

Ruby 判断

Ruby 提供了几种很常见的条件结构。在这里，我们将解释所有的条件语句和 Ruby 中可用的修饰符。

Ruby *if...else* 语句

语法

```
if conditional [then]
  code...
[elsif conditional [then]
  code...]...
[else
  code...]
end
```

if 表达式用于条件执行。值 *false* 和 *nil* 为假，其他值都为真。请注意，Ruby 使用 *elsif*，不是使用 *else if* 和 *elif*。

如果 *conditional* 为真，则执行 *code*。如果 *conditional* 不为真，则执行 *else* 子句中指定的 *code*。

通常我们省略保留字 *then*。若想在一行内写出完整的 *if* 式，则必须以 *then* 隔开条件式和程式区块。如下所示：

```
if a == 4 then a = 7 end
```

实例

```
#!/usr/bin/ruby
# -*- coding: UTF-8 -*-
x=1
if x > 2
  puts "x 大于 2"
elsif x <= 2 and x!=0
  puts "x 是 1"
else
  puts "无法得知 x 的值"
end
```

[尝试一下 »](#)

以上实例输出结果：

```
x 是 1
```

Ruby *if* 修饰符

语法

```
code if condition
```

if修饰词组表示当 if 右边之条件成立时才执行 if 左边的式子。即如果 *conditional* 为真，则执行 *code*。

实例

```
#!/usr/bin/ruby
$debug=1
print "debug\n" if $debug
```

尝试一下 »

以上实例输出结果：

```
debug
```

Ruby *unless* 语句

语法

```
unless conditional [then]
code
[else
code ]
end
```

unless式和 if式作用相反，即如果 *conditional* 为假，则执行 *code*。如果 *conditional* 为真，则执行 else 子句中指定的 *code*。

实例

```
#!/usr/bin/ruby
# -*- coding: UTF-8 -*-
x=1
unless x>2
puts "x 小于 2"
else
puts "x 大于 2"
end
```

尝试一下 »

以上实例输出结果为：

```
x 小于 2
```

Ruby *unless* 修饰符

语法

```
code unless conditional
```

如果 *conditional* 为假，则执行 *code*。

实例

```
#!/usr/bin/ruby
# -*- coding: UTF-8 -*-
$var = 1
print "1 -- 这一行输出\n" if $var
print "2 -- 这一行不输出\n" unless $var
$var = false
print "3 -- 这一行输出\n" unless $var
```

[尝试一下 »](#)

以上实例输出结果：

```
1 -- 这一行输出
3 -- 这一行输出
```

Ruby case 语句

语法

```
case expression
[when expression [, expression ...] [then]
code ]...
[else
code ]
end
```

case先对一个 *expression* 进行匹配判断，然后根据匹配结果进行分支选择。

它使用 `===` 运算符比较 **when** 指定的 *expression*，若一致的话就执行 **when** 部分的内容。

通常我们省略保留字 `then`。若想在一行内写出完整的 `when` 式，则必须以 `then` 隔开条件式和程式区块。如下所示：

```
when a == 4 then a = 7 end
```

因此：

```
case expr0
when expr1, expr2
stmt1
when expr3, expr4
stmt2
else
stmt3
end
```

基本上类似于：

```
_tmp = expr0
if expr1 === _tmp || expr2 === _tmp
stmt1
```

```
elsif expr3 === _tmp || expr4 === _tmp
  stmt2
else
  stmt3
end
```

实例

```
#!/usr/bin/ruby
# -*- coding: UTF-8 -*-
$age = 5
case $age
when 0 .. 2
  puts "婴儿"
when 3 .. 6
  puts "小孩"
when 7 .. 12
  puts "child"
when 13 .. 18
  puts "少年"
else
  puts "其他年龄段的"
end
```

[尝试一下 »](#)

以上实例输出结果为：

小孩

当case的"表达式"部分被省略时，将计算第一个when条件部分为真的表达式。

```
foo = false
bar = true
quu = false
case
when foo then puts 'foo is true'
when bar then puts 'bar is true'
when quu then puts 'quu is true'
end
# 显示 "bar is true"
```

[← Ruby 注释](#)[Ruby 循环 →](#)[点我分享笔记](#)

