

# Introduction to Ruby on Rails

# Ruby on Rails

- **Ruby on Rails** (RoR or Rails) is a web framework written in Ruby.
- RoR was developed by the Danish developer **David Heinemeier Hansson**, and its current **version** is 7.0.
- It's a **server-side** framework, which implements the Model-View-Controller (**MVC**) pattern.
- RoR provides a **default directory structure** to manage the database, web pages and other assets.
- It relies on known **patterns**, such as, Convention over Configuration, Don't repeat yourself (DRY) and Active Record.
- RoR **influenced** many web frameworks, like Grails, Laravel, CakePHP AND Django.
- It is used in **sites** like GitHub, Twitch, Shopify and Airbnb.

# Setting up the environment

- **Ruby** should be installed (check using `ruby -v` in a CLI).
- A DBMS. **SQLite3** is recommended for learning and testing purposes.
  1. Download your OS's version from here: <https://www.sqlite.org/download.html>
  2. Follow the instructions for your OS found here: [https://www.tutorialspoint.com/sqlite/sqlite\\_installation.htm](https://www.tutorialspoint.com/sqlite/sqlite_installation.htm)
  3. In Windows, it will consist in:
    - Downloading SQLite DLL and command line tools.
    - Unzipping all of it in a newly created folder.
    - Adding this folder to the PATH environment variable, as you'll find out here: <https://www.architectryan.com/2018/03/17/add-to-the-path-on-windows-10/>
  4. You can check using `SQLite3 --version` in a CLI.
- Install RoR using `gem install rails` in a CLI. Then, you can check using `rails --version`

# Creating an app

- You can perform many tasks using the `rails` command and its several instructions.
- You can create a new app using:  

```
rails new [app_name]
```
- The command above will create a directory structure detailed here:  
[https://guides.rubyonrails.org/getting\\_started.html](https://guides.rubyonrails.org/getting_started.html)
- Once you are situated in the directory created, you can run the app using the command:
  - `ruby bin/rails server` in Windows.
  - `bin/rails server` otherwise.
- You can stop the server using *Control+C*.

# Basic Components in Ruby on Rails

- Basic **components** in Ruby on Rails are routes, controllers, actions and views.
- A **route** maps a request to a controller action. It's a rule written in a Ruby DSL.
- A **controller action** performs the necessary work to handle the request, it includes preparing data for a view. The controller is a Ruby class whose public methods are actions.
- A **view** prepares the resulting data in a desired format. Views are templates, written using HTML and embedded Ruby.

# Creating a “Hello, World”

1. Add a route to the `config/routes.rb` file:

```
get "/products", to: "products#index"
```

This means that `GET products` requests map to the `index` action within `ProductsController`.

2. Create the appropriate controller and view (but not the route) using the following command:

```
ruby bin/rails generate controller Products index --skip-routes
```

3. It will create, among other files, the controller (`article_controllers.rb`) and the view (`index.html.erb`).
4. You can invoke the view from the controller. If not, the framework will invoke a matching one. You can now edit both files.

# Useful Resources

- Getting Started with Rails:  
[https://guides.rubyonrails.org/getting\\_started.html](https://guides.rubyonrails.org/getting_started.html)
- Rails Routing from the Outside In:  
<https://guides.rubyonrails.org/routing.html>
- Learn Ruby on Rails - Full Course:  
[https://www.youtube.com/watch?v=fmyvWz5TUWg&ab\\_channel=freeCodeCamp.org](https://www.youtube.com/watch?v=fmyvWz5TUWg&ab_channel=freeCodeCamp.org)



# COLOMBIA

## DC

Thanks! Any question?

softserve