

irb Examples:

```
value = 1
value.class
value = "string"
value.class
```

```
hash = {:key1=>'value1', :key2 => 'value2', :key3 => 'value3'}
hash.class
new_hash = hash.reject{|key,value| key == :key1 || value == 'value3'}
hash.reject!{|key,value| key == :key1 || value == 'value3'}
hash
require 'pp'
hash.methods.sort
```

```
-----
# app/models/post.rb
class Post < ActiveRecord::Base
  validates_presence_of :title, :body
  validates_uniqueness_of :title
end
-----
# app/controllers/posts_controller.rb
class PostsController < ApplicationController
  def new
    @post = Post.new
  end
end
```

```
-----
<%-
# app/views/posts/new.html.erb
-%>

<h2>write a new blog post</h2>

<%= error_messages_for 'post' %>

  <% form_for @post do |post_form| %>
  <p>
    <strong>Title:</strong>
    <%= post_form.text_field :title %>
  </p>
  <p>
    <strong>Body:</strong><br />
    <%= post_form.text_area :body %>
  </p>

  <%= post_form.submit 'Save post' %>
  <% end %>
```

```
-----
# app/controllers/posts_controller.rb
# (just the create action)
def create
  @post = Post.new(params[:post])

  if @post.save
    flash[:notice] = 'Your post was saved successfully.'
    redirect_to @post
```

```

else
  render :action => 'new'
end
end
-----
# app/controllers/posts_controller.rb
def show
  @post = Post.find(params[:id])
end
-----

<%-
# app/views/posts/show.html.erb
-%>

<h2><%= @post.title %></h2>
Posted at: <%= @post.created_at %><br />
<%= @post.body %>
-----
# app/controllers/posts_controller.rb
def create
  respond_to do |format|
    format.xml {
      # process xml here
      render :xml => @post.to_xml
    }
    format.json {
      # process json request here
      render :json => @post.to_json
    }
    format.html {
      @post = Post.new(params[:post])
      if @post.save
        flash[:notice] = 'Your post was saved successfully.'
        redirect_to @post
      else
        render :action => 'new'
      end
    }
  end
end
-----
script/generate migration add_author_to_post author:string
-----

<%-
# app/views/posts/new.html.erb
-%>

<h2>write a new blog post</h2>

<%= error_messages_for 'post' %>

  <% form_for @post do |post_form| %>
  <p>
    <strong>Author:</strong>
    <%= post_form.text_field :author %>
  </p>

  <p>
    <strong>Title:</strong>
    <%= post_form.text_field :title %>

```

ruby-code-snippets.txt

```
</p>
<p>
  <strong>Body:</strong><br />
  <%= post_form.text_area :body %>
</p>
```

```
<%= post_form.submit 'Save post' %>
<% end %>
```

```
-----
<%-
# app/views/posts/show.html.erb
3. -%>
4.
5. <h2><%= @post.title %></h2>
6.
7. Posted at: <%= @post.created_at %> by <strong><%= @post.author %></strong><br
/>
8. <%= @post.body %>
```

```
def index
  @posts = Post.find(:all, :order => 'created_at DESC')
end
```

```
-----
<%-
# app/views/posts/_post.html.erb
-%>

<h2><%= link_to_unless_current post.title, post %></h2>
Posted at: <%= post.created_at %> by <strong><%= post.author %></strong><br />
<%= post.body %>
<hr />
```

```
-----
<%-
# app/views/posts/show.html.erb
-%>

<%= render :partial => 'post' %>
```

```
-----
<%-
# app/views/posts/index.html.erb
%>

<%= render :partial => 'post', :collection => @posts %>
```

```
-----
# config/routes.rb (excerpt!)
map.root :controller => 'posts'
```

```
-----
<%-
# app/views/layouts/posts.html.erb
-%>
```

ruby-code-snippets.txt

```
<html>
  <head>
    <title>My blog</title>
    <style type="text/css">
      body { font-family: Georgia, serif; }
      a { color: blue; }
    </style>
  </head>
  <body>
    <h1>My blog</h1>
    <% if flash[:notice] %>
      <div style="background-color: #c0c0c0">
        <%= flash[:notice] %>
      </div>
    <% end %>
    <%= yield %>
  </body>
</html>
```

script/generate resource comment post_id:integer author:string body:text

```
-----
# app/models/comment.rb
class Comment < ActiveRecord::Base
  belongs_to :post
  validates_presence_of :post, :author, :body
end
```

```
-----
# app/models/post.rb
class Post < ActiveRecord::Base
  validates_presence_of :title, :body
  validates_uniqueness_of :title
  has_many :comments
end
```

```
-----
# app/controllers/posts_controller.rb
# (just the show action)
def show
  @post = Post.find(params[:id])
  @comment = Comment.new
end
```

```
-----
<%-
# app/views/posts/show.html.erb
-%>

<%= render :partial => 'post' %>

<% unless @post.comments.blank? %>
  <h2>Comments</h2>
  <% @post.comments.each do |comment| %>
    By <strong><%= comment.author %></strong> on <%= comment.created_at %>:<br />
    <%= comment.body %>
  <hr />
  <% end %>
<% end %>

<h2>Add a comment</h2>
```

```

ruby-code-snippets.txt
<% form_for @comment do |comment_form| %>
  <%= hidden_field_tag :post_id, @post.id %>
  <p>
    <strong>Author:</strong>
    <%= comment_form.text_field :author %>
  </p>
  <p>
    <strong>Body:</strong><br />
    <%= comment_form.text_area :body %>
  </p>
  <%= comment_form.submit 'Save post' %>
<% end %>

```

```

-----
# app/controllers/comments_controller.rb
class CommentsController < ApplicationController
  def create
    @post = Post.find(params[:post_id])
    @comment = Comment.new(params[:comment])
    @comment.post = @post
    if @comment.save
      flash[:notice] = 'Thanks for your comment!'
    else
      flash[:notice] = 'Something went wrong while adding your comment'
    end
    redirect_to @post
  end
end

```

```

-----
script/plugin install http://github.com/thoughtbot/shoulda.git/ --force
-----

```

```

class PostTest < ActiveSupport::TestCase

  should_require_attributes :body
  should_require_unique_attributes :title
  should_have_many :comments

  def test_truth
    assert true
  end
end

```

```

-----
# test/unit/comment_test.rb

class CommentTest < ActiveSupport::TestCase
  # Replace this with your real tests.
  should_belong_to :post
  should_require_attributes :post,:author,:body

  def test_truth
    assert true
  end
end

```