

MATZ ON RUBY 1.9

R U B Y C O N F 2 0 0 7
C H A R L O T T E , N C

These notes were recreated from memory by
Eric at LearnRuby.com



LearnRuby.com

RUBY 1.9

- ✻ Released by Christmas 2007
- ✻ Unfortunately, won't be fully stable when released
 - ✻ Considered transitional to 2.0
 - ✻ 1.8 is still the “go to release” for stability
- ✻ Will break some code
 - ✻ A few major changes
 - ✻ Many minor changes
- ✻ Some changes have been “backed out” because they wouldn't be ready for the Christmas release

MULTILINGUALIZATION (“M17N”)

USE OF MAGIC COMMENTS

```
# -*- coding: encoding_name -*-
```

PARAMETER TO OPEN

```
open("filename", "r:encoding_name")
```

VALUES FOR *encoding_name*

utf-8

utf-16

ascii-8bit

euc-jp

and so forth...

STRING INDEXING RETURNS STRINGS

CODE EXAMPLE

```
s = "Hello World!"
```

```
c = s[4] # equals "o"
```

EXPLANATION

Prior to 1.9 indexing into a String returned the ASCII code of the character at that position.

SCOPE OF BLOCK ARGUMENTS LIMITED TO BLOCK

CODE EXAMPLE

```
x = 15  
  
(1..5).each do |x|  
  puts x  
end  
  
puts x # outputs 15
```

OPTIONAL ARGUMENTS IN ANY POSITION

CODE EXAMPLE

```
def m1(a, b = 7, c, d = 8)
  # code
end
```

MEANING

m1(10, 20) ➤

a = 10

b = 7

c = 20

d = 8

m1(10, 20, 30) ➤

a = 10

b = 20

c = 30

d = 8

m1(10, 20, 30, 40) ➤

a = 10

b = 20

c = 30

d = 40

EXTERNAL ITERATORS

CODE EXAMPLE

```
i1 = [2, 12, 13, 22].each
i2 = [5, 10, 15].each
loop {
  puts i1.next + i2.next
}
```

OUTPUT

7

22

28

EXPLANATION

Fourth call to `i2.next` raises some sort of “iterator completed” exception, which is caught by the *loop* construct and causes it to exit. The exception can be caught by your own code within the loop.

STRING ITERATORS

CODE EXAMPLE

```
s = "ab\nxy"  
s.each.char { |c| puts c }  
puts "----"  
s.each.line { |l| puts l }  
puts "----"  
s.each.byte { |b| puts b }
```

OUTPUT

a

b

x

y

ab

xy

97

98

10

120

121

EXPLANATION

Previously, each allowed one to iterate through a String line by line. But Strings can be used as collections of chars, lines, or bytes. So now all will be available and lines will not be "privileged".

OPERATING SYSTEM THREADS

- ✻ Transition from 1.8's green threads to OS threads
- ✻ There will be a "Giant Lock", so only one thread can execute at a time
- ✻ This will be a transition phase on the way to a more flexible threading model in 2.0

“BACKED OUT” CHANGES

- ✻ Some changes planned for Ruby 1.9 had to be backed out due to time limitations
- ✻ A namespace mechanism for “monkey patching”, so modifications to a core class (e.g., String) could be scoped
- ✻ Continuations

ERRORS? THANKS!

I re-created these notes from memory, so I may have made some mistakes. Please report any mistakes you find through the [LearnRuby.com "Contact Us" page](#).

Thanks go to Brandon Dimcheff, Patrick Hurley, and Winston Tsang for helping me remember what was said in Matz's keynote!

Eric