

Just Enough Ruby for Chef

A quick & dirty crash course

v2.1.6

irb - the interactive ruby shell

- irb is the interactive ruby shell
- It is a REPL for Ruby
 - Read-Eval-Print-Loop
 - LISP, Python, Erlang, Clojure, etc.
- An interactive programming environment
- Super-handdy for trying things out

Exercise: Start irb on your 'target' node

```
irb(main):001:0>
```

Variable assignment

```
irb(main):001:0> x = "hello"  
=> "hello"
```

```
irb(main):002:0> puts x  
hello  
=> nil
```

Arithmetic

```
irb(main):003:0> 1 + 2  
=> 3
```

Arithmetic

```
irb(main):004:0> 18 - 5  
=> 13
```

Arithmetic

```
irb(main):005:0> 2 * 7  
=> 14
```


Arithmetic

```
irb(main):006:0> 5 / 2  
=> 2
```

Arithmetic

```
irb(main):007:0> 5 / 2.0  
=> 2.5
```

Arithmetic

```
irb(main):008:0> 5.class  
=> Fixnum
```

```
irb(main):009:0> 5.0.class  
=> Float
```

Arithmetic

```
irb(main):010:0> 1 + (2 * 3)  
=> 7
```

Strings

```
irb(main):011:0> 'jungle'  
=> "jungle"
```

Strings

```
irb(main):012:0> 'it\'s  
alive'  
=> "it's alive"
```

Strings

```
irb(main):013:0> "animal"  
=> "animal"
```

Strings

```
irb(main):014:0> "\"so easy\""  
=> "\"so easy\""
```

```
irb(main):015:0> puts "\"so  
easy\""  
"so easy"  
=> nil
```


Strings

```
irb(main):016:0> x = "pretty"  
=> "pretty"
```

```
irb(main):017:0> "#{x} nice"  
=> "pretty nice"
```

```
irb(main):018:0> '#{x} nice'  
=> "\#{x} nice"
```

Truthiness

```
irb(main):019:0> true  
=> true
```

```
irb(main):020:0> false  
=> false
```

```
irb(main):021:0> nil  
=> nil
```

```
irb(main):022:0> !!nil  
=> false
```

```
irb(main):023:0> !!0  
=> true
```

```
irb(main):024:0> !!x  
=> true
```

Operators

```
irb(main):025:0> 1 == 1  
=> true
```

```
irb(main):026:0> 1 == true  
=> false
```

```
irb(main):027:0> 1 != true  
=> true
```

```
irb(main):028:0> !!1 == true  
=> true
```

Operators

```
irb(main):029:0> 2 < 1  
=> false
```

```
irb(main):030:0> 2 > 1  
=> true
```

```
irb(main):031:0> 4 >= 3  
=> true
```

```
irb(main):032:0> 4 >= 4  
=> true
```

```
irb(main):033:0> 4 <= 5  
=> true
```

```
irb(main):034:0> 4 <= 3  
=> false
```

Operators

```
irb(main):035:0> 5 <=> 5  
=> 0
```

```
irb(main):036:0> 5 <=> 6  
=> -1
```

```
irb(main):037:0> 5 <=> 4  
=> 1
```

Arrays

```
irb(main):038:0> x = ["a", "b", "c"]  
=> ["a", "b", "c"]
```

```
irb(main):039:0> x[0]  
=> "a"
```

```
irb(main):040:0> x.first  
=> "a"
```

```
irb(main):041:0> x.last  
=> "c"
```

Arrays

```
irb(main):042:0> x + ["d"]  
=> ["a", "b", "c", "d"]
```

```
irb(main):043:0> x  
=> ["a", "b", "c"]
```

```
irb(main):044:0> x = x + ["d"]  
=> ["a", "b", "c", "d"]
```

```
irb(main):045:0> x  
=> ["a", "b", "c", "d"]
```

Arrays

```
irb(main):046:0> x << "e"  
=> ["a", "b", "c", "d", "e"]
```

```
irb(main):047:0> x  
=> ["a", "b", "c", "d", "e"]
```


Arrays

```
irb(main):048:0> x.map { |i| "the  
letter #{i}" }  
=> ["the letter a", "the letter b",  
"the letter c", "the letter d", "the  
letter e"]
```

```
irb(main):049:0> x  
=> ["a", "b", "c", "d", "e"]
```

Arrays

```
irb(main):050:0> x.map! { |i| "the letter  
#{i}" }  
=> ["the letter a", "the letter b", "the  
letter c", "the letter d", "the letter e"]
```

```
irb(main):051:0> x  
=> ["the letter a", "the letter b", "the  
letter c", "the letter d", "the letter e"]
```

Hashes

```
irb(main):052:0> h = {  
irb(main):053:1*  "first_name" =>  
"Gary",  
irb(main):054:1*  "last_name" =>  
"Gygax"  
irb(main):055:1> }  
=> {"first_name"=>"Gary",  
"last_name"=>"Gygax"}
```

Hashes

```
irb(main):056:0> h.keys  
=> ["first_name", "last_name"]
```

```
irb(main):057:0> h["first_name"]  
=> "Gary"
```

```
irb(main):058:0> h["age"] = 33  
=> 33
```

```
irb(main):059:0> h.keys  
=> ["first_name", "last_name", "age"]
```

Hashes

```
irb(main):060:0> h.values  
=> ["Gary", "Gygax", 33]
```

Hashes

```
irb(main):061:0> h.each { |k, v| puts  
  "#{k}: #{v}" }  
first_name: Gary  
last_name: Gygax  
age: 33  
=> {"first_name"=>"Gary",  
  "last_name"=>"Gygax", "age"=>33}
```

Regular Expressions

```
irb(main):062:0> x = "I want to believe"  
=> "I want to believe"
```

```
irb(main):063:0> x =~ /I/  
=> 0
```

```
irb(main):064:0> x =~ /lie/  
=> 12
```

```
irb(main):065:0> x =~ /smile/  
=> nil
```

```
irb(main):066:0> x !~ /smile/  
=> true
```

Regular Expressions

```
irb(main):067:0> x.sub(/t/, "T")  
=> "I wantT to believe"
```

```
irb(main):068:0> puts x  
I want to believe  
=> nil
```

```
irb(main):069:0> x.gsub!(/t/, "T")  
=> "I wantT To believe"
```

```
irb(main):070:0> puts x  
I wantT To believe  
=> nil
```


Conditionals

```
irb(main):071:0> x = "happy"  
=> "happy"
```

```
irb(main):072:0> if x == "happy"  
irb(main):073:1>   puts "Sure am!"  
irb(main):074:1> elsif x == "sad"  
irb(main):075:1>   puts "Boo!"  
irb(main):076:1> else  
irb(main):077:1*   puts "Therapy?"  
irb(main):078:1> end  
Sure am!  
=> nil
```

Conditionals

```
irb(main):079:0> case x
irb(main):080:1> when "happy"
irb(main):081:1>   puts "Sure Am! "
irb(main):082:1>   1
irb(main):083:1> when "sad"
irb(main):085:1>   puts "Boo! "
irb(main):086:1>   2
irb(main):087:1> else
irb(main):088:1>   puts "Therapy? "
irb(main):089:1>   3
irb(main):090:1> end
Sure Am!
=> 1
```

Method Definition

```
irb(main):091:0> def metal(str)
irb(main):092:1>   puts "!!#{str} is
metal!!"
irb(main):093:1> end
=> nil

irb(main):094:0> metal("ozzy")
!!ozzy is metal!!
=> nil
```

Classes

```
irb(main):095:0> class Person
irb(main):096:1>   attr_accessor :name, :is_metal
irb(main):097:1>
irb(main):098:1>   def metal
irb(main):099:2>     if @is_metal
irb(main):100:3>       puts "!!#{@name} is metal!!"
irb(main):101:3>     end
irb(main):102:2>   end
irb(main):103:1> end
=> nil
```

Classes

```
irb(main):104:0> p = Person.new
=> #<Person:0x891ab4c>

irb(main):105:0> p.name = "Adam Jacob"
=> "Adam Jacob"

irb(main):106:0> p.is_metal = true
=> true

irb(main):107:0> p.metal
!!Adam Jacob is metal!!
=> nil

irb(main):108:0> p.is_metal = false
=> false

irb(main):109:0> p.metal
=> nil
```

Review Questions

- What is irb?
- What is true in ruby? What is false?
- How do you press for the truth?
- What does `>=` do?
- How do you define a method?
- What is a class?
- What is an object?
- The book you want: "Programming Ruby 1.9" <http://pragprog.com/book/ruby3/programming-ruby-1-9>