# day 1 语法

## 注释

1. #
2. =begin xxxxxxxxxxxxxxxx =end (行首)
3. 代码中的注释
   1. <http://192.168.75.10/cn/index.php?cmd=read&page=%E6%8A%80%E6%9C%AF%2F%E5%BF%85%E9%A1%BB%E5%AD%A6%E7%9F%A5%E8%AF%86%E7%82%B9%2FRuby%2F%E7%BC%96%E7%A0%81%E8%A7%84%E7%BA%A6%2F%E4%BB%A3%E7%A0%81%E6%B3%A8%E9%87%8A&word=%E6%B3%A8%E9%87%8A>
4. 追求代码的美观，美感。

## 分行

1. Ruby 中用分号“；”来表示一个语句的结束。一行如果有多个语句，每个语句用分号隔开，而最后一个语句可以省略分号。换行符表示一行结束。如果语句太长，可以用“ \ ”连接下一行。
2. 单独的一句可以省略分号

## 关键字

模块定义：module

类定义：class

方法定义：def

检查类型：defined?

条件语句：if ， then ， else ，elsif，case ，when ，unless

循环语句：for ，in ，while ，until ，next， break ，do ，

redo ，retry ，yield

逻辑判断：not ，and ，or

逻辑值和空值：true ，false ， nil

异常处理：rescue ，ensure

对象引用：super ，self

块的起始：begin/end

文件相关：\_\_FILE\_\_ ， \_\_LINE\_\_

方法返回：return

别名：alias

## 运算符

。。。。。。

## 标识名和变量的作用域

Ruby 的标识名用来指向常量，变量，方法，类和模块。标识名的首字符用来

帮助我们确定标识所指向内容的作用域。一些标识名，就是上面所示的关键字，不

能用来当作常量，变量，方法，类或模块的名字。

Ruby的标识名区分大小写。

Ruby 使用一个约定来帮助它区别一个名字的用法：

名字前面的第一个字符表明这个名字的用法。

局部变量、方法参数和方法名称应该用一个小写字母开头或者用一个下划线开头；

全局变量用美元符作为前缀 $；

而实例变量用 @ 开头；

类变量用@@开头；

类名、模块名和常量应该用大写字母开头。

词首字母后面可以是字母、数字和下划线的任意组合； @ 后面不可以直接跟数字。

## 类库

前面我们使用了 puts 和 print ，你一定注意到它们并不是关键字，为什么能够直接使用？

I/O就是输入/输出，这是Ruby语言Kernel模块的方法，Mix-in 在根类Object 中的。

puts 把它的所有参数写出来，每一个参数结束都加入一个换行符；

print 也写出它的参数，不过没有换行；

还有一个常用的输出方法是printf，它按格式输出参数。

printf "Number: %4.3f, String: %s", 7.8, "hi!" >>>Number: 7.800, String: hi!

介绍了三个输出方法，再介绍一个输入方法：gets ，它从你的程序的标准输入流中返回一行。一般用来从键盘或文件读入一行数据。

## 数据类型

字符串，数组，哈希表，区间，正则表达式Regexp

1. 数字分为整数型、浮点型。
2. 字符串是在‘’（单引号）、“”（双引号）之间的代码。
3. 数组的下标从0开始。Ruby的数组和其它语言不同，数组的每个元素可以是不同的类型。

例：[ 2.4， 99，“thank you”，[ a, b ,c ] ，78 ]。

1. 区间：1..5 表示1，2，3，4，5 ；

1...5表示1，2，3，4 。

## 赋值和条件运算符

1. 赋值符号： =
2. 条件运算符：

== 比较两个对象的值是否相等 ,返回 true, flase；

!=比较两个对象的值是否不相等 ,返回 true, flase；

eql?比较两个对象的值、类型是否相等,返回 true, flase；

a=1; b=1.0; a.eql?(b) #flase（a为整数型，b为浮点型）；

equal?比较两个对象在内存中地址是否相同,返回 true, flase；

a=1.0; b=1.0; a.equal?(b)#flase

a=1.0; b=a ; a.equal?(b) # true

<=>比较两个对象的大小，大于、等于、小于分别返回1,0,-1

"aab" <=> "acb" # -1 （第二个 a 的 ASCII 码小于 c）

===右边的对象是否在左边区间之内,返回true, flase

puts (0..9)=== 3.14 #true

=~用来比较是否符合一个正则表达式,返回模式在字符串中被匹配到的

!~断言不符合一个正则表达式,返回 true, flase

<= <>>=小于等于小于大于大于等于

## 条件判断语句

1. if(单行)
   1. if 条件① then 语句1; 语句2 ; 语句… end puts “1真的大于2” if 1 > 2
   2. （语句1; 语句2 ; 语句…） if 条件
2. if(多行)

if 条件

语句1; 语句2 ; 语句…

elsif条件

语句1; 语句2 ; 语句…

else

语句1; 语句2 ; 语句…

End

1. unless

unless 条件 = if not (条件)

1. case

case 对象

when 可能性1

语句1….

语句2….

when 可能性1

语句1….

语句2….

else

语句1….

语句2….

End

## 循环语句

1. while

while 条件

语句1；语句2；……

end

1. until

until 条件

语句1；语句2；……

End

1. for…in

for 变量 in 对象 # 对象可以是数组，区间，集合…,

语句1；语句2；……

end

1. break, next, redo, retry

在循环体内，如果遇到：

break ，跳出当层循环；

next ，忽略本次循环的剩余部分，开始下一次的循环；

redo ，重新开始循环，还是从这一次开始；

retry ，重头开始这个循环体。

1. times , upto , downto , each ,step

## 异常及线程

begin

rescue

ensure

end

与 Java 中的 try…catch…finally…throw 相对应 , Ruby 中用begin/end …rescue…ensure … raise 来处理异常，retry可以用在 rescue 中。可以只用 rescue 或是 ensure，两者都使用时，rescue 必须在 ensure前。

作业

1. 求50以内的素数。
2. 用冒泡法对数组[9,2,7,5,1,4,3,6]进行升序排序，然后依次输出所有的值，以及该值现在的坐标和原来的坐标