

Section Review

Learn to Code with Ruby

Intro to Methods

- A **method** is a procedure, a function, a sequence of steps to follow.
- Methods encapsulate repeated logic/behavior so that we do not have to write it multiple times.
- Invoke methods by writing their name with an optional parentheses. ***Declaring*** a method is a different idea from ***invoking*** the method.
- Declare a method with the **def** keyword followed by a name. Add the **end** keyword. Any lines between **def** and **end** belong to that method. Indent them.

Parameters and Arguments

- A **parameter** is a name for an expected input that will be provided to a method when it is invoked. Methods can have any number of parameters, including 0.
- An **argument** is the concrete value that we pass for a method's parameter when we invoke the method.
- When invoking a method, the number of arguments must equal the number of parameters.
- If a method includes one or more parameters, I recommend adding parentheses in the method invocation.

Local Variables

- A **local variable** is a variable declared within a method body. It is “local” or “limited” to that execution context.
- **Local variables** exist only as long as the method is running.
- **Scope** refers to the “area” or “boundary” of the program where a name like a variable exists.
- A method body is a separate scope from the top-level Ruby file. A variable within a method body is a different one from a variable with the same name within the file.

Return Values

- A **return value** is the final output of a method. It is what the method *gives back* to the main program.
- All Ruby methods must produce a return value (i.e., an object).
- The **return** keyword terminates a method and returns the evaluation/object on that line.
- Without a **return** keyword, Ruby will implicitly return the evaluation/object from the last line in the method.
- A method with no body returns **nil**.