

```
1                                     # Ruby.rb
2
3 # Note:
4 #     =begin/ =end/ (#).
5 #     are Comment tags
6
7 =begin
8                                     Hello,
9                                     I am Athresh Kumar Labde
10                                    This Document is an basic
11                                    overview of programming language
12                                    Ruby
13                                    For any Queries do contact me
14                                    Gmail: athreshkumar2001@gmail.com
15 =end
16
17
18 # Printing
19 puts "Hello"
20 print "World"
21 puts "!"
22
23 # Variables and Data Types
24
25 =begin
26   Names are case-sensitive and may begin with:
27       letters, _
28   After, may include
29       letters, numbers, _
30   Convention says
31       Start with a lowercase word, then additional words are lowercase separated
32       by an underscore
33       ex. my_first_variable
34 =end
35 name = "Athresh"      # Strings
36 age = 20              # Integer
37 gpa = 2.0             # Decimal
38 is_tall = true        # Boolean → True/False
39
40 name = "John"
41
42 puts "Your name is #{name}"
43 puts "Your name is " + name
44
45
46 # Casting and Converting
47 puts 3.14.to_i
48 puts 3.to_f
49 puts "3.0".to_s
50
51 puts 100 + "50".to_i
52 puts 100 + "50.99".to_f
53
54
55 # Strings
56
57 greeting = "Hello"
58 #indexes:  01234
59
60 puts greeting.length
```

```
61 puts greeting[0]
62 puts greeting.include? "llo"
63 puts greeting.include? "z"
64 puts greeting[1,3]
65
66
67 # Numbers
68
69 puts 2 * 3          # Basic Arithmetic: +, -, /, *
70 puts 2**3          # Exponent
71 puts 10 % 3         # Modulus Op. : returns remainder of 10/3
72 puts 1 + 2 * 3      # order of operations
73 puts 10 / 3.0       # int's and doubles
74
75
76 num = 10
77 num += 100          # +=, -=, /=, *=
78 puts num
79
80 num = -36.8
81 puts num.abs()
82 puts num.round()
83
84 # Math class has useful math methods
85 puts Math.sqrt(144)
86 puts Math.log(0)
87
88
89
90 # User Input
91
92 puts "Enter your name: "
93 name = gets          #.chomp
94 puts "Hello #{name}, how are you"
95
96 puts "Enter first num: "
97 num1 = gets.chomp
98 puts "enter second num: "
99 num2 = gets.chomp
100 puts num1.to_f + num2.to_f
101
102
103 # Arrays
104
105 lucky_numbers = [4, 8, "fifteen", 16, 23, 42.0]
106 #           indexes 0 1         2         3  4  5
107
108 lucky_numbers[0] = 90
109 puts lucky_numbers[0]
110 puts lucky_numbers[1]
111 puts lucky_numbers[-1]
112
113 puts "\n\n"
114 puts lucky_numbers[2,3]
115 puts "\n\n"
116 puts lucky_numbers[2..4]
117 puts "\n\n"
118
119 puts lucky_numbers.length
120
```

```
121
122 # 2 Dimensional Arrays
123
124 number_grid = [[],[ ]]
125 # numberGrid = [ [1, 2], [3, 4] ]
126 number_grid[0][0] = 99
127
128 puts number_grid[0][0]
129 puts number_grid[0][1]
130
131 # Array Methods
132
133 friends = []
134 friends.push("Athresh")
135 friends.push("kumar")
136 friends.push("ashish")
137
138 # friends.pop
139 puts friends
140 puts "\n"
141
142 puts friends.reverse
143 puts "\n"
144
145 puts friends.sort
146 puts "\n"
147
148 puts friends.include? "drake"
149
150
151 # Methods
152
153 def add_numbers(num1, num2=99)
154   return num1 + num2
155 end
156
157 sum = add_numbers(4, 3)
158 puts sum
159
160 # If Statements
161
162 is_student = false
163 is_smart = false
164
165 if is_student and is_smart
166   puts "You are a student"
167 elsif is_student and !is_smart
168   puts "You are not a smart student"
169 else
170   puts "You are not a student and not smart"
171 end
172
173 # >, <, ≥, ≤, ≠, ==, String.equals()
174 if 1 > 3
175   puts "number comparison was true"
176 end
177
178 if "a" > "b"
179   puts "string comparison was true"
180 end
```

```
181
182 # Switch Statements
183
184 my_grade = "A"
185 case my_grade
186   when "A"
187     puts "You Pass"
188   when "F"
189     puts "You fail"
190   else
191     puts "Invalid grade"
192 end
193
194
195 # Dictionaries
196
197 test_grades = {
198   "Athresh" => "B+",
199   :Stanley => "C",
200   "Rohith" => "A",
201   3 => 95.2
202 }
203
204 test_grades["Athresh"] = "B-"
205 puts test_grades["Athresh"]
206 puts test_grades[:Stanley]
207 puts test_grades[3]
208
209
210 # For Loops
211
212 for index in 0..5
213   puts index
214 end
215
216 5.times do |index|
217   puts index
218 end
219
220 lucky_nums = [4, 8, 15, 16, 23, 42]
221 for lucky_num in lucky_nums
222   puts lucky_num
223 end
224
225 lucky_nums.each do |lucky_num|
226   puts lucky_num
227 end
228
229
230 # Exception Catching
231
232 begin
233   # puts bad_variable
234   num = 10/0
235 rescue ZeroDivisionError
236   puts "Error"
237 rescue
238   puts "All other errors"
239 end
240
```

```
241 raise "Made Up Exception"
242
243
244 # Classes and Objects
245
246 class Book
247     attr_accessor :title, :author
248
249     def readBook()
250         puts "Reading #{self.title} by #{self.author}"
251     end
252 end
253
254 book1 = Book.new()
255 book1.title = "Harry Potter"
256 book1.author = "JK Rowling"
257
258 book1.readBook()
259 puts book1.title
260
261
262
263 # Constructors
264 class Book
265     attr_accessor :title, :author
266     def initialize(title, author)
267         @title = title
268         @author = author
269     end
270
271     def readBook()
272         puts "Reading #{self.title} by #{@author}"
273     end
274 end
275
276 book1 = Book.new("Harry Potter", "JK Rowling")
277 # book1.title = "Half-Blood Prince"
278
279 puts book1.title
280
281
282
283 # Getters and Setters
284 class Book
285     attr_accessor :title, :author
286     def initialize(title, author)
287         self.title = title
288         @author = author
289     end
290
291     def title=(title)
292         puts "Set"
293         @title = title
294     end
295     def title
296         puts "Get"
297         return @title
298     end
299 end
300
```

```
301 book1 = Book.new("Harry Potter", "JK Rowling")
302
303 puts book1.title
304
305
306
307 # Inheritance
308 class Chef
309
310     attr_accessor :name, :age
311     def initialize(name, age)
312         @name = name
313         @age = age
314     end
315
316     def make_chicken()
317         puts "The chef makes chicken"
318     end
319
320     def make_salad()
321         puts "The chef makes salad"
322     end
323
324     def make_special_dish()
325         puts "The chef makes a special dish"
326     end
327 end
328
329 class ItalianChef < Chef
330
331     attr_accessor :country_of_origin
332     def initialize(name, age, country_of_origin)
333         @country_of_origin = country_of_origin
334         super(name, age)
335     end
336
337     def make_pasta()
338         puts "The chef makes pasta"
339     end
340
341     def make_special_dish()
342         puts "The chef makes chicken parm"
343     end
344 end
345
346 my_chef = Chef.new()
347 my_chef.make_chicken()
348
349 my_italian_chef = ItalianChef.new()
350 my_italian_chef.make_chicken()
351
352
353
354
355
356
357
358 # Athresh Kumar Labde
359 # athreshkumar2001
360 # Instagram: athresh_jr
```