

Javascript

Javascript is an object-oriented language like Ruby, but requires different syntax.

```
function sayHello(name) {  
  alert("Hello, " + name);  
}
```

Like in Ruby, you can create arbitrary blocks of code by simply creating a function without a name:

```
function() {  
  alert("Wazzzzup!");  
end
```

JQuery Part 1

jQuery provides an easy way to manipulate your HTML after the browser has loaded it.

Here's how it works:

1. Write a function you want the browser to run when your page loads.
2. Pass your function to the document's **ready** method.
3. When the browser has finished loading the HTML and is ready to display it on the screen, your code block will execute.
4. Use jQuery's **selectors** and **object methods** to manipulate the DOM.

JQuery Part 2

Effective use of jQuery requires a decent understanding of **Javascript anonymous functions, selectors, and methods.**

Step 1: Use the .ready event:

```
$(document).ready()
```

which can be simplified as

```
$( )
```

Step 2: Provide your function

```
$( function() {  
    alert("Hello there!");  
} );
```

Using jQuery UI in Rails

3 Easy Steps

1. Load the jquery-ui javascript file from your application.css manifest:

```
//= require jquery-ui
```

2. You probably want to use the CSS custom-made for the jQuery UI widgets

You can link to Google's copy of the CSS:

<http://ajax.googleapis.com/ajax/libs/jqueryui/1.8.16/themes/redmond/jquery-ui.css>

3. Use the widget according to the documentation. For example:

```
$("#mytextbox").datepicker();
```

Ajax in Rails

3 Easy Steps

1. `:remote => true`

Add this option to any link or form to submit an AJAX request instead of a blocking request.

2. Enhance your `respond_to` block to accept JS requests.

3. Generate a Javascript response instead of HTML. You can use a view template if you want, i.e:

```
create.js.erb
```

NOTE: if you render a partial inside of your template, use the `escape_javascript` helper.