**Ruby on Rails**

* Introduction:-

**Ruby on Rails**, or **Rails**, is a [server-side](https://en.wikipedia.org/wiki/Server-side) [web application framework](https://en.wikipedia.org/wiki/Web_application_framework) written in [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)) under the [MIT License](https://en.wikipedia.org/wiki/MIT_License). Rails is a [model–view–controller](https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller) (MVC) framework, providing default structures for a [database](https://en.wikipedia.org/wiki/Database), a [web service](https://en.wikipedia.org/wiki/Web_service), and [web pages](https://en.wikipedia.org/wiki/Web_page). It encourages and facilitates the use of [web standards](https://en.wikipedia.org/wiki/Web_standards) such as [JSON](https://en.wikipedia.org/wiki/JSON) or [XML](https://en.wikipedia.org/wiki/XML) for data transfer and [HTML](https://en.wikipedia.org/wiki/HTML), [CSS](https://en.wikipedia.org/wiki/CSS) and [JavaScript](https://en.wikipedia.org/wiki/JavaScript) for user interfacing. In addition to MVC, Rails emphasizes the use of other well-known [software engineering](https://en.wikipedia.org/wiki/Software_engineering) [patterns](https://en.wikipedia.org/wiki/Software_design_pattern) and paradigms, including [convention over configuration](https://en.wikipedia.org/wiki/Convention_over_configuration) (CoC), [don't repeat yourself](https://en.wikipedia.org/wiki/Don%27t_repeat_yourself) (DRY), and the [active record pattern](https://en.wikipedia.org/wiki/Active_record_pattern).

Ruby on Rails' emergence in 2005 greatly influenced web app development, through innovative features such as seamless database [table](https://en.wikipedia.org/wiki/Table_(database)) creations, [migrations](https://en.wikipedia.org/wiki/Schema_migration), and [scaffolding](https://en.wikipedia.org/wiki/Scaffold_(programming)) of views to enable rapid application development. Ruby on Rails' influence on other web frameworks remains apparent today, with many frameworks in other languages borrowing its ideas, including [Django](https://en.wikipedia.org/wiki/Django_(web_framework)) in [Python](https://en.wikipedia.org/wiki/Python_(programming_language)), [Catalyst](https://en.wikipedia.org/wiki/Catalyst_(software)) in [Perl](https://en.wikipedia.org/wiki/Perl), [Laravel](https://en.wikipedia.org/wiki/Laravel), [CakePHP](https://en.wikipedia.org/wiki/CakePHP) and [Yii](https://en.wikipedia.org/wiki/Yii) in [PHP](https://en.wikipedia.org/wiki/PHP), [Grails](https://en.wikipedia.org/wiki/Grails_(framework)) in [Groovy](https://en.wikipedia.org/wiki/Apache_Groovy), [Phoenix](https://en.wikipedia.org/wiki/Phoenix_(web_framework)) in [Elixir](https://en.wikipedia.org/wiki/Elixir_(programming_language)), [Play](https://en.wikipedia.org/wiki/Play_Framework) in [Scala](https://en.wikipedia.org/wiki/Scala_(programming_language)), and [Sails.js](https://en.wikipedia.org/wiki/Sails.js) in [Node.js](https://en.wikipedia.org/wiki/Node.js).

Some well known sites that use Ruby on Rails include [Airbnb](https://en.wikipedia.org/wiki/Airbnb), [Crunchbase](https://en.wikipedia.org/wiki/Crunchbase), [Bloomberg](https://en.wikipedia.org/wiki/Bloomberg_L.P.), and [Dribbble](https://en.wikipedia.org/wiki/Dribbble).

* Technical overview[[edit](https://en.wikipedia.org/w/index.php?title=Ruby_on_Rails&action=edit&section=2)]

Like other web frameworks, Ruby on Rails uses the [model–view–controller](https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller) (MVC) pattern to organize application programming.

In a default configuration, a *model* in the Ruby on Rails framework maps to a table in a database and to a Ruby file. For example, a model class *User* will usually be defined in the file 'user.rb' in the app/models directory, and linked to the table 'users' in the database. While developers are free to ignore this convention and choose differing names for their models, files, and database table, this is not common practice and is usually discouraged in accordance with the "[convention-over-configuration](https://en.wikipedia.org/wiki/Convention_over_configuration)" philosophy.

A *controller* is a server-side component of Rails that responds to external requests from the web server to the application, by determining which view file to render. The controller may also have to query one or more models for information and pass these on to the view. For example, in an airline reservation system, a controller implementing a flight-search function would need to query a model representing individual flights to find flights matching the search, and might also need to query models representing airports and airlines to find related secondary data. The controller might then pass some subset of the flight data to the corresponding view, which would contain a mixture of static HTML and logic that use the flight data to create an HTML document containing a table with one row per flight. A controller may provide one or more actions. In Ruby on Rails, an action is typically a basic unit that describes how to respond to a specific external web-browser request. Also, note that the controller/action will be accessible for external web requests only if a corresponding route is mapped to it. Rails encourages developers to use [RESTful](https://en.wikipedia.org/wiki/Representational_state_transfer) routes, which include actions such as create, new, edit, update, destroy, show, and index. These mappings of incoming requests/routes to controller actions can be easily set up in the routes.rb configuration file.

A *view* in the default configuration of Rails is an [erb](https://en.wikipedia.org/wiki/ERB-Templating" \o "ERB-Templating) file, which is evaluated and converted to [HTML](https://en.wikipedia.org/wiki/HTML) at run-time. Alternatively, many other templating systems can be used for views.

Ruby on Rails includes tools that make common development tasks easier "out-of-the-box", such as [scaffolding](https://en.wikipedia.org/wiki/Scaffold_(programming)) that can automatically construct some of the models and views needed for a basic [website](https://en.wikipedia.org/wiki/Website).[[39]](https://en.wikipedia.org/wiki/Ruby_on_Rails#cite_note-39) Also included are [WEBrick](https://en.wikipedia.org/wiki/WEBrick" \o "WEBrick), a simple Ruby web server that is distributed with Ruby, and [Rake](https://en.wikipedia.org/wiki/Rake_(software)), a build system, distributed as a [gem](https://en.wikipedia.org/wiki/RubyGems). Together with Ruby on Rails, these tools provide a basic development environment.

Ruby on Rails is most commonly not connected to the Internet directly, but through some front-end [web server](https://en.wikipedia.org/wiki/Web_server). [Mongrel](https://en.wikipedia.org/wiki/Mongrel_(web_server)) was generally preferred[[*by whom?*](https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Words_to_watch#Unsupported_attributions)] over WEBrick in the early days] but it can also run on [Lighttpd](https://en.wikipedia.org/wiki/Lighttpd" \o "Lighttpd), [Apache](https://en.wikipedia.org/wiki/Apache_(web_server)), [Cherokee](https://en.wikipedia.org/wiki/Cherokee_(Webserver)), [Hiawatha](https://en.wikipedia.org/wiki/Hiawatha_(web_server)), [Nginx](https://en.wikipedia.org/wiki/Nginx) (either as a module – [Phusion Passenger](https://en.wikipedia.org/wiki/Phusion_Passenger) for example – or via [CGI](https://en.wikipedia.org/wiki/Common_Gateway_Interface), [FastCGI](https://en.wikipedia.org/wiki/FastCGI" \o "FastCGI) or [mod ruby](https://en.wikipedia.org/wiki/Mod_ruby)), and many others. From 2008 onward, Passenger replaced Mongrel as the most-used web server for Ruby on Rails.[[40]](https://en.wikipedia.org/wiki/Ruby_on_Rails#cite_note-40) Ruby is also supported natively on [IBM i](https://en.wikipedia.org/wiki/IBM_i).[[41]](https://en.wikipedia.org/wiki/Ruby_on_Rails#cite_note-41)

Ruby on Rails is also noteworthy for its extensive use of the [JavaScript](https://en.wikipedia.org/wiki/JavaScript) libraries [Prototype](https://en.wikipedia.org/wiki/Prototype_JavaScript_Framework) and [Script.aculo.us](https://en.wikipedia.org/wiki/Script.aculo.us) for scripting [Ajax](https://en.wikipedia.org/wiki/Ajax_(programming)) actions.[[42]](https://en.wikipedia.org/wiki/Ruby_on_Rails#cite_note-42) Ruby on Rails initially utilized lightweight [SOAP](https://en.wikipedia.org/wiki/SOAP) for web services; this was later replaced by RESTful [web services](https://en.wikipedia.org/wiki/Web_services). Ruby on Rails 3.0 separates the markup of the page (which defines the structure of the page) from scripting (which determines functionality or logic of the page). jQuery is fully supported as a replacement for Prototype and is the default JavaScript library in Rails 3.1, reflecting an industry-wide move towards jQuery. Additionally, [CoffeeScript](https://en.wikipedia.org/wiki/CoffeeScript" \o "CoffeeScript) was introduced in Rails 3.1 as the default JavaScript language.

Since version 2.0, Ruby on Rails offers both HTML and XML as standard output formats. The latter is the facility for RESTful web services.

Rails 3.1 introduced [Sass](https://en.wikipedia.org/wiki/Sass_(stylesheet_language)) as standard [CSS](https://en.wikipedia.org/wiki/CSS) templating.

By default, the server uses [Embedded Ruby](https://en.wikipedia.org/wiki/ERuby) in the HTML views, with files having an html.erb extension. Rails supports swapping-in alternative templating languages, such as [HAML](https://en.wikipedia.org/wiki/HAML) and [Mustache](https://en.wikipedia.org/wiki/Mustache_(template_system)" \o "Mustache (template system)).

Ruby on Rails 3.0 has been designed to work with Ruby 1.8.7, Ruby 1.9.2, and [JRuby](https://en.wikipedia.org/wiki/JRuby" \o "JRuby) 1.5.2+; earlier versions are not supported.[[43]](https://en.wikipedia.org/wiki/Ruby_on_Rails#cite_note-43)

Ruby on Rails 3.2 is the last series of releases that support Ruby 1.8.7.

* Framework structure:-

Ruby on Rails is separated into various packages, namely [ActiveRecord](https://en.wikipedia.org/wiki/Active_record_pattern" \o "Active record pattern) (an [object-relational mapping](https://en.wikipedia.org/wiki/Object-relational_mapping) system for database access), Action Pack, Active Support and Action Mailer. Prior to version 2.0, Ruby on Rails also included the Action Web Service package that is now replaced by Active Resource. Apart from standard packages, developers can make [plugins](https://en.wikipedia.org/wiki/Plug-in_(computing)) to extend existing packages. Earlier Rails supported plugins within their own custom framework; version 3.2 deprecates these in favor of standard Ruby "gems".[[44]](https://en.wikipedia.org/wiki/Ruby_on_Rails#cite_note-Rails_3.2.0.rc2_has_been_released!-44)

* Deployment:-

Ruby on Rails is often installed using [RubyGems](https://en.wikipedia.org/wiki/RubyGems" \o "RubyGems), a package manager[[45]](https://en.wikipedia.org/wiki/Ruby_on_Rails#cite_note-45) which is included with current versions of Ruby. Many free [Unix-like systems](https://en.wikipedia.org/wiki/Unix-like_system) also support installation of Ruby on Rails and its dependencies through their native [package management system](https://en.wikipedia.org/wiki/Package_management_system).

Ruby on Rails is typically deployed with a database server such as [MySQL](https://en.wikipedia.org/wiki/MySQL) or [PostgreSQL](https://en.wikipedia.org/wiki/PostgreSQL), and a web server such as [Apache](https://en.wikipedia.org/wiki/Apache_HTTP_Server) running the [Phusion Passenger](https://en.wikipedia.org/wiki/Phusion_Passenger) module.

* Philosophy and design[[edit](https://en.wikipedia.org/w/index.php?title=Ruby_on_Rails&action=edit&section=5)]

Ruby on Rails is intended to emphasize [*Convention over Configuration*](https://en.wikipedia.org/wiki/Convention_over_Configuration) (CoC), and the [*Don't Repeat Yourself*](https://en.wikipedia.org/wiki/Don%27t_Repeat_Yourself) (DRY) principle.

[The Rails Doctrine](https://rubyonrails.org/doctrine/) is an enduring enabler that guides the philosophy, design, and implementation of the Ruby on Rails framework.

"Convention over Configuration" means a developer only needs to specify unconventional aspects of the application. For example, if there is a class *Sale* in the model, the corresponding table in the database is called *sales* by default. It is only if one deviates from this convention, such as calling the table "products sold", that the developer needs to write code regarding these names. Generally, Ruby on Rails conventions lead to less code and less repetition.[[46]](https://en.wikipedia.org/wiki/Ruby_on_Rails#cite_note-46)

"Don't repeat yourself" means that information is located in a single, unambiguous place. For example, using the [ActiveRecord](https://en.wikipedia.org/wiki/Active_record_pattern" \o "Active record pattern) module of Rails, the developer does not need to specify database column names in class definitions. Instead, Ruby on Rails can retrieve this information from the database based on the class name.

"Fat models, skinny controllers" means that most of the application logic should be placed within the model while leaving the controller as light as possible.