Ruby Developer Job Interview Questions And Answers

Interview Questions Answers

https://interviewquestionsanswers.org/

About Interview Questions Answers

Interview Questions Answers . ORG is an interview preparation guide of thousands of Job Interview Questions And Answers, Job Interviews are always stressful even for job seekers who have gone on countless interviews. The best way to reduce the stress is to be prepared for your job interview. Take the time to review the standard interview questions you will most likely be asked. These interview questions and answers on Ruby Developer will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts.

If you find any **question or answer** is incorrect or incomplete then you can **submit your question or answer** directly with out any registration or login at our website. You just need to visit Ruby Developer Interview Questions And Answers to add your answer click on the *Submit Your Answer* links on the website; with each question to post your answer, if you want to ask any question then you will have a link *Submit Your Question*; that's will add your question in Ruby Developer category. To ensure quality, each submission is checked by our team, before it becomes live. This Ruby Developer Interview preparation PDF was generated at **Wednesday 29th November, 2023**

You can follow us on FaceBook for latest Jobs, Updates and other interviews material.
www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter for latest Jobs and interview preparation guides.
https://twitter.com/InterviewQA

If you need any further assistance or have queries regarding this document or its material or any of other inquiry, please do not hesitate to contact us.

Best Of Luck.

Interview Questions Answers.ORG Team
https://InterviewQuestionsAnswers.ORG/
Support@InterviewQuestionsAnswers.ORG

Ruby Developer Interview Questions And Answers Guide.

Question - 1:

Tell me how can you declare a block in Ruby?

Ans:

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

View All Answers

Question - 2:

Do you know about Dig, Float and Max?

Ans:

* 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.

View All Answers

Question - 3:

Explain me what is the purpose of RJs in Rails?

Ans:

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.

View All Answers

Question - 4:

Tell me what is Mixin in Rails?

Ans:

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

View All Answers

Question - 5:

What is interpolation in Ruby?

Ans:

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.

View All Answers

Question - 6:

Tell us the types of variables available in Ruby Class?

Ans:

Types of variables available in Ruby Class are,
* Local Variables
* Global Variables
* Class Variables
* Instance Variables

View All Answers

Question - 7:

Tell me what is Rails Migration?

Ans:

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.

View All Answers

Question - 8:

Do you know when self.up and self.down method is used?

Ans:

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

View All Answers

Question - 9:

Tell me what is Rails Active Record in Ruby on Rails?

Ans:

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

View All Answers

Question - 10:

Explain me the log that has to be seen to report errors in Ruby Rails?

Ans:

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

View All Answers

Question - 11:

What is the difference between a single quote and double quote?

Ans:

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

View All Answers

Question - 12:

Tell me how you can list all routes for an application?

Ans:

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

View All Answers

Question - 13:

Can you list out the few features of Ruby?

Ans:

* 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.

View All Answers

Question - 14:

Please explain the three levels of access control for Ruby methods?

Ans:

In Ruby, methods may either be public, protected, or private. Public methods can be called by anyone. Protected methods are only accessible within their defining class and its subclasses. Private methods can only be accessed and viewed within their defining class.

View All Answers

Question - 15:

Tell me what is the function of garbage collection in Ruby on Rails?

Ans:

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

View All Answers

Question - 16:

Can you please explain what is the difference between String and Symbol?

Ans:

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.

View All Answers

Question - 17:

Explain me what is the role of Rails Controller?

Ans:

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

View All Answers

Question - 18:

Tell me what is Polymorphic Association in Ruby on Rails?

Ans:

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.

View All Answers

Question - 19:

Do you know what are the types of caching used in rails, what are they?

Ans:

There are primarily three types of caching used in Rails.
1. Page Caching
Page caching is a Rails mechanism which allows the request for a generated page to be fulfilled by the webserver (i.e. Apache or NGINX) without having to go through the entire Rails stack.
2. Action Caching
Page Caching cannot be used for actions that have before filters - for example, pages that require authentication. This is where Action Caching comes in. Action Caching works like Page Caching except the incoming web request hits the Rails stack so that before filters can be run on it before the cache is served. This allows authentication and other restrictions to be run while still serving the result of the output from a cached copy.
3. Fragment Caching
Dynamic web applications usually build pages with a variety of components not all of which have the same caching characteristics. When different parts of the page need to be cached and expired separately you can use Fragment Caching.

View All Answers

Question - 20:

Tell me what is the difference between calling super and calling super()?

Ans:

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.
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.

View All Answers

Question - 21:

What is the difference between a gem and a plugin in Ruby?

Ans:

* 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.

View All Answers

Question - 22:

Tell us what is the difference between redirect and render in Ruby on Rails?

Ans:

* 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.

View All Answers

Question - 23:

Tell me what is ORM (Object-Relationship-Model) in Rails?

Ans:

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.

View All Answers

Question - 24:

What is the difference between Procs and Blocks?

Ans:

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

View All Answers

Question - 25:

Tell us how would you implement hash in Ruby internally?

Ans:

The purpose of a hash function is to convert a given a key into an integer of limited range. In order to reduce the range, we use a technique called the division method. In the division method, the key is divided by the size of the storage and the remainder is the location inside that table where a record can be stored.
But in real life programming the keys includes strings, objects, etc as well. This is solved by using a one-way hash function over the key and then applying the division method to get the location. The hash function is a mathematical function that takes a string of any length and produces a fixed length integer value. The hash data structure derives it's name from this hashing mechanism. Ruby uses the murmur hash function and then applies the division method with a prime number M, which Ruby determines based on the table size that is needed for storage.

View All Answers

Question - 26:

Tell me how do you remove nil values in array using ruby?

Ans:

Array#compact removes nil values.
Example:
>> [nil,123,nil,"test"].compact
=> [123, "test"]

View All Answers

Question - 27:

Do you know how Ruby looks up a method to invoke?

Ans:

Since Ruby is a pure object-oriented language, it's important to make sure your developer thoroughly understands how objects work. The first place that Ruby looks for a method is in the object's metaclass or eigenclass-the class that contains methods directly defined on the object.
If the method cannot be found in an object's metaclass, Ruby will then search for the method in the ancestors of an object's class. The list of ancestors for any class starts with the class of the object itself, and climbs parent classes until it reaches the Object, Kernel, and BasicObject classes at the top of the Ruby class hierarchy.
If Ruby cannot find the method, it will internally send another method aptly called "method_missing?" to the object class. Ruby will repeat another search for this method, and will at least find it in the object class, provided the programmer did not see fit to define the "method_missing?" class earlier in the ancestry of the object.

View All Answers

Question - 28:

What is a class library in Ruby?

Ans:

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

View All Answers

Question - 29:

Please explain what are the positive aspects of Rails?

Ans:

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.

View All Answers

Question - 30:

Do you know what is Cross-Site Request Forgery (CSRF) and how Rails is protected against it?

Ans:

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.

View All Answers

Question - 31:

Tell me how you define Instance Variable, Global Variable and Class Variable in Ruby?

Ans:

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

View All Answers

Question - 32:

Tell me 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?

Ans:

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.

View All Answers

Question - 33:

Please explain the role of thread pooling in relation to the thread lifecycle in Ruby?

Ans:

In Ruby, the lifecycle of a single thread starts automatically as soon as CPU resources are available. The thread runs the code in the block where it was instantiated and obtains the value of the last expression in that block and returns it upon completion. Threads use up resources, but running multiple threads at a time can improve an app's performance.
Thread pooling is a technique wherein multiple pre-instantiated reusable threads are left on standby, ready to perform work when needed. Thread pooling is best used when there are a large number of short tasks that must be performed. This avoids the overhead of having to create a new thread every time a small task is about to be performed.

View All Answers

Question - 34:

Explain me what is the difference between concern and application_controller.rb in ruby on rails?

Ans:

Concerns are basically modules that get mixed into controller or model classes for instance. A Concern extends ActiveSupport::Concern module. It helps slim down the model/controller classes, and makes it easier to reuse common code across multiple model/controller classes. Concerns are like helper modules. We can define helper methods in concerns and can include or extend those concerns from various controller/model classes to sharing and DRY up the model/controller code.
On the other hand, ApplicationController is a controller class (NOT a module) and in your Rails application, it is the base class from which all of your other controllers classes are derived from. e.g.

```
class FooController < ApplicationController
  . . .
end
class BarController < ApplicationController
  . . .
end
```

In, Model-View-Controller (MVC) software architectural pattern, a controller implements business logics, can send commands to the model to update the model's state. It can also send commands to its associated view to change the view's presentation of the model. ApplicationController is that kind where as the Concerns are like helpers to get their job done.
Additional - Using concerns/services you can put your business logic in the controller; as a controller should not be business logic heavy, its main work is to get the request and get the response out.

View All Answers

Question - 35:

Tell me how would you freeze an object in Ruby? Can you provide a simple example?

Ans:

Sometimes it can be useful to prevent an object from being changed. This can be accomplished using the freeze method (Object.freeze) as in the sample code below.

```
water.freeze
if( water.frozen? )
  puts "Water object is a frozen object"
else
  puts "Water object is a normal object"
end
```

View All Answers

Question - 36:
What is the difference in scope for these two variables: @@name and @name?

Ans:
The difference in scope for these two variables is that:
* @@name is a class variable
* @name is an instance variable

View All Answers

Question - 37:
Do you know what is the defined operator?

Ans:
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.

View All Answers

Question - 38:
Do you know what is sweeper in Rails?

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

View All Answers

Question - 39:
Tell me how you can create a controller for subject?

Ans:
To create a controller for subject you can use the following command
C:rubylibrary> ruby script/generate controller subject

View All Answers

Question - 40:
Do you know what is "Yield" in Ruby on Rails?

Ans:
A Ruby method that receives a code block invokes it by calling it with the "Yield".

View All Answers

Question - 41:
Tell me what does ruby name refers to?

Ans:
* Ruby names refer to the classes, variables, methods, constants and modules that help in the creation of the program.
* These are the names that distinguish other names with the specified ruby names that are being mentioned above.
* Ruby names are useful and used to be written as it is shown. They are useful in the context of writing the programs and while generating the module.
* Ruby names refer to the classes that has to be built and through which the objects will be declared, and methods that will perform on the data that is given.
* The names are reserved such that they can't be used anywhere else and for any other purpose. The name that is used can be in lower or upper case, letter, number or an underscore

View All Answers

Question - 42:
Tell me what is the difference between Ruby 1.9 and Ruby 2.0?

Ans:
Ruby 2.0 differs in various ways, such as:
FASTER
Ruby 2 has a few patches that dramatically improve performance. The biggest of these is a substantial optimization to Kernel#require, which speeds up Rails startup dramatically. Ruby 2 also has improved GC, VM optimizations, and improvements to floating point operations.
UTF-8 BY DEFAULT
All Ruby scripts now default to UTF-8 encoding. This makes the #encoding: utf-8 magic comment no longer necessary.
LITERAL SYMBOL ARRAY SYNTAX
You can now use %i and %I to make an array of symbols:
LAZY ENUMERATORS
Ruby 2 introduces the concept of a lazy Enumerable object. A lazy Enumerable will not consume excess memory, and will process individual elements as needed instead of the entire array/range passed to it.

View All Answers

Question - 43:
Tell me what are blocks and procs?

Ans:

A block is basically Ruby's version of a closure-a block of code that can be wrapped up into a proc (a type of function) that can then be stored in a variable or passed to a method and run when desired. Blocks can syntactically be written as blocks of code between { } or the do and end keywords. The standard way to create a proc is depicted in the code block below.
> my_proc = Proc.new { |arg1| print "#{arg1}! " }

View All Answers

Question - 44:

Please explain what are some advantages of using Ruby?

Ans:

You want a programmer who can really play to the strengths of the Ruby programming language. Here are some of the key advantages of this language:
* Pure Object-Oriented Language: Everything in Ruby is an object-even methods, classes, and booleans. This greatly simplifies things from the coder's perspective and opens up a range of possibilities.
* Open-Source: Ruby is 100% free and open-source, with a large and enthusiastic community that can be tapped into as a resource.
* Metaprogramming: Ruby is widely considered to be one of the best programming languages out there for metaprogramming, or the ability to write code that can act on other code instead of data.
* Clean and Simple Syntax: The syntax is simple and concise, which allows developers to solve complex programs with fewer lines of code. It also helps that the code is human readable, and easy to follow.

View All Answers

Question - 45:

Tell us in Ruby, it explains about the defined operator?

Ans:

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.

View All Answers

Question - 46:

Explain me what are the limits of Ruby on Rails?

Ans:

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

View All Answers

Question - 47:

Tell me what is the difference between the Observers and Callbacks in Ruby on Rails?

Ans:

* 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

View All Answers

Question - 48:

Can you explain me how Rails implements Ajax?

Ans:

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

View All Answers

Question - 49:

Explain me what is the naming convention in Rails?

Ans:

* 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.

View All Answers

Question - 50:

Explain me what is the difference between Symbol and String?

Ans:

* The symbol in Ruby on rails act the same way as the string but the difference is in their behaviors that are opposite to each other.
* The difference remains in the object_id, memory and process time for both of them when used together at one time.
* Strings are considered as mutable objects. Whereas, symbols, belongs to the category of immutable.
* Strings objects are mutable so that it takes only the assignments to change the object information. Whereas, information of, immutable objects gets overwritten.
* String objects are written like
p: "string object jack".object_id #=>2250 or
p: "string object jack".to_sym.object_id #=> 2260, and
p: "string object jack". to_s_object_id #=> 2270
* Symbols are used to show the values for the actions like equality or non-equality to test the symbols faster then the string values.

View All Answers

Question - 51:

Please explain each of the following operators and how and when they should be used: ==, ===, eql?, equal?

Ans:

== - Checks if the value of two operands are equal (often overridden to provide a class-specific definition of equality).
=== - Specifically used to test equality within the when clause of a case statement (also often overridden to provide meaningful class-specific semantics in case statements).
eql? - Checks if the value and type of two operands are the same (as opposed to the == operator which compares values but ignores types). For example, 1 == 1.0 evaluates to true, whereas 1.eql?(1.0) evaluates to false.
equal? - Compares the identity of two objects; i.e., returns true iff both operands have the same object id (i.e., if they both refer to the same object). Note that this will return false when comparing two identical copies of the same object.

View All Answers

Question - 52:

Tell me an array [1,2,34,5,6,7,8,9], product it using a method?

Ans:

```
def productarray()
 return array.inject(:*)
end
```

View All Answers

Question - 53:

Please explain what is MVC and why do we use it?

Ans:

MVC stands for Model View Controller.
Model is the data structure that your program uses.
The View is the part that interacts with the screen or the next level up.
The Controller generally processes data between the model and view.
For example, a model will define your table in the database, the controller will dictate what to do behind the scenes and what to display in the view next. The view part generates the user interface which presents data to the user.
Let's take an example of creating a form. The submit form will be a view file, from that view file, when you click on the submit button it goes to the respective controller where you have written your business code about what happens on submitting the application/form, and from that controller the data will be saved in the database.
Thus MVC helps to split up the the business logic, database access and presentation. This is very useful in most web applications, and now lately into software/desktop applications.

View All Answers

Question - 54:

As you know ruby provides four types of variables. List them and provide a brief explanation for each?

Ans:

The four types of variables in Ruby are as follows:
* Global variables begin with $ and are accessible from anywhere within the Ruby program regardless of where they are declared-it stands to reason that they must be handled with care.
* Local variables begin with a lowercase letter or an underscore. The scope of a local variable is confined to the code construct within which it is declared.
* Class variables begin with @@ and are shared by all instances of the class that it is defined in.
* Instance variables begin with @ and are similar to class variables except that they are local to a single instance of a class in which they are instantiated.

View All Answers

Question - 55:

What is the Notation used for denoting class variables in Ruby?

Ans:

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

View All Answers

Question - 56:

Tell me what is the difference between calling super() and super call?

Ans:

* 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.

View All Answers

Question - 57:

Explain how you can run Rails application without creating databases?

Ans:

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]

View All Answers

Question - 58:

Tell me what is the command to create a migration?

Ans:

To create migration command includes
C:rubyapplication>ruby script/generate migration table_name

View All Answers

Question - 59:

Tell me what is the role of sub-directory app/controllers and app/helpers?

Ans:

* 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.

View All Answers

Question - 60:

Tell me what is Ruby on Rails?

Ans:

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

View All Answers

Question - 61:

Explain me what the difference is between false and nil in Ruby?

Ans:

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

View All Answers

Question - 62:

Tell me how Symbol is different from variables?

Ans:

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

View All Answers

Question - 63:

Can you list out what can Rails Migration do?

Ans:

Rails Migration can do following things

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

View All Answers

Question - 64:

Tell me what is the difference between Active support's "HashWithIndifferent" and Ruby's "Hash" ?

Ans:

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.

View All Answers

Question - 65:

Do you know what is rake in Rails?

Ans:

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.

View All Answers

Question - 66:

Tell us what is the difference between Dynamic and Static Scaffolding?

Ans:

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

View All Answers

Question - 67:

Tell me how can we define Ruby regular expressions?

Ans:

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.

View All Answers

Question - 68:

What is the difference between put and putc statement?

Ans:

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.

View All Answers

Question - 69:

What is the syntax for Ruby collect Iterator?

Ans:

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

View All Answers

Question - 70:

What is the difference extend and include?

Ans:

The "include" makes the module's methods available to the instance of a class, while "extend" makes these methods available to the class itself.

View All Answers

Question - 71:

Tell me the role of modules and mixins in Ruby?

Ans:

Modules are Ruby's way of grouping methods, classes, and constants together to provide a namespace for preventing name clashes. The second purpose of modules is to use them as mixins. Technically, Ruby only supports single inheritance, but by using modules as mixins, it is possible to share code among different classes-a key advantage of multiple inheritance-without having to give up the simplicity of the single inheritance paradigm.

View All Answers

Question - 72:

Explain me what is the function of ORM in Ruby on Rails?

Ans:

ORM stands for Object Relationship Model that models the classes and helps in setting a relationship between the existing models.
Functions:
1. It allows the classes to be mapped to the table that is present in the database and objects get mapped to the rows in database table.
2. It shows the relationship that exists between the object and frame it using the model to display the output to the users.
3. It keeps the data in the database according to its relationships and performs the functions accordingly.

View All Answers

Question - 73:

Tell me can you call a private method outside a Ruby class using its object?

Ans:

Yes, with the help of the send method.
Given the class Test:

```
class Test
   private
       def method
         p "I am a private method"
      end
end
```

We can execute the private method using send:

```
>> Test.new.send(:method)
"I am a private method"
```

View All Answers

Question - 74:

Tell us what is class libraries in Ruby?

Ans:

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

View All Answers

Web Development Most Popular & Related Interview Guides

1 : **Bootstrap Interview Questions and Answers.**

2 : Front End Web Developer Interview Questions and Answers.

3 : Typography Interview Questions and Answers.

4 : Junior Developer PHP Interview Questions and Answers.

5 : Laravel PHP Developer Interview Questions and Answers.

6 : HTML Developer Interview Questions and Answers.

7 : WordPress Themes Developer Interview Questions and Answers.

8 : Python Developer Interview Questions and Answers.

9 : Web Development Ninjas Interview Questions and Answers.

10 : Website Developer Interview Questions and Answers.

Follow us on FaceBook
www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter
https://twitter.com/InterviewQA

For any inquiry please do not hesitate to contact us.

Interview Questions Answers.ORG Team
https://InterviewQuestionsAnswers.ORG/
support@InterviewQuestionsAnswers.ORG