

初识类



一、课程介绍

本节课将开始学习类的知识，涉及类的属性、方法、实例化、继承等方面。

```
1 import sys
2 from PySide2.QtWidgets import *
3 from ui_hello import Ui_Hello
4
5 class Hello(QMainWindow, Ui_Hello):
6     def __init__(self):
7         super().__init__()
8         self.setupUi(self)
9         self.show()
10
11 if __name__ == '__main__':
12     app = QApplication(sys.argv)
13     window = Hello()
14     sys.exit(app.exec_())
```

二、知识点

类与对象

类的属性和方法

类的实例化

类的继承



三、重难点解析



类

类是 Python 中一个重要的概念，它是一种把对象分组归类的方法。
类的创建方法如下：

```
1 class 类名():
2     (缩进)代码块
```

用关键字 `class` 来定义类，后面跟着类名，接着是一个小括号。
示例：

```
1 class Cat:
2     print( "我是猫")
```



类的属性与方法

一般情况下，一个类具有属性和方法。

属性就是类具有的一些数据信息（数字、字符串等等），本质就是变量，只不过包含在类中。

方法就是类具有的行为，也就是它能做的事情。

总结：属性是类中定义的变量，方法就是类中定义的函数，函数能做到的，方法都可以做到，包括传递参数和返回值。

```
1 类 = 属性 + 方法
```

利用类，可以把某样事物的属性和方法收集在一起，属性就是它的特征信息，方法就是行为。
示例：

```
1 class Cat():
2     color = "灰色"
3
4     def cry(self):
5         print( "喵喵喵~")
```

`Cat` 类拥有属性 `color`，属性值为“灰色”；拥有方法 `cry()`，方法的作用是打印“喵喵喵”。
`self` 是类中每个方法必须有的第一个参数，表示创建时的具体对象。



类的实例化

通过类的实例化即可创建一个对象，对象拥有类中定义的属性与方法。
实例化代码如下：

```
1 对象名 = 类名()
```

对象可以使用点号访问属性和调用方法，格式如下：

```
1 对象名. 属性
2 对象名. 方法(参数)
```

示例：

`Cat` 类实例化得到 `lvcha` 对象，打印它的 `color` 属性值和调用 `cry()` 方法。

```
1 lvcha = Cat()
2 print(lvcha.color)
3 lvcha.cry()      # 此时方法的 self 参数值就是对象 lvcha
```

对象属性的值是可以修改的，与变量的赋值一样。格式如下：

```
1 对象名. 属性 = 新值
```

示例：

```
1 lvcha = Cat()
2 lvcha.color = "灰色"
3 print(lvcha.color)
4 panghu = Cat()
5 panghu.color = "橘色"
6 print(panghu.color)
```

控制台

```
灰色
橘色
程序运行结束
```



__init__() 方法

__init__() 方法（init的左右两边各有两个下划线，）的作用是给类做初始化工作，也被称为构造方法，在创建对象时被调用执行。

构造方法用来创建对象时执行初始化的操作，它不是必须定义的。

在创建对象时，直接在类名后的小括号里传递的参数会 __init__() 方法自动接收，然后在 __init__() 方法内部将接收到的参数赋值给类中对应的属性值即可。

在类中访问类的属性和调用类的方法采用如下格式：

```
1 self. 属性
2 self. 方法()
```

示例：

```
1 class Cat():
2     def __init__(self, color):
3         self.color = color
4
5     def cry(self):
6         print("喵喵喵~")
7
8 lvcha = Cat(color="灰色")
9 print(lvcha.color)
10 lvcha.cry()
```



类的继承

在 Python 中，一个类可以继承其他类的属性和方法。

继承其他类属性或方法的类，被称为子类；被继承的类称为父类。

类继承的格式如下：

```
1 子类(父类)
```

继承父类的子类，就拥有了父类所有的属性和方法。

示例：

```

1 class Cat():
2     def __init__(self, name, color):
3         self.name = name
4         self.color = color
5
6     def cry(self):
7         print("喵喵喵~")
8
9
10 class SongCat(Cat):
11     def sing(self):
12         print("我会唱歌~")
13
14
15 class DanceCat(Cat):
16     def dance(self):
17         print("我会跳舞~")

```

以上代码为类的单继承，即子类只继承一个父类。

还有一种情况是类的多继承，当继承的父类大于等于 2 个时，这样的继承就称为多继承。

注意：多继承时，小括号中从左至右，如果继承的父类中都具有一个相同的属性或方法，那么子类只会保留最左边父类中的属性或方法。

示例：

```

1 class AlCat(SongCat, DanceCat):
2     pass

```



super().__init__()

super().__init__() 方法的主要作用是调用父类中的 __init__() 方法，确保父类被正确的初始化，从而使子类显示继承父类的某些属性与方法。

四、知识延伸



None

None 在 Python 中表示一个空对象，常用于设置还不确定的值。

使用 if 判断值为 None 的变量时，结果是 False。

```

1 a = None
2 if a:
3     print("True")
4 else:
5     print("False")

```

控制台

```
False
程序运行结束
```



单词卡

class [klas]

n. 类

例句：They belong to the same **class**.

翻译：他们属于同一类人。

object [əb'dʒekt]

n. 对象

例句：**Objects** are created by classes.

翻译：对象是由类创建的。

self [self]

n. 自己

例句：I saw in him a mirror image of my younger **self**.

翻译：我在他身上看见了年轻时的自己。

五、课中练习



视频1-初识类

题目

本节课我们将会学习什么知识点？

选项



A Python 中的类

B JavaScript



视频2-类的属性和方法

题目

关于类的描述，以下哪个代码块是正确的？

①

```
1 class 类名() :  
2 代码块
```

②

```
1 对象名 = 类名
```

③

```
1 类 = 属性 + 方法
```

选项

A ①

B ②



C ③



视频3—类的实例化

题目

下列关于类的描述，哪些是正确的？

- ①类中必须包含 `__init__()` 方法
- ②如果把类（class）比作菜谱，那么对象（object）就是已经做好的菜。
- ③类中定义的函数还是叫做类函数
- ④类中定义的函数叫做方法，且它的第一个参数必须为 `self`

选项

A ①②

B ②③



C ②④

D ①④



视频4—类的单继承

题目

类的继承有什么作用？



视频5—类的多继承

题目

下面程序的输出结果是？

```
1 class A():
2     f1 = 1
3 class B():
4     f1 = 3
5 class C(A, B):
6     c = 6
7 new = C()
8 print(new.f1, new.c)
```

选项



A 1 6

B 3 6

C 1 3

D 6 6



视频6—改写案例

题目

下面代码的输出结果是？

```
1 class People :
2     def talk (self) :
3         pass
4 class Chinese (People) :
5     def talk (self) :
6         print( " 中国人在说")
7 class US(People) :
8     def talk (self) :
9         print( " 美国人在说")
10 c = Chinese()
11 u = US()
12 c.talk()
```



选项

A pass

B 美国人在说



C 中国人在说

D 程序报错

六、扩展练习

题目

定义一个 Bird 类，具有 name、color 属性和 fly、sing 方法，达到如下效果：

```
1 class Bird:
2     def __init__(self, name, color):
3         self.name = name
4         self.color = color
5     def fly(self):
6         print("我会飞")
7     def sing(self):
8         print("我会唱歌")
9
10
11
12
13 bird = Bird("小明", "白色")
14 bird.fly()
15 bird.sing()
```

输出结果为：

```
我是一只鸟，我会飞
我不仅会飞，还会唱歌
```

题目

定义具有的属性 name 和 age，bark 方法的狗类。

```

1
2
3
4
5
6
7
8
9
10
11 # 实例化类
12 a = MyDog("Tom", 11)
13
14 # 访问类中的属性
15 print(a.name)
16 print(a.age)
17
18 # 调用类中的方法
19 a.bark()

```

输出结果为：

```

Tom
11
汪汪汪，我是Tom，我的年龄是11

```

七、附录



视频1-初识类

解析

A。本节课我们将会学习 Python 中类的相关知识。



视频2-类的属性和方法

解析

C。①中代码块没有缩进。②中对象名与类名应该是不同的。



视频3-类的实例化

解析

C。



视频4-类的单继承

解析

使子类在类中不用定义，就直接使用父类中已有的属性和方法。



视频5-类的多继承

解析

A。

在 Python 中子类可以继承多个父类，当继承的父类大于等于 2 个时，这样的继承我们称之为多继承。多继承时，如果多个父类中具有同一个相同的属性，那么子类只会保留第一个父类中的属性。



视频6-改写案例

解析

C。对象 c 是由 Chinese 类实例化得到，它的 talk() 方法打印的是“中国人在说”。



扩展练习

解析

```
1 class Bird():
2     def __init__(self, name, color):
3         self.name = name
4         self.color = color
5     def fly(self):
6         print("我是一只鸟，我会飞")
7     def sing(self):
8         print("我不仅会飞，还会唱歌")
```

解析

```
1 # 定义狗类
2 class MyDog():
3     def __init__(self, name, age):
4         self.name = name
5         self.age = age
6     def bark(self):
7         print("汪汪汪，我是" + self.name + "，我的年龄是" + str(self.age))
8 # 实例化类
9 a = MyDog("Tom", 11)
10 # 访问类中的属性
11 print(a.name)
12 print(a.age)
13 # 调用类中的方法
14 a.bark()
```

