**Render by Default:**

Controller

* organize the whole process
* hands heavy code to model
* response back to users through the view

routes file:

* determines which view to navigate to
* resources **:books**
* look for app/views/books/index.html.erb

action:

A picture containing knife, table

Description automatically generated

* index is an action in controller
* look for app/views/books/index.html.erb (action\_name.html.erb)

**Render by Default:**

* render with an action's view:

A picture containing bird

Description automatically generated

* render "edit" / render :edit will render edit.html.erb template belonging to the same controller
* :edit is a symbol
* render an action's view from another controller:

A close up of a logo

Description automatically generated

* + use full path relative to app/views
  + the template is at app/views/products/show
* all ways of render do the same thing:

A picture containing bird

Description automatically generated

* **render plain: "OK"** => renders text
* **render html: helpers.tag.strong('Not Found')** => renders html
* **render json: @product** => render JSON
* **render xml: @product**
* **render js: "alert('Hello Rails');"**

**Options for Render:**

A screenshot of a cell phone

Description automatically generated

Find Layouts:

* rails look for app/views/layoutswith the same base name as the controller
  + PhotosController => app/views/layouts/photos.html.erb

A picture containing knife

Description automatically generated

* can overwrite layout (layout "inventory")
  + all views for ProductsController will use app/views/layouts/inventory.html.erb for their layout

\*\*\* render :action doesn't run code in the target action

\*\*\* render “index” directly renders index.html.erb

\*\*\* render :action “index” requires an index action in controller

Using redirect\_to:

* handle returning to HTTP request
* tells browser to send a new request for a different URL
* vs.
* *render tells Rails which view to use in constructing a response*

A screenshot of a cell phone

Description automatically generated

**Structuring Layouts:**

* Rails renders a view as a response by combining the current layout with the view.
* **Asset Tag Helpers:**
  + Can link views to JavaScript, stylesheets, images…
* **Yielding:**
  + Defines a section where view should be inserted

<html>

<head>

</head>

<body>

<%= yield %>

</body>

</html>

* **Content\_for:**
  + Insert content into a name yield block
* **Partials:**
  + Move the code for rendering a particular piece of a response to its own file

<%= render "menu" %>

* + - Render ­\_menu.html.erb (partials are named with underscore, different from normal views)

<%= render partial: "link\_area", layout: "graybar" %>

* + - Render \_link\_area.html.erb partial with layout \_graybar.html.erb
    - Need to specify :partial when passing additional options