

Ruby - Strings

```
"a string"          'a string'
"Concat" + "nation" "#{con cat e}#{nation}"
astrin g.l ength    astrin g.e mpty?
" a,b ,c".s pl it(' ', Result: [a,b,c]
')
```

Single quotes string, auto escape characters.

Ruby - Conditionals

```
puts "string is not empty" if
!astrin g.empty?
puts " string is not empty"
unless astrin g.e mpty?
if <ex pre ssi on>
...
else
...
end if
```

Ruby - Class (cont)

```
> end
def NiceString
  "#{@attributeA} #{@attributeA}"
end
end
```

Ruby - Array

```
foo = [a,b,c,d]    Used for all examples
                  below
foo[n]             foo.first,foo.secon
                  d
foo.length         foo.last == foo[-1]
foo.empty?         foo.in clude?(x)
foo.sort[!]        foo.sh uffle[!]
foo.re verse[!]    foo.push(x) or foo <
                  < x
foo.jo in( " , " Result: "a, b, c, d"
)
foo[0..2]          Result: [a,b,c]
foo[1..-1]         Result: [b,c,d]
('a...' d' ).to_a == foo
```

Ruby - Block

```
3.times { puts " foo " }    Inline Block
(1..5).each do |i|          Multi Line Block
  puts 2 * i
end
%w[A B C].map { |char| char.d owncase
}
%w[A B C].map (& :do wncase)
```

The last 2 lines result in an array of ["a", " b", " c"]. The %w creates an array of string. The last one is a weird Ruby shortcut.

Ruby - Hash + Symbol

```
foo = { " nam e" => " Joh n", " id" => " JH" }
foo["na me"]              ➡ " Joh n"
foo = { name: " Joh n", id: " JH" }
foo[:id]                  ➡ " JH"
foo.each do |key, value|  ➡ :name - " J
  puts " #{k ey.i ns pect} - #{value.inspe oh n"
ct}"                      ➡ :id - " JH"
end
```

Ruby - Objects

```
anobje ct.nil?          True or False
anobje ct.to_s          To String
```

Ruby - Class

```
module BunchOfMethods
  def module Method1
    ...
  end
  def module Method2
    ...
  end
end
class MyClass < MySupe rClass
  att r_a ccessor :attri -
buteA, :attri buteB
  def initia liz e(a ttr ibutes
= {})
    @at tri buteA = attrib -
ute s[: att rib uteA]
    @at tri buteB = attrib -
ute s[: att rib uteB]
```



By GlutenBoy

cheatography.com/glutenboy/

Not published yet.

Last updated 6th October, 2018.

Page 1 of 1.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>