Top 53 Ruby on Rails Interview Questions & Answers

1) Explain what is Ruby on Rails?

- **Ruby:** It is an object oriented programming language inspired by PERL and PYTHON.
- **Rails:** It is a framework used for building web application

2) Explain what is class libraries in Ruby?

Class libraries in Ruby consist of a variety of domains, such as data types, thread programming, various domains, etc.

3) Mention what is the naming convention in Rails?

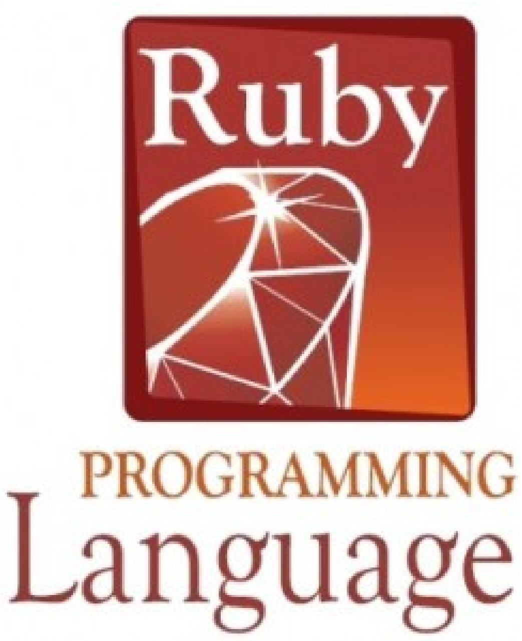
- **Variables:** For declaring Variables, all letters are lowercase, and words are separated by underscores
- **Class and Module:** Modules and Classes uses MixedCase and have no underscore; each word starts with a uppercase letter
- **Database Table:** The database table name should have lowercase letters and underscore between words, and all table names should be in the plural form for example invoice_items
- **Model:** It is represented by unbroken MixedCase and always have singular with the table name
- **Controller:** Controller class names are represented in plural form, such that OrdersController would be the controller for the order table.

4) Explain what is “Yield” in Ruby on Rails?

A Ruby method that receives a code block invokes it by calling it with the “Yield”.

5) Explain what is ORM (Object-Relationship-Model) in Rails?

ORM or Object Relationship Model in Rails indicate that your classes are mapped to the table in the database, and objects are directly mapped to the rows in the table.



6) Mention what the difference is between false and nil in Ruby?

In Ruby False indicates a Boolean datatype, while Nil is not a data type, it have an object_id 4.

7) Mention what are the positive aspects of Rails?

Rails provides many features like

- **Meta-programming:** Rails uses code generation but for heavy lifting it relies on meta-programming. Ruby is considered as one of the best language for Meta-programming.
- **Active Record:** It saves object to the database through Active Record Framework. The Rails version of Active Record identifies the column in a schema and automatically binds them to your domain objects using metaprogramming
- **Scaffolding:** Rails have an ability to create scaffolding or temporary code automatically
- **Convention over configuration:** Unlike other development framework, Rails does not require much configuration, if you follow the naming convention carefully
- **Three environments:** Rails comes with three default environment testing, development, and production.
- **Built-in-testing:** It supports code called harness and fixtures that make test cases to write and execute.

8) Explain what is the role of sub-directory app/controllers and app/helpers?

- App/controllers: A web request from the user is handled by the Controller. The controller sub-directory is where Rails looks to find controller classes

- App/helpers: The helper’s sub-directory holds any helper classes used to assist the view, model and controller classes.

9) Mention what is the difference between String and Symbol?

They both act in the same way only they differ in their behaviors which are opposite to each other. The difference lies in the object_id, memory and process tune when they are used together. Symbol belongs to the category of immutable objects whereas Strings are considered as mutable objects.

10) Explain how Symbol is different from variables?

Symbol is different from variables in following aspects

- It is more like a string than variable
- In Ruby string is mutable but a Symbol is immutable
- Only one copy of the symbol requires to be created
- Symbols are often used as the corresponding to enums in Ruby

11) Explain what is Rails Active Record in Ruby on Rails?

Rails active record is the Object/Relational Mapping (ORM) layer supplied with Rails. It follows the standard ORM model as

- Table map to classes
- Rows map to objects
- Columns map to object attributes

12) Explain how Rails implements Ajax?

Ajax powered web page retrieves the web page from the server which is new or changed unlike other web-page where you have to refresh the page to get the latest information.

Rails triggers an Ajax Operation in following ways

- **Some trigger fires:** The trigger could be a user clicking on a link or button, the users inducing changes to the data in the field or on a form
- **Web client calls the server:** A Java-script method, XMLHttpRequest, sends data linked with the trigger to an action handler on the server. The data might be the ID of a checkbox, the whole form or the text in the entry field
- **Server does process:** The server side action handler does something with the data and retrieves an HTML fragment to the web client
- **Client receives the response:** The client side JavaScript, which Rails generates automatically, receives the HTML fragment and uses it to update a particular part of the current

13) Mention how you can create a controller for subject?

To create a controller for subject you can use the following command

C:\ruby\library> ruby script/generate controller subject

14) Mention what is Rails Migration?

Rails Migration enables Ruby to make changes to the database schema, making it possible to use a version control system to leave things synchronized with the actual code.

15) List out what can Rails Migration do?

Rails Migration can do following things

- Create table
- Drop table
- Rename table
- Add column
- Rename column
- Change column
- Remove column and so on

16) Mention what is the command to create a migration?

To create migration command includes

C:\ruby\application>ruby script/generate migration table_name

17) Explain when self.up and self.down method is used?

When migrating to a new version, **self.up** method is used while **self.down** method is used to roll back my changes if needed.

18) Mention what is the role of Rails Controller?

The Rails controller is the logical center of the application. It faciliates the interaction between the users, views, and the model. It also performs other activities like

- It is capable of routing external requests to internal actions. It handles URL extremely well
- It regulates helper modules, which extend the capabilities of the view templates without bulking of their code
- It regulates sessions; that gives users the impression of an ongoing interaction with our applications

19) Mention what is the difference between Active support's "HashWithIndifferent" and Ruby's "Hash" ?

The **Hash** class in Ruby's core library returns value by using a standard **"= ="** comparison on the keys. It means that the value stored for a **symbol** key cannot be retrieved using the equivalent string. While the **HashWithIndifferentAccess** treats Symbol keys and String keys as equivalent.

20) Explain what is Cross-Site Request Forgery (CSRF) and how Rails is protected against it?

CSRF is a form of attack where hacker submits a page request on your behalf to a different website, causing damage or revealing your sensitive data. To protect from CSRF attacks, you have to add **"protect_from_forgery"** to your **ApplicationController**. This will cause Rails to require a CSRF token to process the request. CSRF token is given as a hidden field in every form created using Rails form builders.

21) Explain what is Mixin in Rails?

Mixin in Ruby offers an alternative to multiple inheritances, using mixin modules can be imported inside other class.

22) Explain how you define Instance Variable, Global Variable and Class Variable in Ruby?

- Ruby Instance variable begins with -- @
- Ruby Class variables begin with -- @@
- Ruby Global variables begin with -- **$**

23) Explain how you can run Rails application without creating databases?

You can execute your application by uncommenting the line in environment.rb

path=> rootpath conf/environment.rb

config.frameworks = [action_web_service, :action_mailer, :active_record]

24) Mention what is the difference between the Observers and Callbacks in Ruby on Rails?

- **Rails Observers:** Observers is same as Callback, but it is used when method is not directly associated to object lifecycle. Also, the observer lives longer, and it can be detached or attached at any time. For example, displaying values from a model in the UI and updating model from user input.
- **Rails Callback:** Callbacks are methods, which can be called at certain moments of an object's life cycle for example it can be called when an object is validated, created,

updated, deleted, A call back is short lived. For example, running a thread and giving a call-back that is called when thread terminates

25) Explain what is rake in Rails?

Rake is a Ruby Make; it is a Ruby utility that substitutes the Unix utility ‘make’, and uses a ‘Rakefile’ and ‘.rake files’ to build up a list of tasks. In Rails, Rake is used for normal administration tasks like migrating the database through scripts, loading a schema into the database, etc.

26) Explain how you can list all routes for an application?

To list out all routes for an application you can write rake routes in the terminal.

27) Explain what is sweeper in Rails?

Sweepers are responsible for expiring or terminating caches when model object changes.

28) Mention the log that has to be seen to report errors in Ruby Rails?

Rails will report errors from Apache in the log/Apache.log and errors from the Ruby code in log/development.log.

29) Explain what is the difference between Dynamic and Static Scaffolding?

Dynamic Scaffolding	**Static Scaffolding**
• It automatically creates the entire content and user interface at runtime	• It requires manual entry in the command to create the data with their fields
• It enables to generation of new, delete, edit methods for the use in application	• It does not require any such generation to take place
• It does not need a database to be synchronized	• It requires the database to be migrated

30) Mention what is the function of garbage collection in Ruby on Rails?

The functions of garbage collection in Ruby on Rails includes

- It enables the removal of the pointer values which is left behind when the execution of the program ends
- It frees the programmer from tracking the object that is being created dynamically on runtime
- It gives the advantage of removing the inaccessible objects from the memory, and allows other processes to use the memory

31) Mention what is the difference between redirect and render in Ruby on Rails?

- Redirect is a method that is used to issue the error message in case the page is not issued or found to the browser. It tells browser to process and issue a new request.

- Render is a method used to make the content. Render only works when the controller is being set up properly with the variables that require to be rendered.

32) Mention what is the purpose of RJs in Rails?

RJs is a template that produces JavaScript which is run in an eval block by the browser in response to an AJAX request. It is sometimes used to define the JavaScript, Prototype and helpers provided by Rails.

33) Explain what is Polymorphic Association in Ruby on Rails?

Polymorphic Association allows an ActiveRecord object to be connected with Multiple ActiveRecord objects. A perfect example of Polymorphic Association is a social site where users can comment on anywhere whether it is a videos, photos, link, status updates etc. It would be not feasible if you have to create an individual comment like photos_comments, videos_comment and so on.

34) Mention what are the limits of Ruby on Rails?

Ruby on Rails has been designed for creating a CRUD web application using MVC. This might make Rails not useful for other programmers. Some of the features that Rails does not support include

- Foreign key in databases
- Linking to multiple data-base at once
- Soap web services
- Connection to multiple data-base servers at once

35) Mention what is the difference between calling super() and super call?

- **super()**: A call to super() invokes the parent method without any arguments, as presumably expected. As always, being explicit in your code is a good thing.
- **super call**: A call to super invokes the parent method with the same arguments that were passed to the child method. An error will therefore occur if the arguments passed to the child method don’t match what the parent is expecting.

36) Explain about Dig, Float and Max?

- Float class is used whenever the function changes constantly.
- Dig is used whenever you want to represent a float in decimal digits.
- Max is used whenever there is a huge need of Float.

37) Explain how can we define Ruby regular expressions?

Ruby regular expression is a special sequence of characters that helps you match or find other strings. A regular expression literal is a pattern between arbitrary delimiters or slashes followed by %r.

38) Explain what is the defined operator?

Define operator states whether a passed expression is defined or not. If the expression is defined, it returns the description string and if it is not defined it returns a null value.

39) List out the few features of Ruby?

- Free format – You can start writing from program from any line and column
- Case sensitive – The uppercase and lowercase letters are distinct
- Comments - Anything followed by an unquoted #, to the end of the line on which it appears, is ignored by the interpreter
- Statement delimiters- Multiple statements on one line must be separated by semicolons, but they are not required at the end of a line.

40) Mention the types of variables available in Ruby Class?

Types of variables available in Ruby Class are,

- Local Variables
- Global Variables
- Class Variables
- Instance Variables

41) Explain how can you declare a block in Ruby?

In Ruby, the code in the block is always enclosed within braces ({}). You can invoke a block by using “yield statement”.

42) Explain what is the difference between put and putc statement?

Unlike the puts statement, which outputs the entire string onto the screen. The Putc statement can be used to output one character at a time.

43) Explain what is a class library in Ruby?

Ruby class libraries consist of a variety of domains, such as thread programming, data types, various domains, etc. These classes give flexible capabilities at a high level of abstraction, giving you the ability to create powerful Ruby scripts useful in a variety of problem domains. The following domains which have relevant class libraries are,

- GUI programming
- Network programming
- CGI Programming
- Text processing

44) In Ruby, it explains about the defined operator?

The defined operator tells whether a passed expression is defined or not. If the expression is not defined, it gives null, and if the expression is defined it returns the description string.

45) Mention what is the difference in scope for these two variables: @@name and @name?

The difference in scope for these two variables is that:

- @@name is a class variable
- @name is an instance variable

46) Mention what is the syntax for Ruby collect Iterator?

The syntax for Ruby collect Iterator collection = collection.collect.

47) In Ruby code, often it is observed that coder uses a short hand form of using an expression like array.map(&:method_name) instead of array.map { |element| element.method_name }. How this trick actually works?

When a parameter is passed with "&" in front of it. Ruby will call to_proc on it in an attempt to make it usable as a block. So, symbol to_Proc will invoke the method of the corresponding name on whatever is passed to it. Thus helping our shorthand trick to work.

48) Explain what is Interpolation in Ruby?

Ruby Interpolation is the process of inserting a string into a literal. By placing a Hash (#) within {} open and close brackets, one can interpolate a string into the literal.

49) Mention what is the Notation used for denoting class variables in Ruby?

In Ruby,

- A constant should begin with an uppercase letter, and it should not be defined inside a method
- A local must begin with the _ underscore sign or a lowercase letter
- A global variable should begin with the $ sign. An uninitialized global has the value of "nil" and it should raise a warning. It can be referred anywhere in the program.
- A class variable should begin with double @@ and have to be first initialized before being used in a method definition

50) Mention what is the difference between Procs and Blocks?

The difference between Procs and Blocks,

- Block is just the part of the syntax of a method while proc has the characteristics of a block
- Procs are objects, blocks are not
- At most one block can appear in an argument list
- Only block is not able to be stored into a variable while Proc can

51) Mention what is the difference between a single quote and double quote?

A single-quoted strings don't process ASCII escape codes, and they don't do string interpolation.

51) Mention what is the difference between a gem and a plugin in Ruby?

- **Gem:** A gem is a just ruby code. It is installed on a machine, and it’s available for all ruby applications running on that machine.
- **Plugin:** Plugin is also ruby code, but it is installed in the application folder and only available for that specific application.

53) Mention what is the difference extend and include?

The “include” makes the module’s methods available to the instance of a class, while “extend” makes these methods available to the class itself.