

## Guida Ruby on Rails: introduzione

Benvenuti! Oggi vi presento la guida dedicata a **Ruby on Rails!**

Nel corso di questo primo articolo affronteremo i temi fondamentali per approcciarci a questo framework: cos'è Ruby on Rails, perché scegliere di utilizzarlo e come configurare l'ambiente di lavoro nei tre principali sistemi operativi.

Senza ulteriori preamboli, iniziamo!

### Cos'è Ruby on Rails

Ruby on Rails, abbreviato spesso in RoR o Rails, è un framework open source scritto in Ruby e basato sul design pattern Model-View-Control.

Analizziamo questa prima affermazione:

- Cos'è un **framework**? Un framework è, in breve, un'architettura logica che facilita il lavoro del programmatore. Attraverso un insieme di librerie, il programmatore avrà, infatti, il solo compito di concentrarsi sul codice dell'applicazione stessa, evitando di ripetersi nella scrittura di funzioni già scritte e, invece, direttamente implementabili.
- Che vuol dire **Model-View-Control**? Sintetizzato nella sigla MVC, questo paradigma rappresenta un pattern architetturale basato sulla separazione dei ruoli fra tre differenti componenti, per l'appunto il model, la view e il control. Nel corso dei prossimi articoli, avremo modo di approfondire i vari ruoli dei componenti e l'interazione tra essi.

Bene. Ora che abbiamo la definizione di cosa è RoR, dobbiamo capire perché utilizzarlo, preferendolo ad altri!

### Perché scegliere Ruby on Rails

I maggiori vantaggi di RoR, riguardano la semplicità! Il linguaggio di scripting Ruby permette di esprimere idee, muovendosi all'interno di un codice pulito e facile da scrivere. La filosofia attorno alla quale ruota interamente RoR è riassumibile in due concetti chiave:

- **DRY** - Queste tre lettere sono, in realtà, l'acronimo di *Don't Repeat Yourself*. Chi conosce l'inglese, avrà intuito che ciò significa, nella pratica, l'assenza di ripetizioni all'interno del codice. Le definizioni necessitano di essere scritte un'unica volta!
- **Convention Over Configuration** - Ovvero, se rispetti le convenzioni standard di Rails, non sono necessarie configurazioni aggiuntive.

Inoltre Rails è agile. Cosa vuol dire agile? Semplice! Vuol dire che Rails segue le linee guida del [Manifesto per lo sviluppo agile](#) sintetizzabili in:

1. I singoli e le interazioni prevalgono sui processi e gli strumenti;
2. La collaborazione dei clienti prevale sulla negoziazione contrattuale;
3. L'adattamento alle modifiche prevale sull'adesione a un piano;
4. Il software funzionante prevale sulla documentazione estesa;

Nella pratica i clienti e gli sviluppatori della comunità hanno più controllo sul software in corso di sviluppo e ciò rende possibile una maggiore reattività ai cambiamenti, con modifiche al codice più veloci ed efficaci. Inoltre, pur non disprezzando la comune documentazione, Rails è avvalorato da un'ampia comunità online che generalmente è in grado di soddisfare quasi ogni richiesta di supporto. Ecco i principali link delle comunità:

- [Google Groups](#)
- [Ruby on Rails community](#)
- [Rails Forum](#)

In sintesi creare complesse applicazioni come blog, e-commerce e social network diventa un compito meno arduo del previsto con Ruby on Rails. Tutto, infatti, ruota intorno al concetto fondamentale di **gems**, ovvero di librerie sempre ben documentate e facilmente implementabili e personalizzabili, attraverso le quali diventa semplice creare funzioni che altrimenti richiederebbero giorni di sviluppo.

## Installazione di Rails

Le seguenti istruzioni fanno ovviamente riferimento alle ultime versioni degli OS in questione al momento della scrittura dell'articolo. Se desiderate ricevere istruzioni per differenti sistemi, oppure tramite metodi differenti, potete visitare il sito [GoRails](#).

### Windows

L'installazione su macchine Windows è, senza dubbio, la più semplice delle tre. Vi basterà effettuare il download di **RailsInstaller** dal [sito ufficiale](#) e seguire la procedura guidata che vi permetterà, volendo, di installare anche un sistema di Version Control come **Git**.

### Mac OS X 10.9

Come prima cosa, avete bisogno di installare **Xcode**. Potete facilmente effettuare il download attraverso l'AppStore. Una volta completata la procedura, aprite il sito dedicato agli sviluppatori Apple, effettuate l'accesso e scaricate il pacchetto **Command Line Tools** per la vostra versione di Xcode. Installato il pacchetto, aprite il terminale e installate la versione development di **RVM**

usando il seguente comando:

Appena completato il processo, aggiornate RVM all'ultima versione e installate **Ruby**:

Infine digitate, sempre nel terminale, il seguente comando per installare l'ultima versione di **Rails**:

## Linux Ubuntu 14.04

Il primo passo è quello di aggiornare alcune **dipendenze** necessarie a Ruby. Dal terminale, utilizzate il seguente comando:

Una volta completato, potete passare all'installazione di **RVM**:

Procedete con l'installare **NodeJS**, indispensabile per gestire i file CoffeeScripts:

Finalmente, come ultimo passo, installate **Rails** e controllate l'avvenuta configurazione:

## Conclusioni

Dalle procedure d'installazione avrete capito che è richiesta una certa familiarità con la shell di comando del vostro sistema operativo, sia esso il terminale di Mac OS X o il cmd di Windows. Nel corso della guida utilizzeremo numerose volte la shell, quindi, nel caso non l'abbiate mai usata prima, vi consiglio di imparare per lo meno i comandi base come, ad esempio, spostarsi da una cartella all'altra, creare e copiare file, etc.

Inoltre, prima di proseguire nella lettura dei futuri articoli, vi faccio presente che sono richiesti, come prerequisiti, la conoscenza perlomeno basilare di HTML, CSS e JavaScript. In poche parole, dovete aver chiaro com'è costituita una pagina web.

Nel corso della [prossima lezione](#) ci occuperemo di configurare l'ambiente di lavoro, scegliendo quindi un editor e un sistema di versioning, e dando vita alla nostra prima applicazione!

## GUIDA RUBY ON RAILS: INDICE LEZIONI

- 1) Introduzione
- 2) [L'ambiente di lavoro e la nostra prima app](#)

- 3) [Un assaggio di dinamicità](#)
- 4) [Architettura di un'applicazione](#)
- 5) [Il linguaggio Ruby - Parte 1](#)
- 6) [Il linguaggio Ruby - Parte 2](#)
- 7) [Creiamo c-Bookcase](#)
- 8) [Convalida e test](#)
- 9) [Ruby on Rails: page layout](#)
- 10) [Guida Ruby on Rails: creiamo il carrello](#)