RJS Cheat Sheet - Code Examples

<<(javascript)

Writes raw JavaScript to the page.

[](id)

```
# Ruby code
page['header'].show
// Generated JavaScript
$("header").show();
# Ruby code
page['header'].first.second
// Generated JavaScript
$("header").first().second();
```

assign(variable, value)

```
# Ruby code
page.assign 'name', { :first => "Alex", :last =>
"Nesbitt" }
// Generated JavaScript
name = { "first": "Alex", "last": "Nesbitt" };
```

alert(message)

```
# Ruby code
page.alert 'An error occurred while processing your
request'
// Generated JavaScript
alert("An error occurred while processing your
request");
```

call(function, arg, ...)

```
# Ruby code
page.call 'displayError', 'An error occurred', 'Critical'
// Generated JavaScript
displayError("An error occurred", "Critical");
# Ruby code
page.call 'inventory.showTotal'
// Generated JavaScript
inventory.showTotal();
```

delay(seconds = 1)

```
# Ruby code
page.delay(5) do
page.visual_effect :highlight, 'navigation'
end
// Generated JavaScript
setTimeout(function() {
;
new Effect.Highlight("navigation", {});
}, 5000);
```

draggable(id, options = {})

```
# Ruby code
page.draggable('photo', :revert => true)
// Generated JavaScript
new Draggable('photo', {revert: true});
```

drop_receiving(id, options = {})

```
# Ruby code
page.drop_receiving('photo', :url => { :action => 'add'
})
// Generated JavaScript
Droppables.add("photo",
{onDrop:function(element){new
Ajax.Request('/hello_world/add', {asynchronous:true,
evalScripts:true,
parameters:'id=' +
encodeURIComponent(element.id)})}});
```

hide(id, ...)

```
# Ruby code
page.hide('first', 'second')
// Generated JavaScript
Element.hide("first", "second");
```

insert_html(position, id, *options_for_render)
Options - :before , :after, :top , :bottom

```
# Ruby code
page.insert_html(:bottom, 'products',
'<li>Refrigerator</li>')
// Generated JavaScript
new Insertion.Bottom("products",
"<li>Refrigerator</li>");
```

redirect_to(location)

```
# Ruby code
page.redirect_to('http://www.google.com')
// Generated JavaScript
window.location.href = "http://www.google.com";
# Ruby code
page.redirect_to(:controller => 'inventory', :action =>
'list')
// Generated JavaScript
window.location.href =
"http://localhost:3000/inventory/list";
```

remove(id, ...)

```
# Ruby code
page.remove('first', 'second')
// Generated JavaScript
["first", "second"].each(Element.remove);
```

replace(id, *options_for_render)

```
# Ruby code
product.replace 'banner', '<id="banner">Welcome back
Alex</div>'
// Generated JavaScript
Element.replace("banner", "<id=\"banner\">Welcome
back Alex</div>");
```

Source: RJS Templates for Rails

RJS Cheat Sheet - Code Examples

replace_html(id, *options_for_render)

```
# Ruby code
page.replace_html 'timestamp', Time.now
// Generated JavaScript
Element.update("timestamp", "Sat Apr 29 15:14:24
EDT 2006");
```

select(pattern)

```
# Ruby code
page.select "#content p"
// Generated JavaScript
# => $$("#content p");
```

sortable(id, options = {})

```
# Ruby code
# Assuming the current controller is ProjectsController
page.sortable 'project-65', :url => { :action => 'sort' }
// Generated JavaScript
Sortable.create("project-65", {onUpdate:function(){
new Ajax.Request('/projects/sort', {
asynchronous:true, evalScripts:true,
parameters:Sortable.serialize("project-65")
}
)}
});
```

show(id, ...)

```
# Ruby code
page.show 'element-20', 'element-30'
// Generated JavaScript
Element.show("element-20", "element-30");
```

toggle(id, ...)

```
# Ruby code
page.toggle 'product-1', 'product-2', 'product-3'
// Generated JavaScript
Element.toggle("product-1", "product-2", "product-3");
```

visual_effect(name, id = nil, options = {})

```
# Ruby code
page.visual_effect :highlight, 'list-item-69', :duration
=> 5
// Generated JavaScript
new Effect.Highlight("list-item-69",{duration:5});
appear, blind_down, blind_up, drop_out, fade, fold,
grow, highlight, puff, pulsate, shake, shrink,
slide_down, slide_up, squish, switch_off,
toggle_appear, toggle_blind, toggle_slide
```

reload

```
page['header'].reload
# Equivalent to:
page['header'].replace :partial => 'header'
```

all(variable) {|value,index| block }

```
# Ruby code
page.select('#line-items li').all('allVisible') do |value,
index|
value.visible
end
// Generated JavaScript
var allVisible = $$("#line-items li").all(function(value,
index) {
return value.visible();
});
```

any(variable){|value, index| block }

```
# Ruby code
page.select('#line-items li').any('anyVisible') do |value,
index|
value.visible
end
// Generated JavaScript
var anyVisible = $$("#line-items li").any(function(value,
index) {
return value.visible();
});
```

collect(variable){|value, index| block }

```
# Ruby code
page.select('#orders tr').collect('heights') do |value,
index|
value.get_height
end
// Generated JavaScript
var heights = $$("#orders tr").collect(function(value,
index) {
return value.getHeight();
});
```

detect(variable){|value, index| block }

```
# Ruby code
page.select('#content p').detect('first_visible') do
|value, index|
value.visible
end
// Generated JavaScript
var first_visible = $$("#content
p").collect(function(value, index) {
return value.visible();
});
```

each{|value, index| block }

```
# Ruby code
page.select('#content p').each do |value, index|
page << 'alert(value.innerHTML);'
end
// Generated JavaScript
$$("p").each(function(value, index) {
alert(value.innerHTML);
});
```

Source: RJS Templates for Rails

find(variable){|value, index| block }

see detect()

find_all(variable){|value, index| block }

```
# Ruby code
page.select('#content p').find_all('empty_paragraphs')
do |value, index|
value.empty
end
// Generated JavaScript
var empty_paragraphs = $$("#content
p").collect(function(value, index) {
return value.empty();
});
```

grep(variable, pattern){|value, index| block }

```
# Ruby code
page.select('#content p').grep('hidden',
/Div|Paragraph/) do |value, index|
value.hidden
end
// Generated JavaScript
var hidden = $$("#content p").grep(/Div|Paragraph/,
function(value, index) {
return value.hidden();
});
```

inject(variable, memo){|memo, value, index| block }

```
# Ruby code
page.select('#content .columns').inject('totalWidth', 0)
do |memo, value, index|
page << '(memo + value.offsetWidth)'
end
// Generated JavaScript
var totalWidth = $$(".columns").inject(0,
function(memo, value, index) {
return(memo + value.offsetWidth);
});
```

map(variable){|value, index| block }

see collect()

max(variable){|value, index| block }

```
# Ruby code
page.select('p').max('longestParagraphLength') do
|value, index|
page << 'value.innerHTML.length'
end
// Generated JavaScript
var longestParagraphLength =
$$("p").max(function(value, index) {
return value.innerHTML.length;
});
```

min(variable){|value, index| block }

```
# Ruby code
page.select('p').min('shortestParagraphLength') do
|value, index|
page << 'value.innerHTML.length'
end
// Generated JavaScript
var shortestParagraphLength =
$$("p").min(function(value, index) {
return value.innerHTML.length;
});
```

partition(variable){|value, index| block }

```
# Ruby code
page.select('.products').partition('productsByVisibility')
do |value, index|
value.visible
end
// Generated JavaScript
var productsByVisibility =
$$(".products").partition(function(value, index) {
return value.visible();
});
```

pluck(variable, property)

```
# Ruby code
page.select('.amounts').pluck('amounts', 'innerHTML')
// Generated JavaScript
var amounts = $$(".amounts").pluck("innerHTML");
```

reject(variable){|value, index| block }

```
# Ruby code
page.select('p').reject('paragraphsWithContent') do
|value, index|
value.empty
end
// Generated JavaScript
var paragraphsWithContent =
$$("p").reject(function(value, index) {
return value.empty();
});
```

select(variable){|value, index| block }

see find_all()

sort_by(variable){|value, index| block }

```
# Ruby code
page.select('p').sort_by('sortedParagraphs') do |value,
index|
page << 'value.innerHTML.length;'
end
// Generated JavaScript
var sortedParagraphs = $$("p").sortBy(function(value,
index) {
return value.innerHTML.length;
});
```

Source: RJS Templates for Rails

zip(variable, arg1, ...){|value, index| block }

assume that the page has the following paragraph elements:

```
<p id="first">First Paragraph</p>
<p id="second">Second Paragraph</p>
```

First, applying zip without a block:

```
# Ruby code
page.select('p').zip('zipped', [1, 2], [3,4])
// Generated JavaScript
var zipped = $$('p').zip([1,2], [3,4]);
```

The resulting array of arrays, where the paragraphs (represented by their ids) are DOM objects: [[< id="first">, 1, 3], [<p id="second" >, 2, 4]]

Applying zip with a block:

```
# Ruby code
page.select('p').zip('zipped', [1, 2], [3,4]) do |array|
page.call 'array.reverse'
end
// Generated JavaScript
var zipped = $$("p").zip([1,2], [3,4], function(array) {
return array.reverse();
});
```

The resulting Array of arrays, where the paragraphs (represented by their ids) are DOM objects: [[3, 1, <p id="first">], [4, 2, <p id="second" >]]

Source: RJS Templates for Rails