Reserved Words:

The following list shows the reserved words in Ruby. These reserved words may not be used as constant or variable names. They can, however, be used as method names.

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

|  |
| --- |
| BEGIN {  code  } |

Declares *code* to be called before the program is run.

|  |
| --- |
| END {  code  } |

Declares *code* to be called at the end of the program.

|  |
| --- |
| class Customer  end |

You terminate a class by using the keyword *end*. All the data members in the *class* are between the class definition and the *end* keyword.

# Variables in a Ruby Class:

Ruby provides four types of variables:

* **Local Variables:** Local variables are the variables that are defined in a method. Local variables are not available outside the method. You will see more detail about method in subsequent chapter. Local variables begin with a lowercase letter or \_.
* **Instance Variables:** Instance variables are available across methods for any particular instance or object. That means that instance variables change from object to object. Instance variables are preceded by the at sign (@) followed by the variable name.
* **Class Variables:** Class variables are available across different objects. A class variable belongs to the class and is a characteristic of a class. They are preceded by the sign @@ and are followed by the variable name.
* **Global Variables:** Class variables are not available across classes. If you want to have a single variable, which is available across classes, you need to define a global variable. The global variables are always preceded by the dollar sign ($).