

You are logged in as [C. O'Keefe](#) ([Logout](#))

Core Ruby 41st Batch

Home ► PORPC101-41C ► 12 April - 18 April ► Week 2: Tutorial

Navigation

Home

■ My home

Site pages

My profile

Current course





PORPC101-41C

Participants

General

5 April - 11 April

12 April - 18 April

 **Week 2: Tutorial** Week 2:
Exercises Week 2: Forum Some useful
Ruby methods Recipes THIMK *PARALLEL
TRACK* Learn
basic HTML and
CSS Week 2: Quiz

19 April - 25 April

26 April - 2 May

3 May - 9 May

10 May - 16 May

17 May - 23 May

24 May - 30 May

31 May - 6 June

7 June - 13 June

Week 2: Tutorial

Before you begin:

First complete: Week 1.

REMEMBER:

- No question is too silly to ask. Just ask it in the relevant forums.
- Have you all been solving the exercises that have been given on the relevant pages? Let me know whether you are able to solve the same or having problems.

Week 2

a. You need to read through the following pages, in the given order. While doing so, please make a note of all your doubts, queries, questions, clarifications and after you have completed all the pages, post these on the forum here. There may be some questions that relate to something that has not been mentioned or discussed by me here; you could post the same too.

- [Ruby ri Tool](#)
- [More on Strings](#)
- [Simple Constructs](#)
- [Ruby Blocks](#)

b. Some of the **important points to remember** after you have read through the above pages are:

- Refer to the **String** documentation to use the various methods available.
- For double-quoted string literals, Ruby looks for

My courses

Settings

Course administration

My profile settings

substitutions - sequences that start with a backslash character - and replaces them with some binary value or does expression interpolation ie. within the string, the sequence `# {expression}` is replaced by the value of the expression.

- It is to be noted that every time a string literal is used in an assignment or as a parameter, a new **String** object is created.
- `%w` is a common usage in strings.
- Observe how one can list all the methods of a class or object.
- Comparing two strings for equality can be done by `==` or `.eq!` (for identical content) and `.equal?` (for identical objects).
- Observe the usage of constructs: **if else end**, **while**, **if elsif end**
- Case Expressions: This form is fairly close to a series of **if** statements: it lets you list a series of conditions and execute a statement corresponding to the first one that's true. **case** returns the value of the last expression executed. Usage: **case when else end**
- Ruby also has a negated form of the **if** statement, the **unless end**.
- **IMPORTANT:** *I'd suggest that you read the topic "Ruby Blocks" a couple of times, as it's an extremely important topic and used everywhere in Ruby.* Ruby Code blocks are chunks of code between braces or between `do- end` that you can associate with method invocations.
- Code blocks may appear only in the source adjacent to a method call; the block is written starting on the same line as the method call's last parameter (or the closing parenthesis of the parameter list). The code in the block is not executed at the time it is encountered. Instead, Ruby remembers the context in which the block appears (the local variables, the current object, and so on) and then enters the method.

- The Ruby standard is to use braces for single-line blocks and do- end for multi-line blocks. Keep in mind that the braces syntax has a higher precedence than the do..end syntax.
- Inside a method, you can call a Ruby block using the **yield** keyword with a value.
- You can provide parameters to the call to **yield**: these will be passed to the block. Within the block, you list the names of the arguments to receive the parameters between vertical bars (|).

c. Exercises:

Please complete the Week 2 exercises and discuss in the Week 2 forum.

d. Quiz:

Please take the quiz after you have completed Week 2. All the questions are either of multiple-choice and/or true/false and based on what you have learned in Week 2. You have only 2 attempts to complete the same. Grades will be allotted for the quiz.

Last modified: Friday, 1 June 2012, 10:06 PM

You are logged in as [C. O'Keefe](#) ([Logout](#))



© 2008-2013 RubyLearning.org : A Ruby Learning Hub