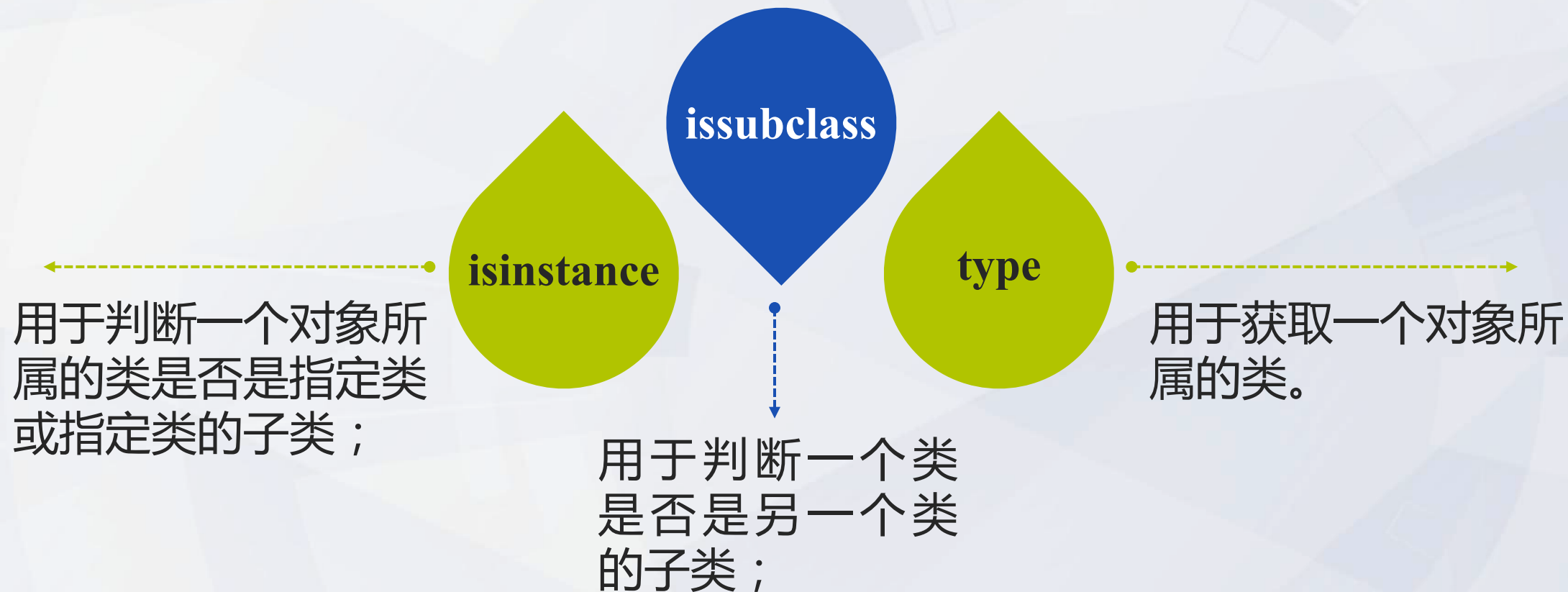


PYTHON

编程基础

内置函数instance、 issubclass和type

功能简介



示例



例：isinstance、issubclass和type使用示例。

```
1 class Person: #定义Person类
2     pass
3 class Student(Person): #以Person类作为父类定义子类Student
4     pass
5 class Flower: #定义Flower类
6     pass
```

示例



例：isinstance、issubclass和type使用示例。

```
7     if __name__ == '__main__':  
8         stu=Student() #创建Student类对象stu  
9         f=Flower() #创建Flower对象f  
10        print('stu是Person类或其子类对象：', isinstance(stu,Person))  
11        print('stu是Student类或其子类对象：', isinstance(stu,Student))
```

stu是Person类或其子类对象： True

stu是Student类或其子类对象： True

示例



例：isinstance、issubclass和type使用示例。

```
12     print('f是Person类或其子类对象：', isinstance(f, Person))  
13     print('Student是Person类的子类：', issubclass(Student, Person))  
14     print('Flower是Person类的子类：', issubclass(Flower, Person))
```

f是Person类或其子类对象： False

Student是Person类的子类： True

Flower是Person类的子类： False

示例



```
15     print('stu对象所属的类：', type(stu))
16     print('f对象所属的类：', type(f))
17     print('stu是Person类对象：', type(stu)==Person)
18     print('stu是Student类对象：', type(stu)==Student)
```

stu对象所属的类： <class '__main__.Student'>

f对象所属的类： <class '__main__.Flower'>

stu是Person类对象： False

stu是Student类对象： True

示例



提示：

如果我们要判断一个对象的类型是否是指定类或该类的子类，则可以使用`isinstance`函数。

如果我们要判断一个对象的类型是否是指定类，则可以使用“`type(对象名)==类名`”的方式。