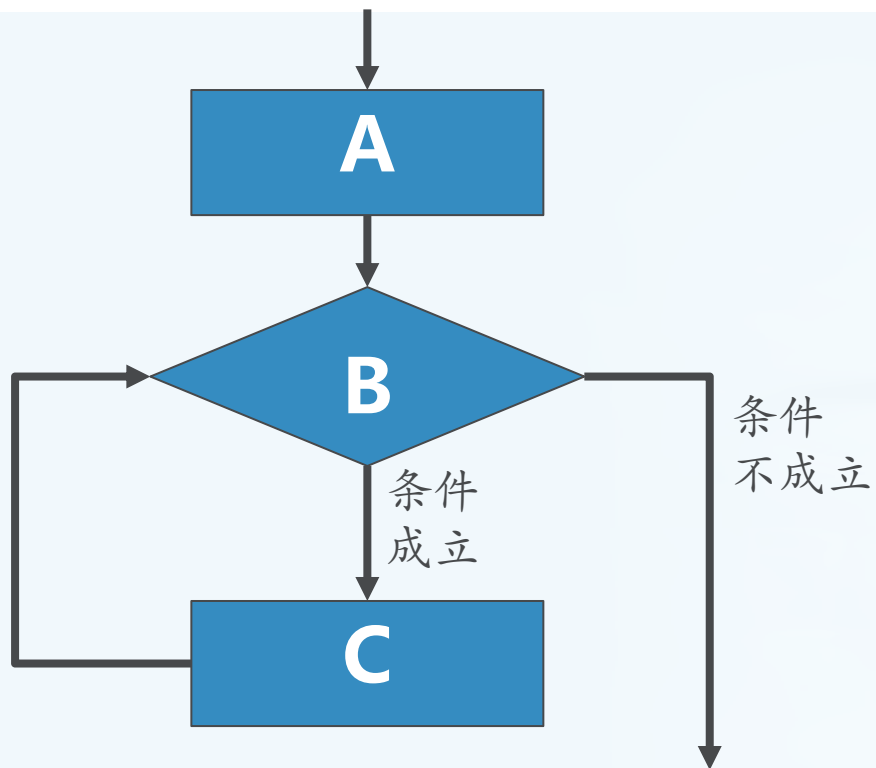
**顺序结构**

# 控制流语句



**条件分支: if**

# 控制流语句



循环结构for、while

# 定义语句：def、class

- › 定义语句也用来组织语句，把一系列运算语句集合起来给一个名字
- › 描述了一个包含一系列处理过程的计算单元，主要为了源代码的各种复用

```
1  def sum_list(alist): # 定义一个带参数的函数
2      sum_temp = 0
3      for i in alist:
4          sum_temp += i
5      return sum_temp # 函数返回值
6
7
8  print(sum_list) # 查看函数对象sum_list
9
10 my_list = [23, 45, 67, 89, 100]
11 # 调用函数，将返回值赋值给my_sum
12 my_sum = sum_list(my_list)
13 print("sum of my list:%d" % (my_sum,))
```



# 定义语句

- › 可以定义函数、类等“代码”对象
- › 调用函数或者类，也可以得到数据对象，Python里所有可调用的事物称为callable

- 函数定义
- 类定义

```
1 def sum_list(alist): # 定义一个带参数的函数
2     sum_temp = 0
3     for i in alist:
4         sum_temp += i
5     return sum_temp # 函数返回值
6
7
8 print(sum_list) # 查看函数对象sum_list
9
10 my_list = [23, 45, 67, 89, 100]
11 # 调用函数，将返回值赋值给my_sum
12 my_sum = sum_list(my_list)
13 print("sum of my list:%d" % (my_sum,))
```