```
#---------
#But First
#---------
echo "puts 'Hello, World!'" | ruby
puts 'Hello, World!'
print "Hello, World!"
ri String#length

#---------
#Completely OO
#---------
2.methods
-43.abs
1.414.class
'Foobar'.length

#---------
#Duck Typing
#---------
a = 2
print a*3
a = 'Wow! '
print a*3

#---------
#Numbers - Fixnum and Bignum
#---------
Example Needed
a = 2
a.class
a =
2359850983450948598349857398589347539875348968937432942398572894293 87

#---------
#Strings - Single and Double Quote
#---------
print '\n'
puts "\n"
who = "Friday  the 13th of April"
who.split
who[0].chr
who[2, 3]
who[-2, 3]
who.reverse!
who.upcase
who.downcase
who.length
who.chomp
who.gsub('the', 'The')
#who<<" 2007" # Something wrong with my VIm highlighting, had to
comment this

#---------
#Arrays
#---------
arr = ['fee', 'fi', 'fo', 'fum', 2.0]
```

```ruby
arr[0]
arr[-2,1]
arr<<[1,2,3]
arr[-1][0]
arr = %w{fee fi fo fum}
arr.join(', ')


#---------
#Hashes
#---------
music = { 'pop' => 23, 'rock' => 53, 'country' => 12}
music['pop']
music.delete('country')


#---------
#Ranges
#---------
(1..3)
(1...4)
print *(1..3)

#---------
#Expressions and Operators
#---------
3<=>1
'qwerty'<=>'qwerty'
a = 3
b = if a==3 then 2 else 1 end


#---------
#for, if and when
#---------
for i in (1..3)
        puts i
end
a = 1
a *= 2 while a<100
a = 3 if a > 0 unless a < 100

arr = ['fee', 'fi', 'fo', 'fum', 2.0]
for i in arr
        puts i
end

#---------
#Case expressions
#---------
case lang
when /j*/
        puts 'Coffee!'
when 'python'
        puts 'Hiss!'
when 'Ruby'
        puts 'konnichiwa!'
```

```ruby
end

year = 1900
leap = case
           when year%400==0: true
           when year%100==0: false
           else year%4==0
           end

#---------
# Closures
#---------
5.times{|x| puts x}
File.read('file_name').each do |x| # Every Enumerable has an each
method
        puts x
end

arr = ['1', '2', '3', '4']
shorlist = arr.map{|x| x.to_i}.select{|x| x>2} # Selects only items
greater than 2
2.upto(100){|x| puts x if x%2==0} # There is also a downto
memo=0
(1..10).inject{|memo, obj| memo += obj}

#---------
#So, you want your own iterators eh?
#---------
def n_factorial_times(n, &block)
        fact = 1
        (1..n).inject(1) {|fact, i| fact*=i}
        for i in 0..fact do
                yield i
        end
end
n_factorial_times(6) {|x| puts x}

class Array
        def each_even
                self.each{|x| yield(x) if x%2==0}
        end
end
[1,2,3,4,5,6,7].each_even{|n| puts n} # 2, 4, 6


#---------
#Regular Expressions
#---------
while line = gets
        puts line if line =~ /^start.*/ .. line =~ /^stop.*/
end

'kjashu2453kdfj' =~ /\d+/
'kjashu2453kdfj' =~ /\booga/
'kjashu2453kdfj' =~ /booga/
'kjashu2453kdfj'.gsub('k', 'K')
a ='kjashu2453kdfj'.gsub('k', 'K')
```

```ruby
a.gsub('j', 'J')
a.gsub!('j', 'J')
a = 'Abracadabra'.gsub(/[aeiou]/){|x| x.upcase}

#---------
#Classes in Rubyland
#---------

class Foo
        def greet(who)
                puts "Hello #{who}"
        end
end

#---------
# Accessors
#---------

class User
        def initialize(name)
                @name = name
        end
        def name=(name)
                @name = name
        end
end

class User2
        attr_accessor :name
        #also available attr_reader and attr_writer
        def initialize(name)
                @name = name
        end
end


#---------
# Modules and Mixins
#---------
module Lol
        def i_can_has_cheezburger?
                if self.class.name == 'LolCat'
                        true
                else false
                end
        end
end

class LolCat
        include Lol
end

class Bar
        include Lol
end

a, b = LolCat.new, Bar.new
```

```ruby
a.i_can_has_cheezburger?
b.i_can_has_cheezburger?

class Player
        include Comparable
        attr_reader :score
        def initialize(score)
                @score = score
        end
        def <=> (other)
                self.score <=> other.score
        end
end

#---------
# Exception Handling
#---------
def read_file(file_name)
        begin
                File.read(file_name)
        rescue
                puts "Error Recovery"
end

#---------
#Closures
#File reading
#regex in while loops
#retry, continue and redo
#Backtick operator
# #{}
#Map, reduce, other functional stuff
#Only nil and false are false
#Global $ vars
```