

INTRODUCTION TO RAMDA

Christine Legge

August 13, 2016

Open Source Hackathon, GitHub HQ, San Francisco

ZENHUB.COM

The screenshot displays the ZenHub web interface for the **ZenHubHQ / ZenHubHQ** repository. The interface is organized into several sections:

- Header:** Includes a search bar, repository name, and navigation links for Pull requests, Issues, Gist, and To Do.
- Repository Info:** Shows the repository name, privacy status (PRIVATE), and statistics (Unwatch, Star, Fork).
- Navigation:** Links to Code, Issues (26), Pull requests (1), Boards, Burndown, Wiki, Pulse, and Graphs.
- Board Controls:** Includes filters for Repos (1/1), Labels, Milestones, Assignees, and a search bar.
- Board Columns:** The board is divided into columns representing different stages of the workflow:
 - Icebox (3):** Contains issues like "Update social media tag styling" (Help Wanted, Invalid) and "Customer discovery survey" (Engineering, Question).
 - Product Backlog (4):** Contains issues like "Real-time data in dashboard" (0.5 Enhancement) and "Reporting suite research" (Help Wanted, Question).
 - Sprint Backlog (3):** Contains issues like "New features v3.0" (Filter by this epic) and "Additional meta-data for Website" (Discussion, Feature).
 - In Progress (5):** Contains issues like "Webinar offer" (Duplicate, Enhancement) and "Velocity Charts" (Feature).
 - Testing (1):** Contains the issue "Reporting and metrics" (Engineering, Feature).
 - Done (2):** Contains issues like "Tutorials" (Epic) and "Video ex" (Discussion).





“

A practical functional library for
Javascript programmers

-ramdajs.com



FUNCTIONAL PROGRAMMING

- first-class functions
- purity
- immutable data



FUNCTIONAL PROGRAMMING

- no side effects

not functional 😓	functional 😊
<pre>var a1 = 0; function increment () { a1 += 1; }</pre>	<pre>function increment (a2) { return a2 + 1; }</pre>



FUNCTIONAL PROGRAMMING

- Pure functions: same input → same output

not pure 🥵	pure 😊
<pre>var a1 = 0; function increment () { a1 += 1; }</pre>	<pre>function increment (a2) { return a2 + 1; }</pre>
<pre>increment (); // => a1 = 1 increment (); // => a1 = 2 increment (); // => a1 = 3</pre>	<pre>increment (0); // => 1 increment (0); // => 1 increment (0); // => 1</pre>

WHY RAMDA?

➤ Simple functions

native JavaScript (ES6) 🥵	with Ramda 😊
<pre>var myObj = { id: 1, isOn: false }; function enable(obj) { return Object.assign({}, obj, { isOn: true }); } var newObj = enable(myObj);</pre>	<pre>var myObj = { id: 1, isOn: false }; var enable = R.assoc('isOn', true); var newObj = enable(myObj);</pre>



WHY RAMDA?

- Integrates well with React/Redux based applications

native JavaScript (ES6) 🥵

```
function addToDo(state, action) {  
  
  var newTodo = {  
    text: action.text,  
    complete: false  
  };  
  
  var todos = [...state.todos, newTodo];  
  
  return Object.assign({}, state, {  
    todos: todos  
  });  
}
```

with Ramda 😊

```
function addToDo(state, action) {  
  
  var newTodo = {  
    text: action.text,  
    complete: false  
  };  
  
  var todos = R.append(newTodo, state.todos);  
  
  return R.assoc('todos', todos, state);  
}
```



WHY RAMDA?

- Clean and concise code



RESOURCES

<https://github.com/leggechr/intro-to-ramda>

- Mostly Adequate Guide to Functional Programming
- ramdajs.com
- What Ramda Function Should I Use?

