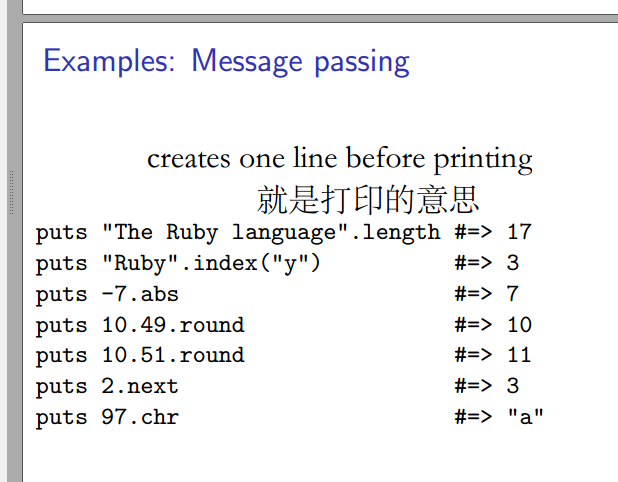
imperative programming:命令式编程 用改变程序状态的语句来描述计算

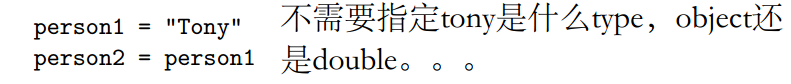
pure object-oriented programming，语言所有的数据类型都是object，并且object的所有操作都能通过给object发送一个message激活

给receiver object发送message会激活它的一个method。 一个message 包括method的名字与对应参数



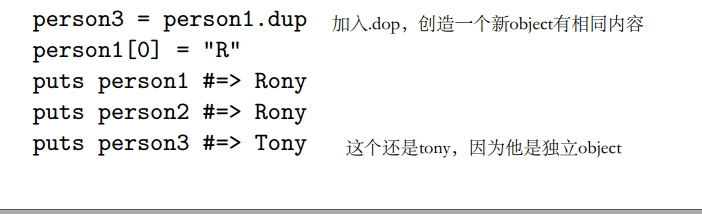
VARIABLE AND aliasing

多个变量指代同一个object叫做aliasing 混叠，



person2=person1这个赋值并不会创造一个object，他只是让person2指向person1

为了避免aliasing，我们要使用dup，会创造一个新object，但是是相同的内容

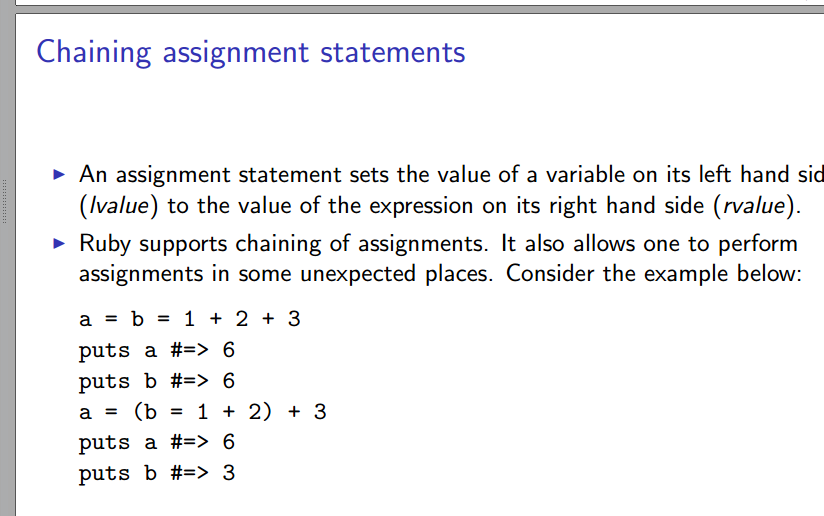


chaining assignment

链式相等

a=b=c这种

变量名字的值写在左边，变量具体的值写在右边

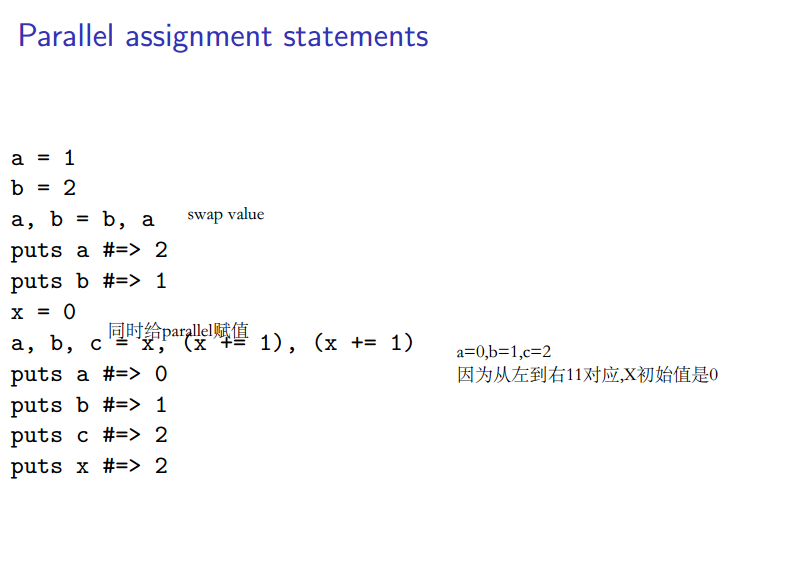


注意下面这个，括号代表着B，3+3=6

会把a,b值交换

a,b,c=1,2,3就是平行相等

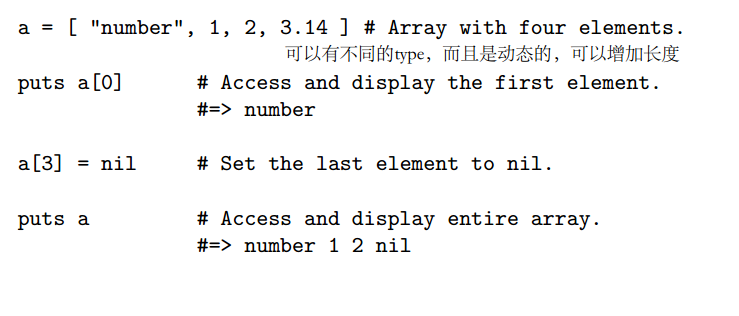
根据逗号相对位置，variable=等号右边对应的值



array

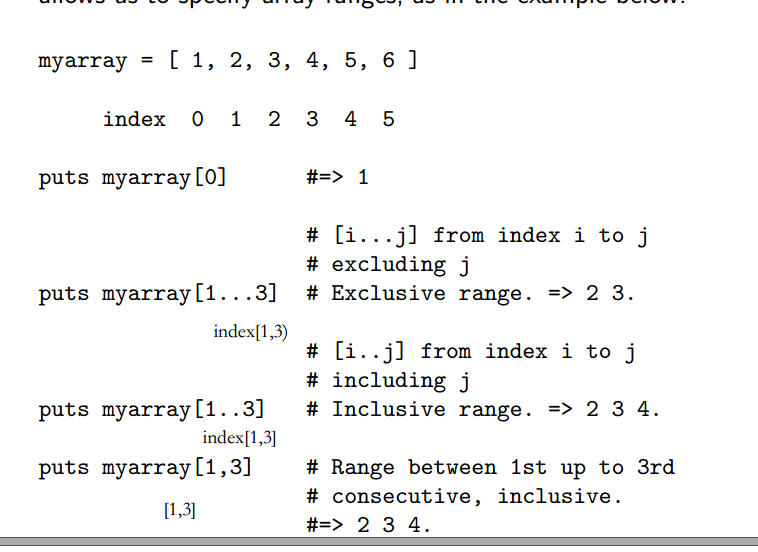
array是一组element的有序集合，每个元素都有integer index标识

最基本的array建立法（默认法）



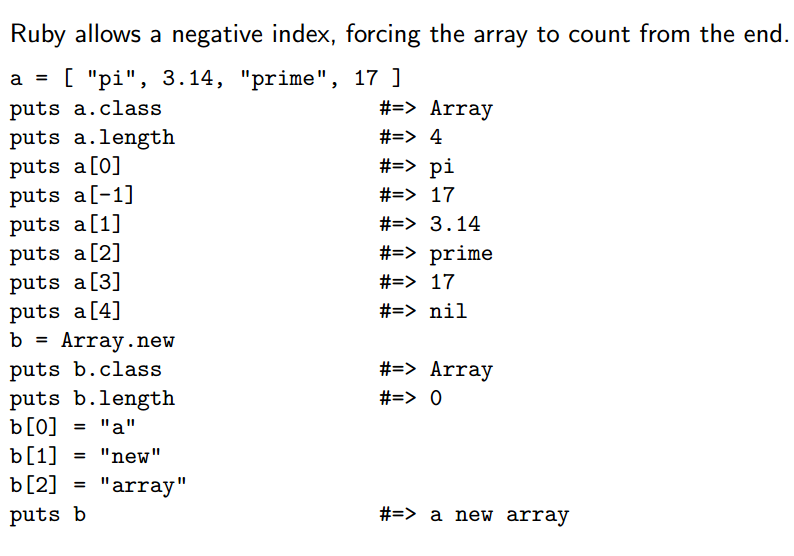
所有object被放在方括号里，这些object是可以有不同type的，位置index就是默认整数

我们也可以通过切割一个原有array来创造一个新array



三个点不包括右边，两个点包括，逗号包括

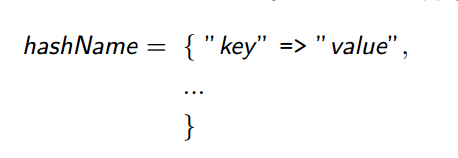
array也接受负数index，会直接转到末尾



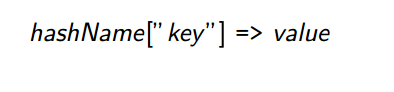
associative array（hash）相连阵列

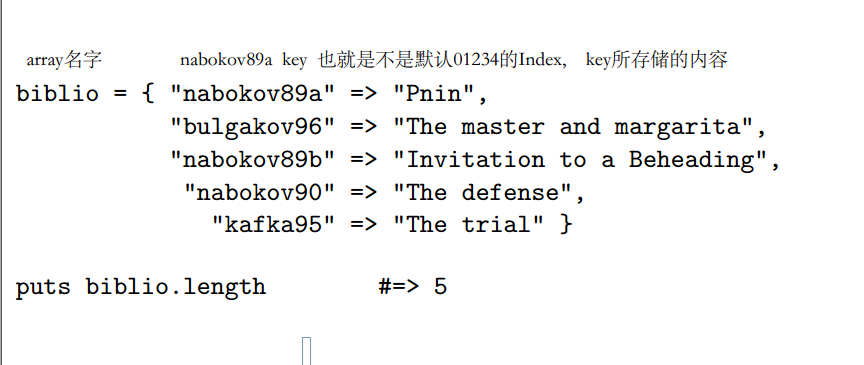
一组没有顺序的element（普通·array有顺序，因为有整数index）

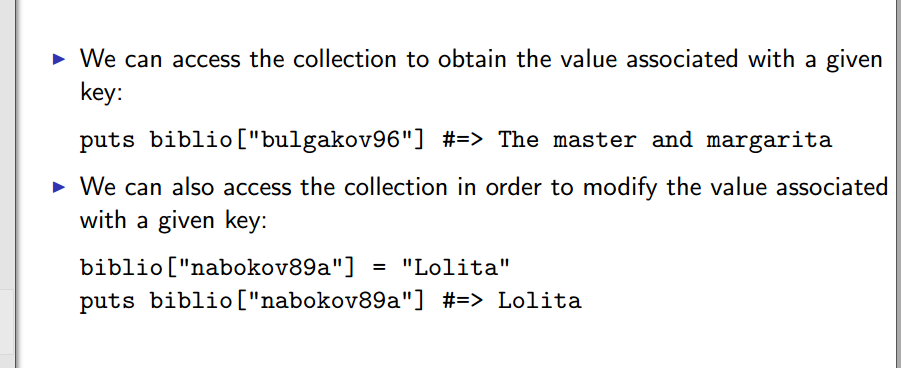
一个element是一对object组成，一个value object和一个key object，通过key我们可以来索引value

注意用的是花括号

在建立完成以后，我们可以通过以下格式利用该key来索引value

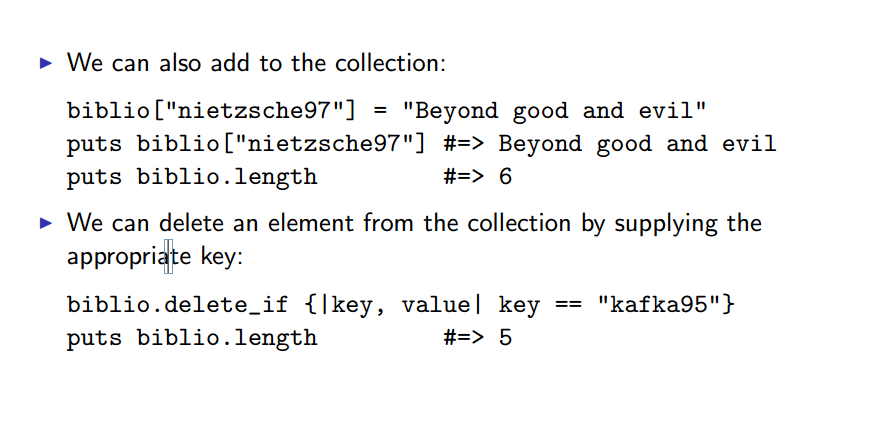




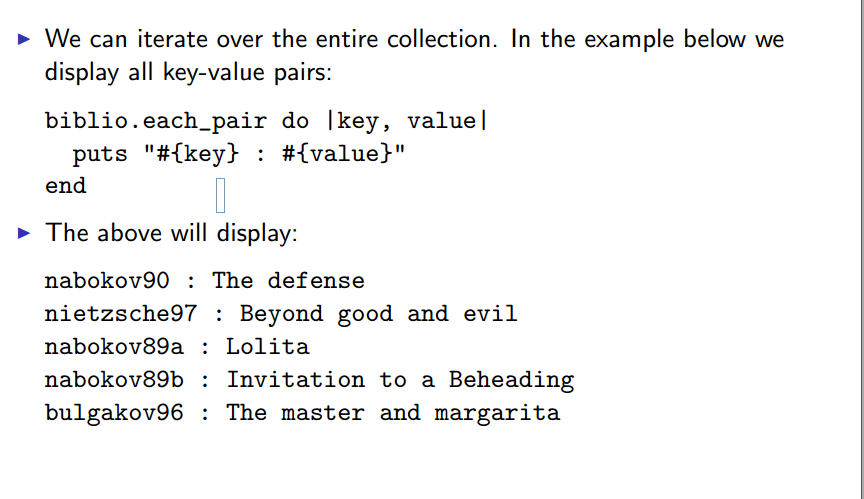


我们也可以通过key来改变value

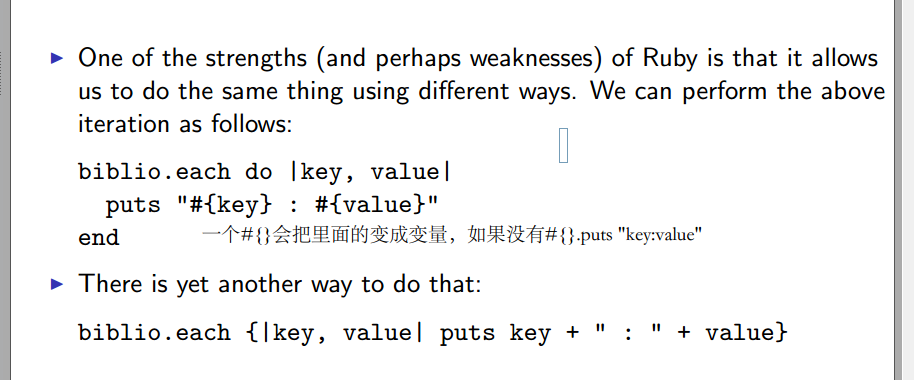
我们也可以在这组element中加入新的element或删除旧的element



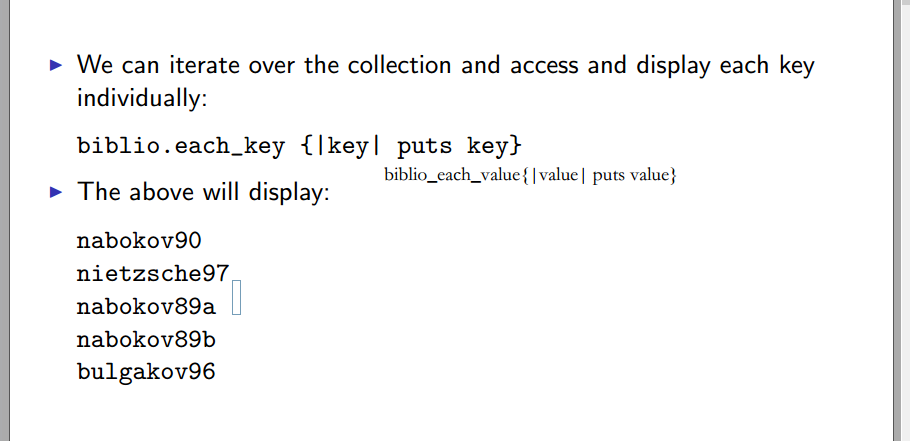
我们可以迭代整个array 的key-value pairs



其他的表示方法



我们也可以迭代并分别print key



variable命名方法，

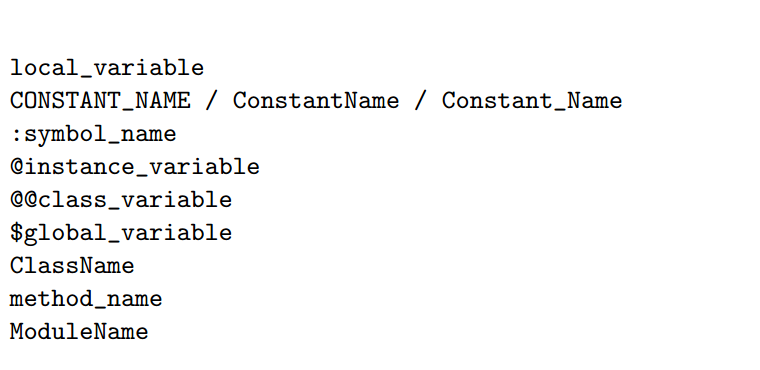
Ruby通过第一个char来确定name的饿类别,

class 变量前面有两个@@

当地变量，method参数，method name，必须是小写字母开头或者\_短下划线开头

global variable 有一个$前缀

实例变量 有一个@前缀



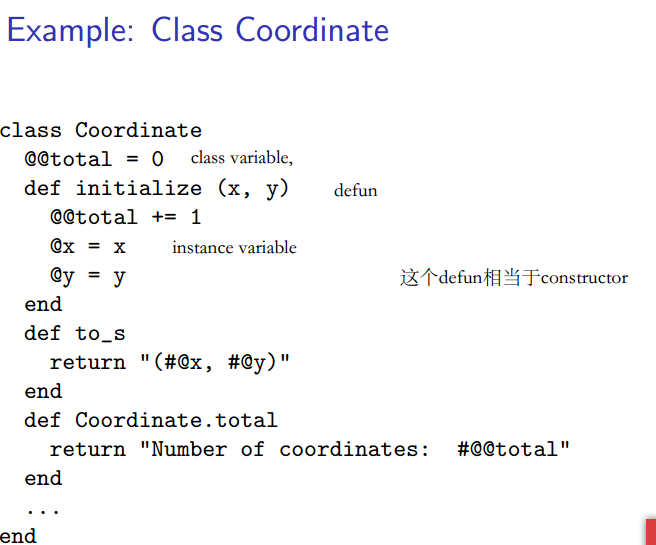
Objects

class中的实例（objects）包括state （实例变量）和behavior （method）

每一个object包括他自己独立的state

而behavvior则是在不同的objects中share的

object的state由一系列参数组成，还有他们的当前值



注意class Coordinate本身是不需要前缀的，只要是大写字母就行，但是class 变量total需要@@

里面的method 名字不需要前缀，小写字母就行

@x实例变量，指这个OBJECT特有的X值，比如说objecct1的X是1，OBJECFT2的X是2

class这个keyword用来描述一个class

initialize这个方法实际上构筑了他真正的数据结构，就如同JAVAmethod外面那一层，每一个class必须有这个initialize method

@x@y就是instance variable

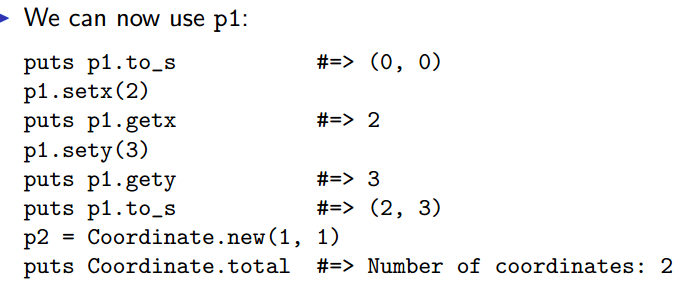
put与print都是打出，区别在于Put会空行，print不会空行

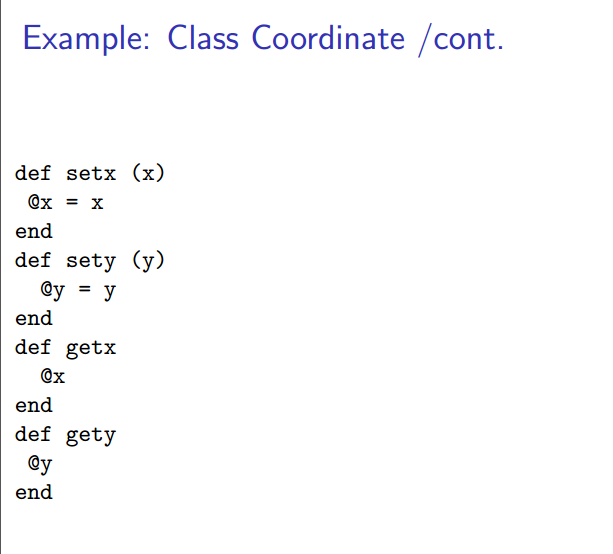
我们实例化一个class的方法是使用new



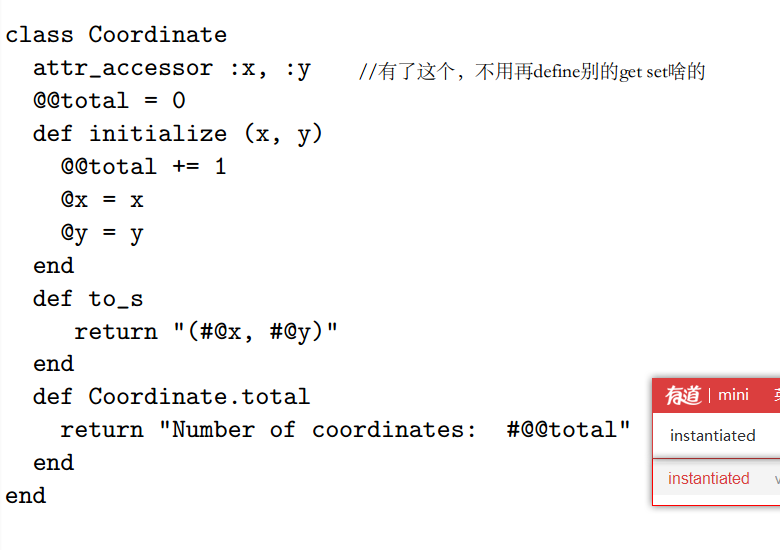
会创造一个instance p1 他的coordinate是（0,0）

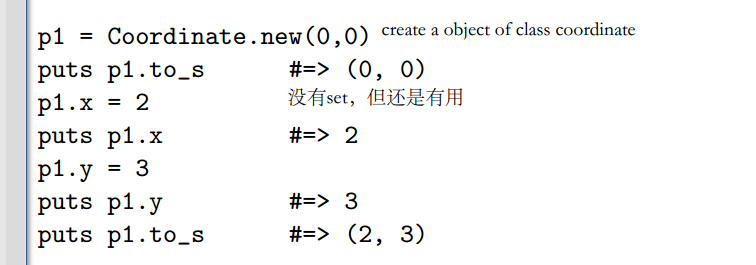
我们现在就可以使用p1了



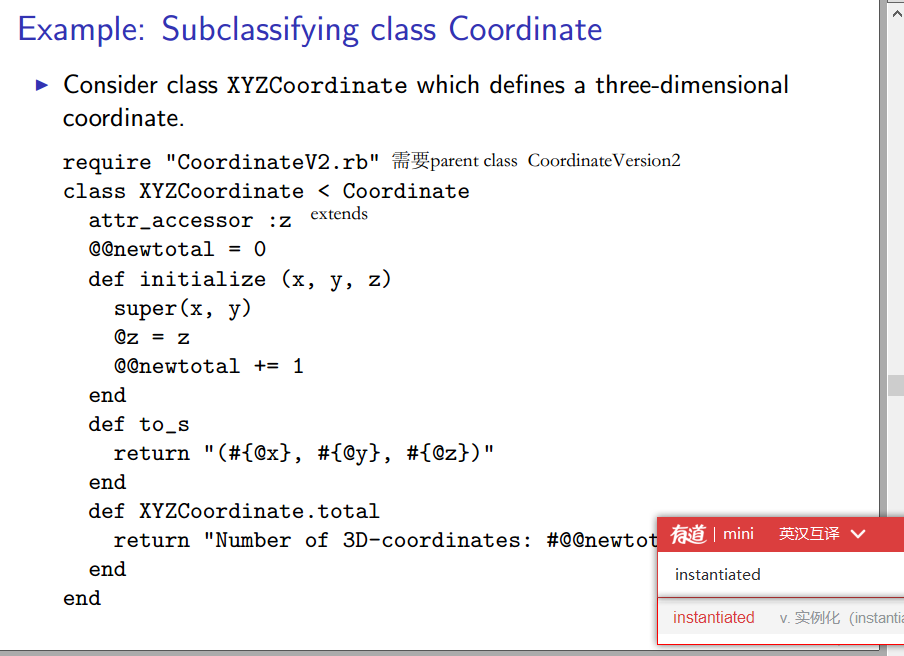


attr\_accessor:有了这个就不用设置getx setx这种



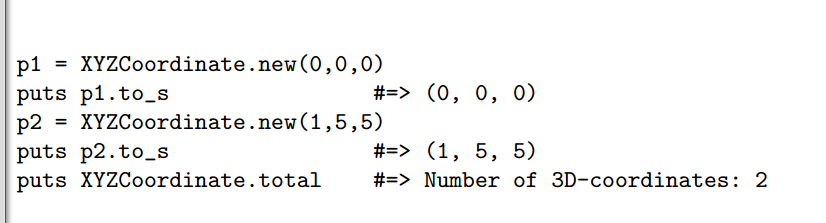


一个带继承的class：XYZ坐标系



小于号就是extend的意思，初始accessor只需要描述新的，

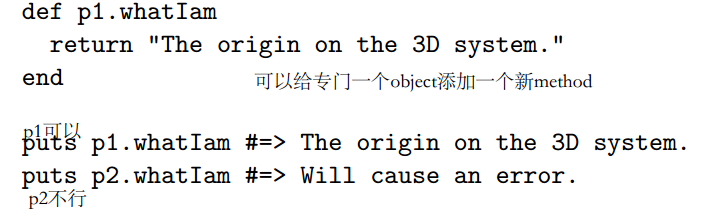
原来的只要super



objects extension

ruby允许我们对单独instance加上新behavior

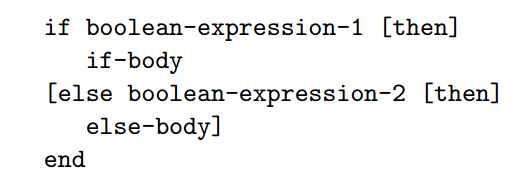
使用def end格式

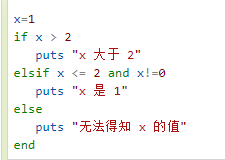


Ruby的·control flow

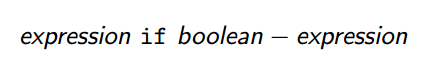
ruby的control flow支持selection和repetition，选择或重复

最简单的if else





还有一句话的形式



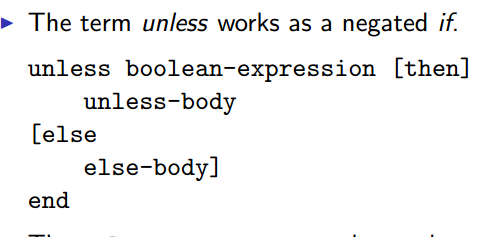
如果boolean expression 是true，进行expression

稍微复杂版本



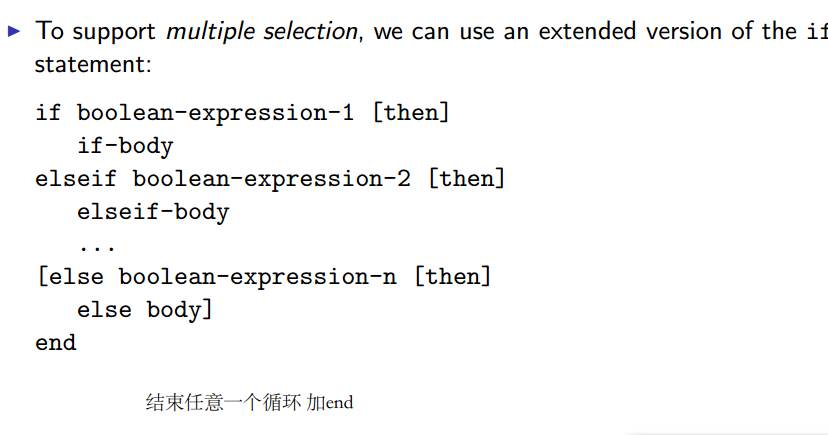
如果boolean-expression是true，进行expression1,不然expression2

ruby还有一个否定意味的if:unless

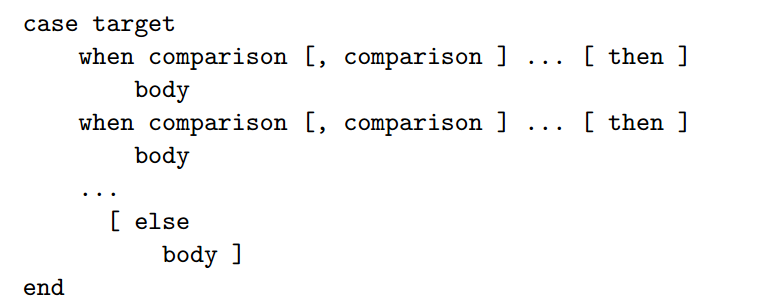


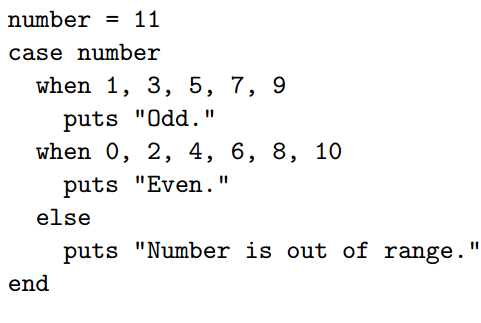


多重选择，elseif

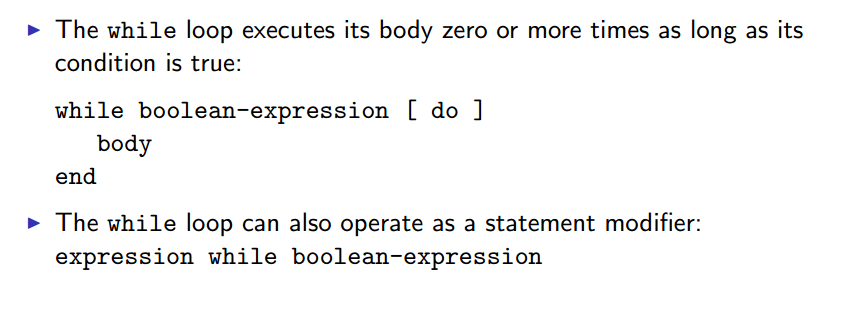


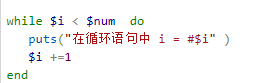
case，相当于switch



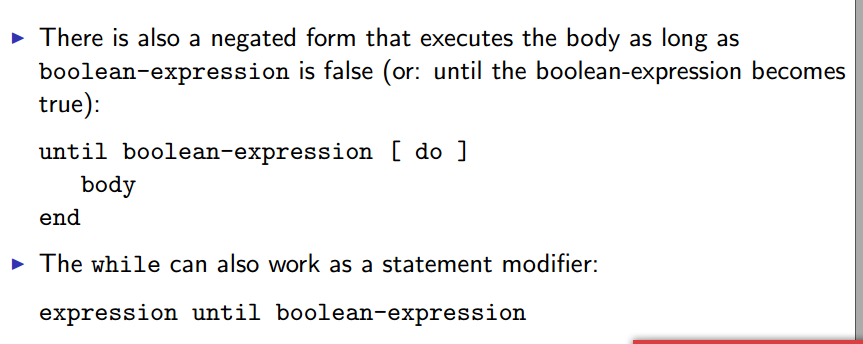


while

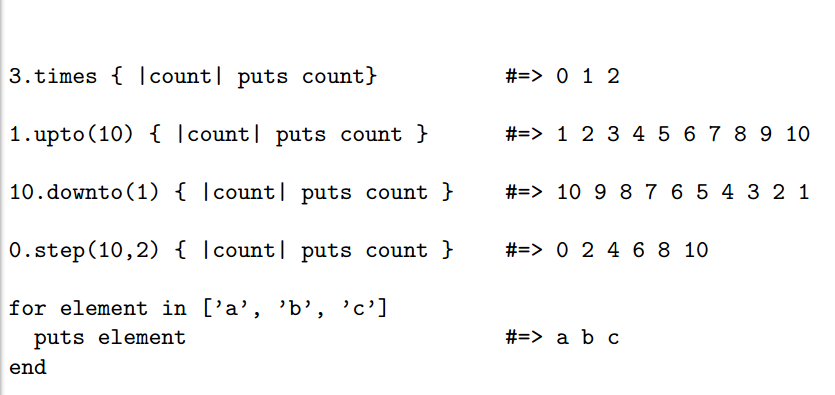




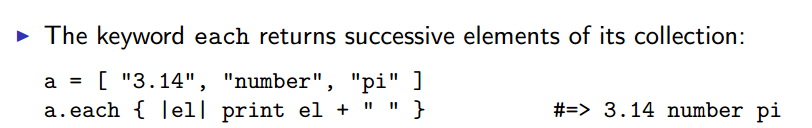
Until，while的否定形式



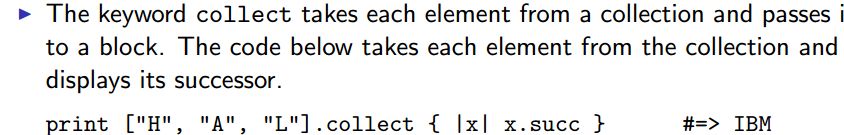
迭代loop



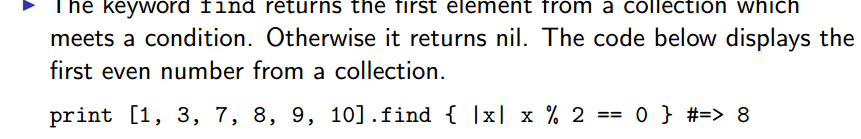
keyword :each 会return所有element



keyword:collect ，回收集集合里的每一个element并且把他传到一个block里，以下代码展示了从collection里收集每一个元素并且展示这些元素的successor



keywordLfind， find只会return第一个满足condition的元素，不然就是return nil

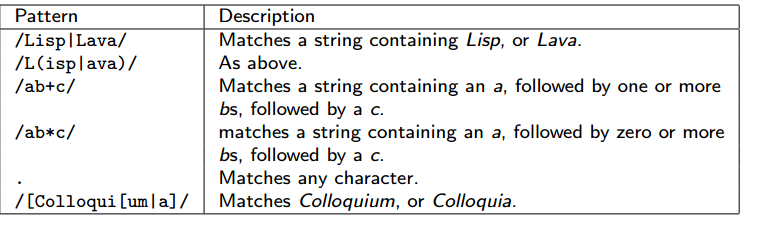


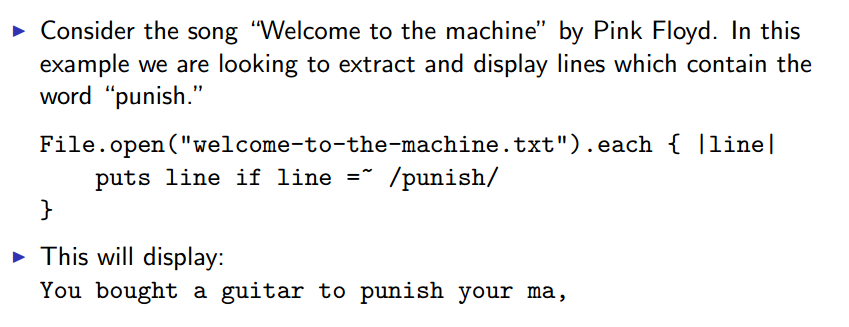
regular expression，正则表达式

正则表达式是一种在string中找到我们所需的一组char的手段

在ruby中，我们使用/PATTERN/，

在ruby中正则表达式是object，所以他们也可以有对应操作



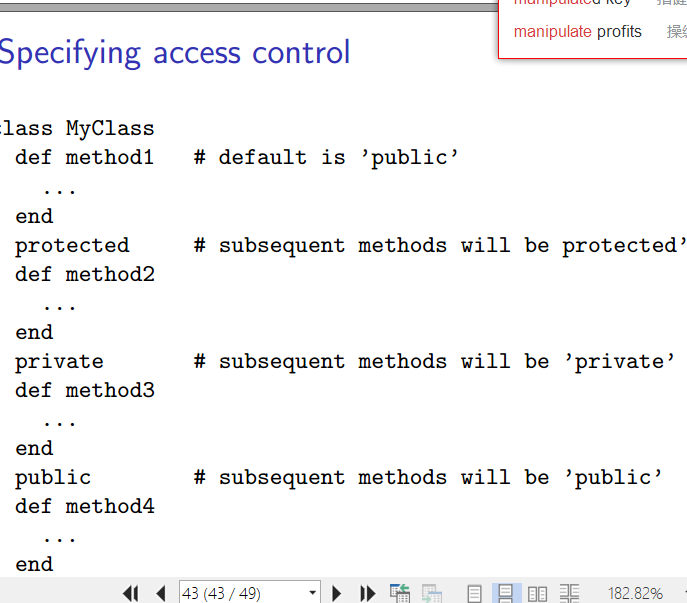


file open,队以这一串string每一行 找到有Punish的Line，最后会display这句话

public method每个人都可以call， method是默认Public的，除了initialize(他永远是private的)

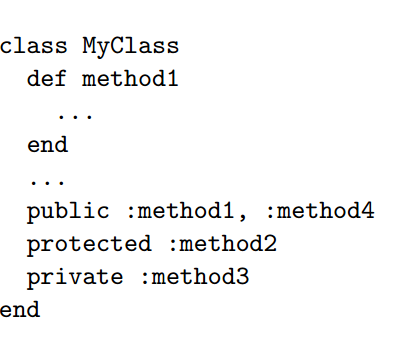
protedcted method只能由他这个class的object或他的subclass子集的object激活

private method只能通过class来激活



Method1是Public，method2是protected,method3是private，method4是public

或者还有一种统一设置属性的方法



Introspection

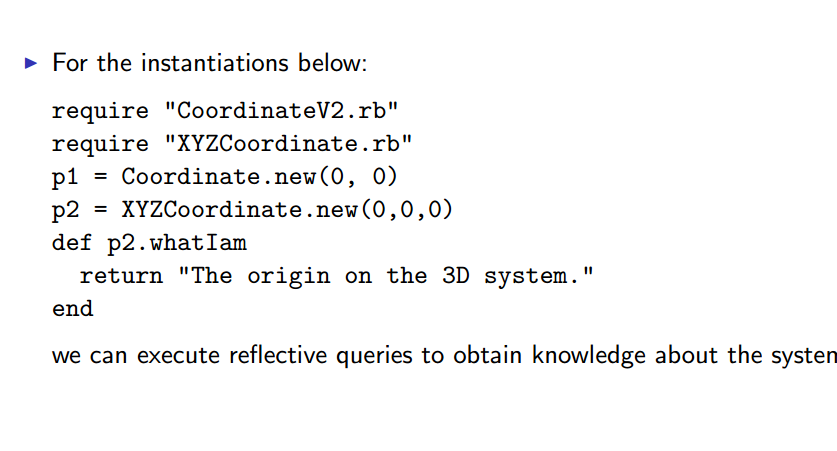
Introspection 是计算系统参考自身结构的能力（但是无法修改）

在ruby中，我们可以得到program的以下信息

他有什么objects

objects的contents和behaviors

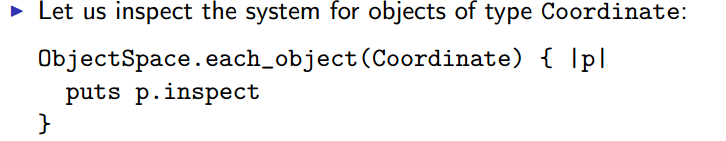
当前class hierarchy 层级

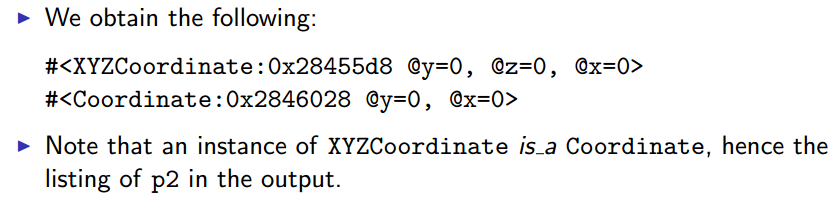


我们可以查询系统自身信息

1.他有什么objects?

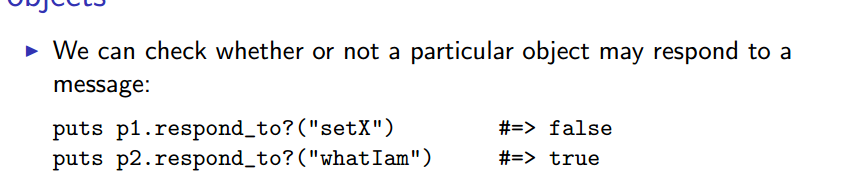
迭代法，



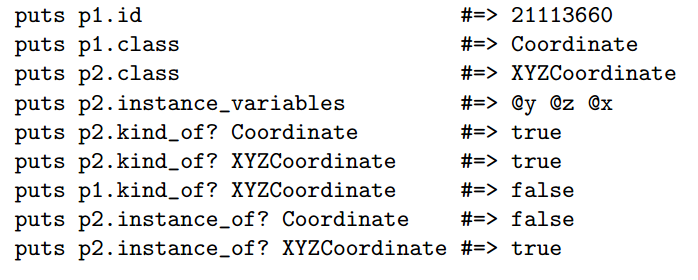


2.每一个object所拥有的content和behavior

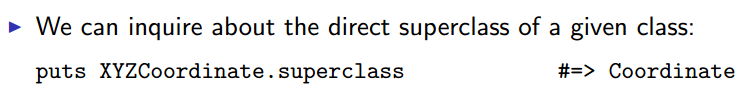
来检查是不是有对应method



我们也可以检查他们的独特ID与他们的class，并且测试他们与class的关系



如何得到一个class的superclass



如何得到class的特殊信息

