Ruby 语法 Ruby 类案例

Ruby 类和对象

Ruby 是一种完美的面向对象编程语言。面向对象编程语言的特性包括：

- 数据封装
- 数据抽象
- 多态性
- 继承

这些特性将在 面向对象的 Ruby 中进行讨论。

一个面向对象的程序，涉及到的类和对象。类是个别对象创建的蓝图。在面向对象的术语中，您的自行车是自行车类的一个实例。

以车辆为例，它包括车轮（wheels）、马力（horsepower）、燃油或燃气罐容量（fuel or gas tank capacity）。这些属性形成了车辆（Vehicle）类的数据成员。借助这些属性您能把一个车辆从其他车辆中区分出来。

车辆也能包含特定的函数，比如暂停（halting）、驾驶（driving）、超速（speeding）。这些函数形成了车辆（Vehicle）类的数据成员。因此，您可以定义类为属性和函数的组合。

类 Vehicle 的定义如下：

实例

```
Class Vehicle
{
Number no_of_wheels
Number horsepower
Characters type_of_tank
Number Capacity
Function speeding
{
}
Function driving
{
}
Function halting
{
}
}
```

通过给这些数据成员分配不同的值，您可以创建类 Vehicle 的不同实例。例如，一架飞机有三个轮子，马力 1,000，燃油罐容量为 100 升。以同样的方式，一辆汽车有四个轮子，马力 200，煤气罐容量为 25 升。

在 Ruby 中定义类

为了使用 Ruby 实现面向对象编程，您需要先学习如何在 Ruby 中创建对象和类。

在 Ruby 中，类总是以关键字 *class* 开始，后跟类的名称。类名的首字母应该大写。类 *Customer* 如下所示：

```
class Customer
end
```

您可以使用关键字 *end* 终止一个类。*类* 中的所有数据成员都是介于类定义和 *end* 关键字之间。

Ruby 类中的变量

Ruby 提供了四种类型的变量：

- **局部变量：**局部变量是在方法中定义的变量。局部变量在方法外是不可用的。在后续的章节中，您将看到有关方法的更多细节。局部变量以小写字母或 _ 开始。
- **实例变量：**实例变量可以跨任何特定的实例或对象中的方法使用。这意味着，实例变量可以从对象到对象的改变。实例变量在变量名之前放置符号（@）。
- **类变量：**类变量可以跨不同的对象使用。类变量属于类，且是类的一个属性。类变量在变量名之前放置符号（@@）。
- **全局变量：**类变量不能跨类使用。如果您想要有一个可以跨类使用的变量，您需要定义全局变量。全局变量总是以美元符号（$）开始。

实例

使用类变量 @@no_of_customers，您可以判断被创建的对象数量，这样可以确定客户数量。

实例

```
class Customer
@@no_of_customers=0
end
```

在 Ruby 中使用 *new* 方法创建对象

对象是类的实例。现在您将学习如何在 Ruby 中创建类的对象。在 Ruby 中，您可以使用类的方法 *new* 创建对象。

方法 *new* 是一种独特的方法，在 Ruby 库中预定义。new 方法属于*类*方法。

下面的实例创建了类 Customer 的两个对象 cust1 和 cust2：

```
cust1 = Customer. new
cust2 = Customer. new
```

在这里，cust1 和 cust2 是两个对象的名称。对象名称后跟着等号（=），等号后跟着类名，然后是点运算符和关键字 *new*。

自定义方法来创建 Ruby 对象

您可以给方法 *new* 传递参数，这些参数可用于初始化类变量。

当您想要声明带参数的 *new* 方法时，您需要在创建类的同时声明方法 *initialize*。

initialize 方法是一种特殊类型的方法，将在调用带参数的类的 *new* 方法时执行。

下面的实例创建了 initialize 方法：

实例

```
class Customer
@@no_of_customers=0
```

```
def initialize(id, name, addr)
@cust_id=id
@cust_name=name
@cust_addr=addr
end
end
```

在本实例中，您可以声明带有 **id、name、addr** 作为局部变量的 *initialize*方法。在这里，*def* 和 *end* 用于定义 Ruby 方法 *initialize*。在后续的章节中，您将学习有关方法的更多细节。

在 *initialize* 方法中，把这些局部变量的值传给实例变量 @cust_id、@cust_name 和 @cust_addr。在这里，局部变量的值是随着 new 方法进行传递的。

现在，您可以创建对象，如下所示：

```
cust1=Customer.new("1", "John", "Wisdom Apartments, Ludhiya")
cust2=Customer.new("2", "Poul", "New Empire road, Khandala")
```

Ruby 类中的成员函数

在 Ruby 中，函数被称为方法。*类*中的每个方法是以关键字 *def* 开始，后跟方法名。

方法名总是以**小写字母**开头。在 Ruby 中，您可以使用关键字 *end* 来结束一个方法。

下面的实例定义了一个 Ruby 方法：

```
class Sample
def function
statement 1
statement 2
end
end
```

在这里，*statement 1* 和 *statement 2* 是类 Sample 内的方法 *function* 的主体的组成部分。这些语句可以是任何有效的 Ruby 语句。例如，我们可以使用方法 *puts* 来输出 *Hello Ruby*，如下所示：

```
class Sample
def hello
puts "Hello Ruby!"
end
end
```

下面的实例将创建类 Sample 的一个对象，并调用 *hello* 方法：

```
#!/usr/bin/ruby
class Sample
def hello
puts "Hello Ruby!"
end
end
# 使用上面的类来创建对象
object = Sample. new
object.hello
```

这将会产生下面的结果：

```
Hello Ruby!
```

简单的案例研究

如果您想要做更多有关类和对象的练习，这里有一个案例研究：

Ruby 类案例

← Ruby 语法 Ruby 类案例 →