**RUBY**

Ruby is an object oriented, multi-paradigm programming language, most commonly used for desktop app development. This language is similar in syntax to Perl and Python but casts a much wider net on functionality. The following is a quick reference sheet for the basic syntax needed to get started in Ruby.

This guide pulls primarily from the Ruby Basics page on TutorialsPoint, linked below.

<https://www.tutorialspoint.com/ruby/index.htm>

You can install Ruby here:

<https://www.ruby-lang.org/en/>

You can learn more about ‘Ruby’ vs ‘Ruby on Rails’ here:

<https://www.appventurez.com/blog/ruby-vs-ruby-on-rails/>

**RESERVED WORDS**

* These words can’t be variable names, but can be method names

|  |  |  |  |
| --- | --- | --- | --- |
| BEGIN | do | next | then |
| END | else | nil | true |
| alias | elsif | not | undef |
| and | end | or | unless |
| begin | ensure | redo | until |
| break | false | rescue | when |
| case | for | retry | while |
| class | if | return | \_\_FILE\_\_ |
| def | in | self | \_\_LINE\_\_ |
| defined? | module | super |  |

**VARIABLES**

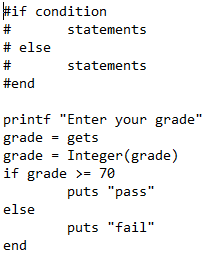
* Basic Literals
  + Integer numbers
  + Floating numbers
  + String literals
  + Arrays
  + Hashes
    - list of key/value pairs between braces, separated by commas, using => notation between the key and the value
  + Ranges
    - Formatted (1..5) or (1…5)
      * (1..5) includes values 1,2,3,4,5
      * (1…5) includes values 1,2,3,4

**OPERATORS**

* Arithmetic
  + +, -, \*, /, %, \*\*
    - Addition, Subtraction, Multiplication, Division, Modulus, Exponents
* Comparison
  + ==, !=, > or >=, < or <=, <=>
  + .eql?
    - Checks if both operands have the same type and equal values
* Assignment
  + =, +=, -=, \*=, /=, %=, \*\*=
* Logical
  + and
  + or
  + &&
    - if both operands are non-zero, the expression is true
  + ||
    - if any operand is non-zero, the expression is true
  + ! or not
    - reverses the logical state of the operand

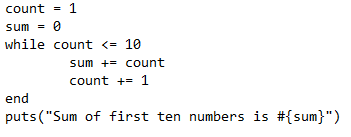
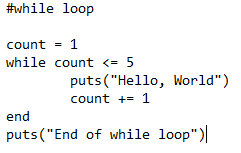
**COMMENTS**

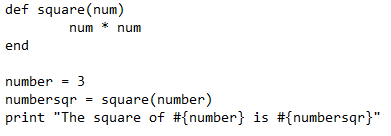
* Single line comments begin with #
* Multiline comments use =begin and =end notation

**IF-ELSE**

* if, elsif, else statements

**LOOPS**

* While loop:
  + Follows the format: while conditional do \n code \n end
* For loop:
  + for variable in expression \n code \n
  + break will terminate the most internal loop
  + next jumps to the next iteration of the most internal loop

**METHODS**

* follows the format: def method\_name (parameters) \n expr… \n end