



write less.  
do more.

who are we?

Yehuda Katz  
Andy Delcambre



How is this going to work?

# Introduction to jQuery

# Event Driven JavaScript

# jQuery on Rails



# Labs!

```
git clone git://github.com/adelcambre/jquery-tutorial.git
```

# Introduction to jQuery

```
h1 {  
  color: #999;  
}
```

## UJS With jQuery

```
$(“h1”).click(function() {  
  $(this).fadeOut();  
});
```

Get Some Elements  
Do Stuff

get some elements.  
but how?

# CSS 3 plus

- div
- div#foo
- div.class
- div, p, a
- div p
- div > p
- div + p
- div ~ p
- div:first
- div:last
- div:not(#foo)
- div:even
- div:odd
- div:eq(1)
- div:gt(1)
- div:lt(1)
- div:header
- div:animated
- div:contains(txt)
- div:empty
- div:has(p)
- div:parent
- div:hidden
- div:visible

# CSS 3 plus

- div[foo]
- div[foo=bar]
- div[foo!=bar]
- div[foo^=bar]
- div[foo\$=bar]
- div[foo\*=bar]
- :nth-child(2)
- :nth-child(even)
- :first-child

- :last-child
- :only-child
- :input
- :text
- :password
- :radio
- :checkbox
- :submit
- :image

- :reset
- :button
- :file
- :hidden
- :enabled
- :disabled
- :checked
- :selected

get some elements.



```
$("table tr:nth-child(even)  
    > td:visible")
```

do stuff.

`$("div")`

RETURNS JQUERY OBJECT

`$("div").fadeIn()`

RETURNS JQUERY OBJECT

```
$(“div”).fadeIn()  
.css(“color”, “green”)
```

RETURNS JQUERY OBJECT

we call this chaining

# Crazy chains

```
$(“ul.open”) // [ ul, ul, ul ]  
  .children(“li”) // [ li, li, li ]  
    .addClass(“open”) // [ li, li, li ]  
  .end() // [ ul, ul, ul ]  
  .find(“a”) // [ a, a, a ]  
    .click(function(){  
      $(this).next().toggle();  
      return false;  
    }) // [ a, a, a ]  
  .end(); // [ ul, ul, ul ]
```

# Lab 1: Selectors

- Select every other row of the table
- Select the Checked checkboxes
- Select the first column of the table
- Select the top level of the outline
- Select any links to [jquery.com](http://jquery.com)
- Select any images that contain flowers



# 5 parts of jquery

DOM

EVENTS

EFFECTS

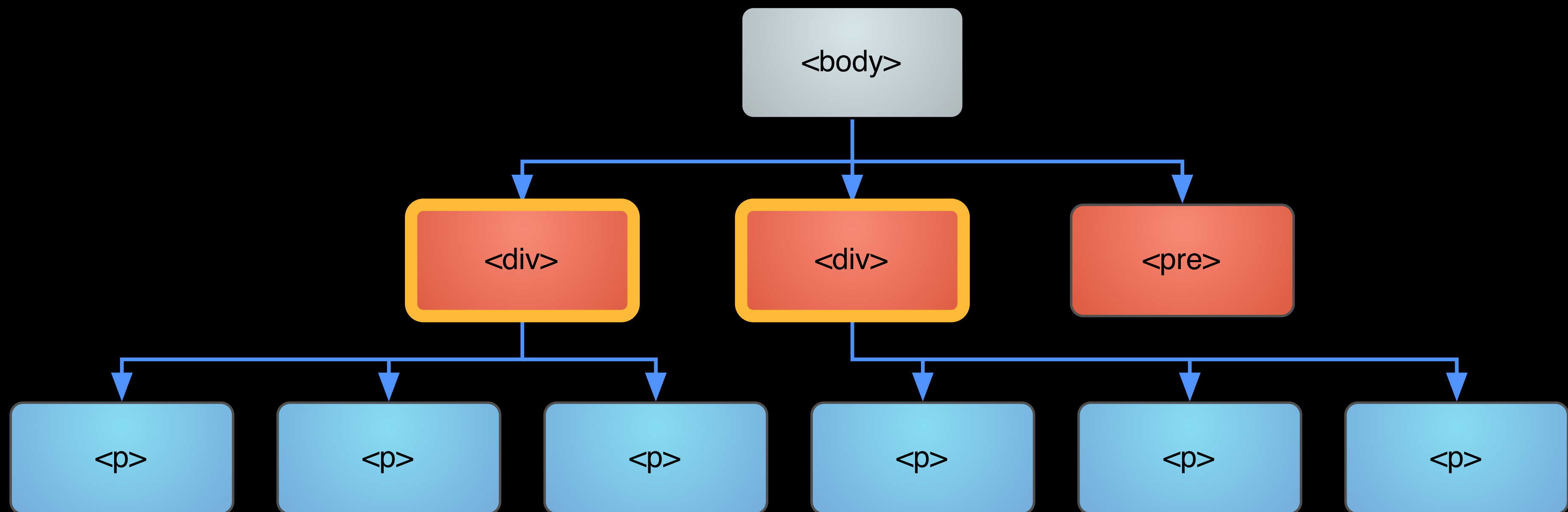
AJAX

PLUGINS

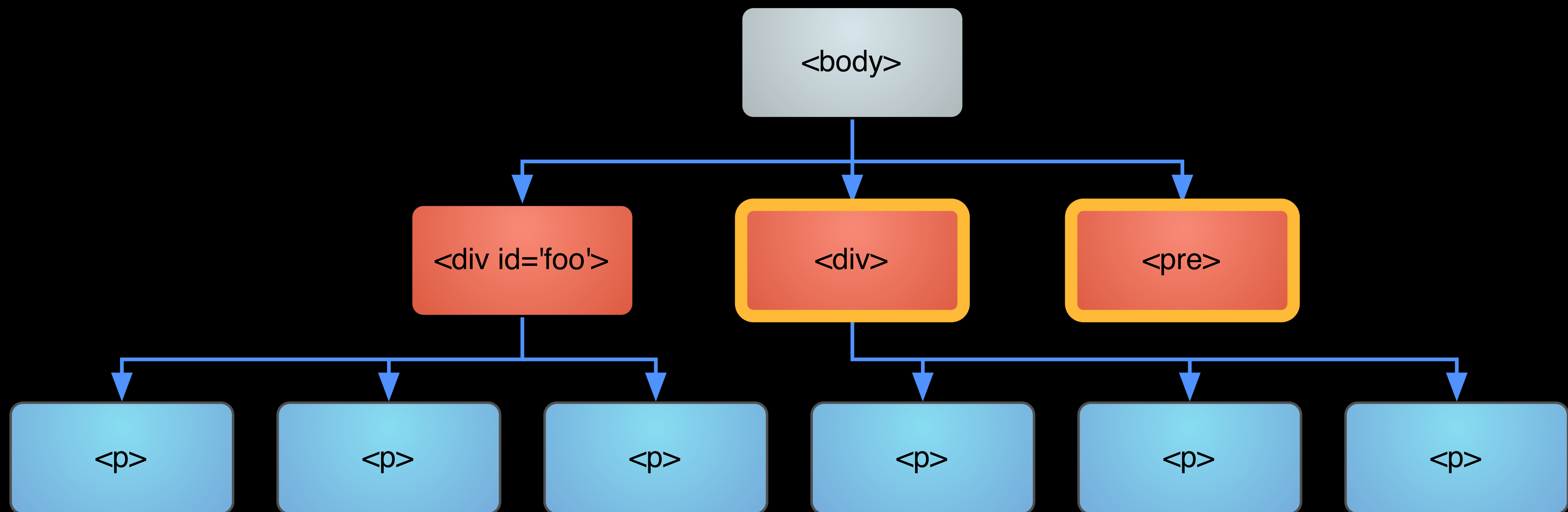
# dom traversal

```
$(“div”).parent();  
$(“div”).siblings();  
$(“div”).next();  
$(“div”).nextAll(“p”);  
$(“div”).nextAll(“p:first”);
```

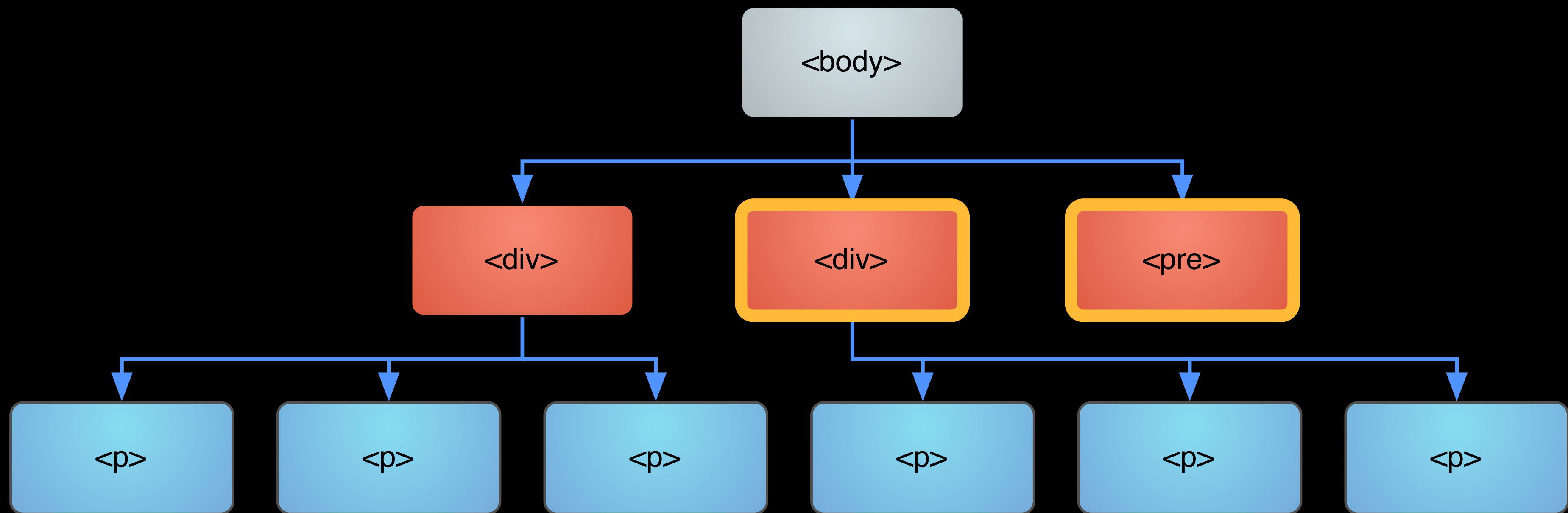
# \$("div")



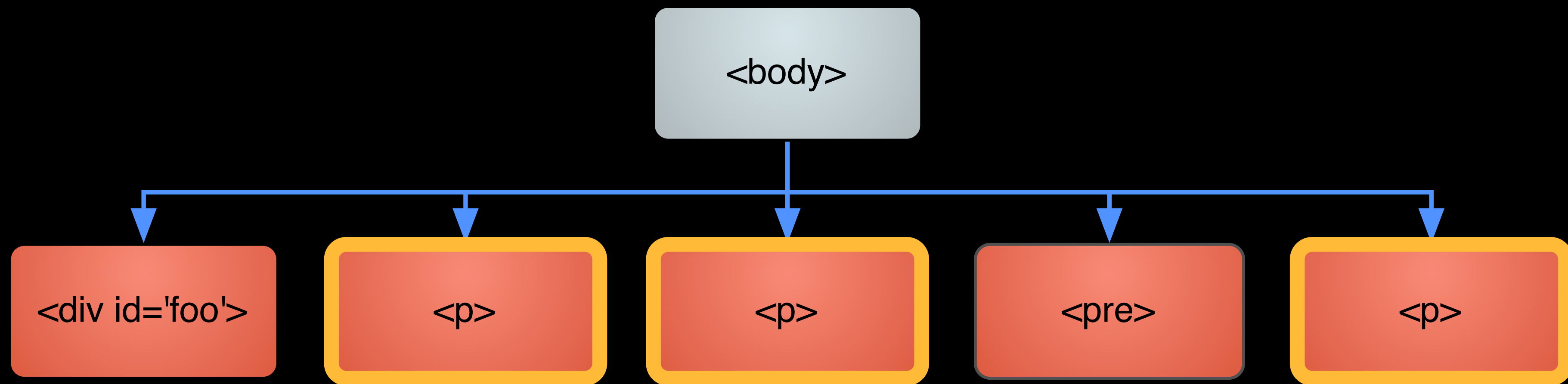
# `$("div#foo").siblings()`



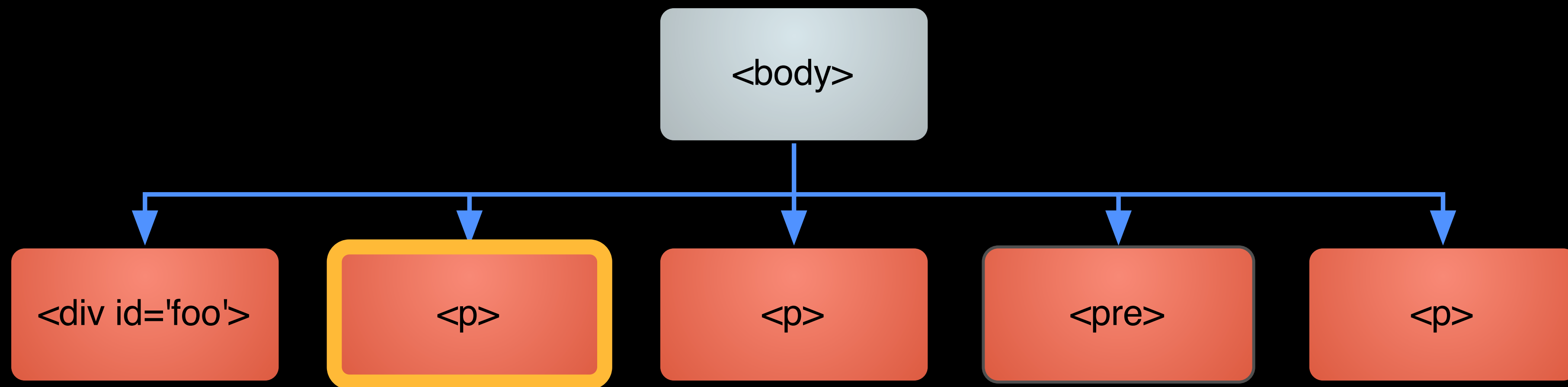
# \$("div").next()



# `$("div").nextall("p")`



# `$("div").nextall("p:first")`



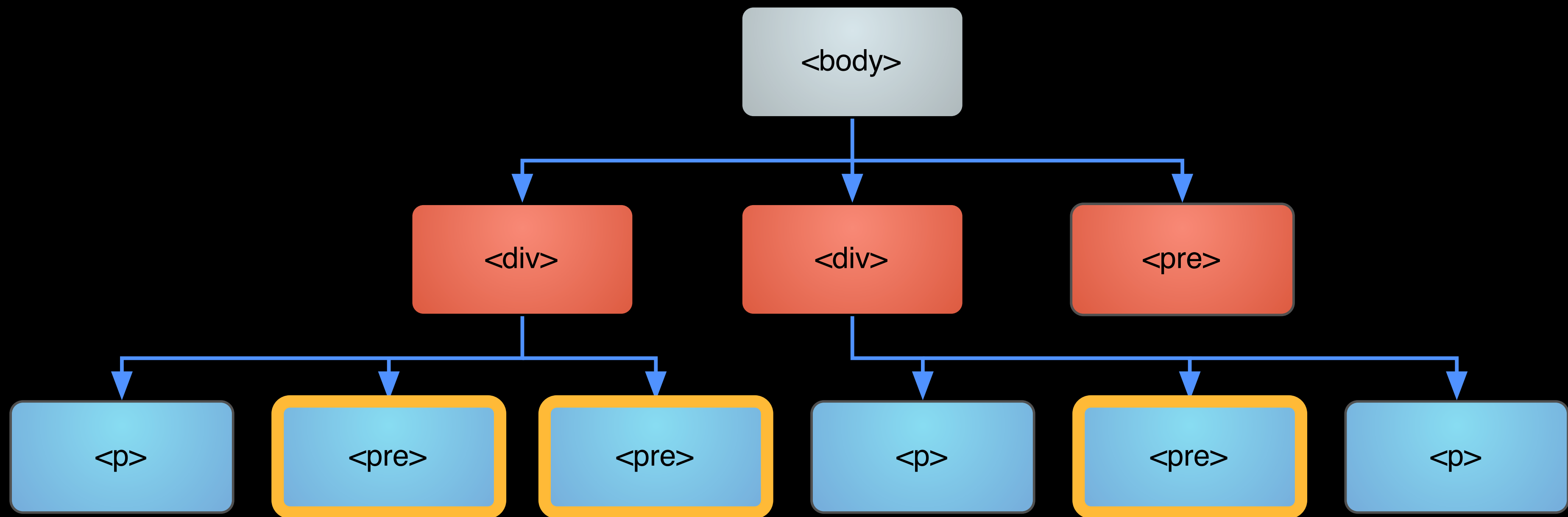
# find

```
$("div pre")
```

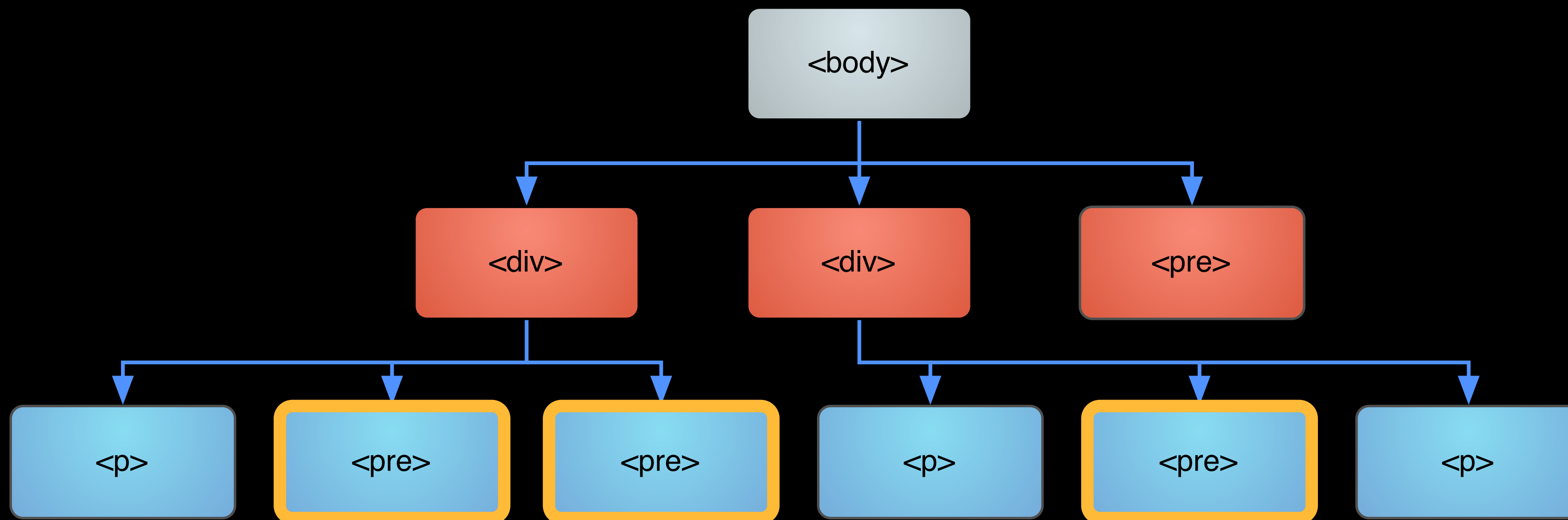
```
$("div").find("pre")
```



# \$("div pre")



# `$("div").find("pre")`



# filter

```
$(“div:contains(some text)”)
```

```
$(“div”).filter(“:contains(some text)”)
```

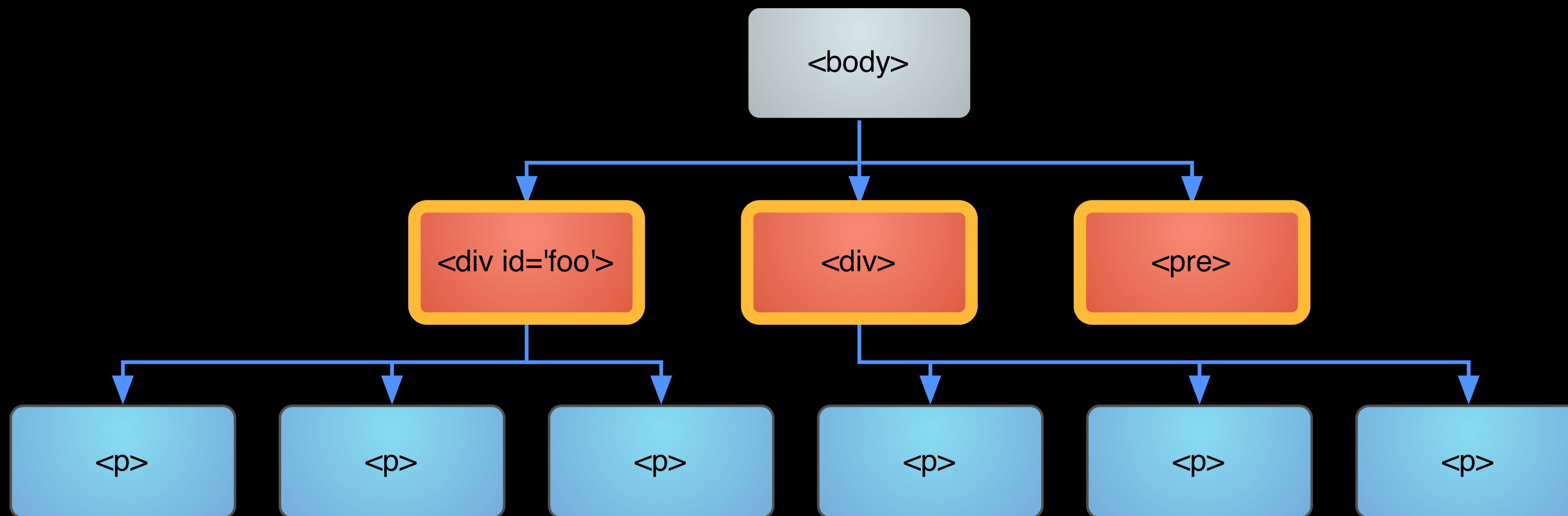
```
$(“div”).filter(function() {  
    return $(this).text().match(“some text”)  
})
```

# andSelf()

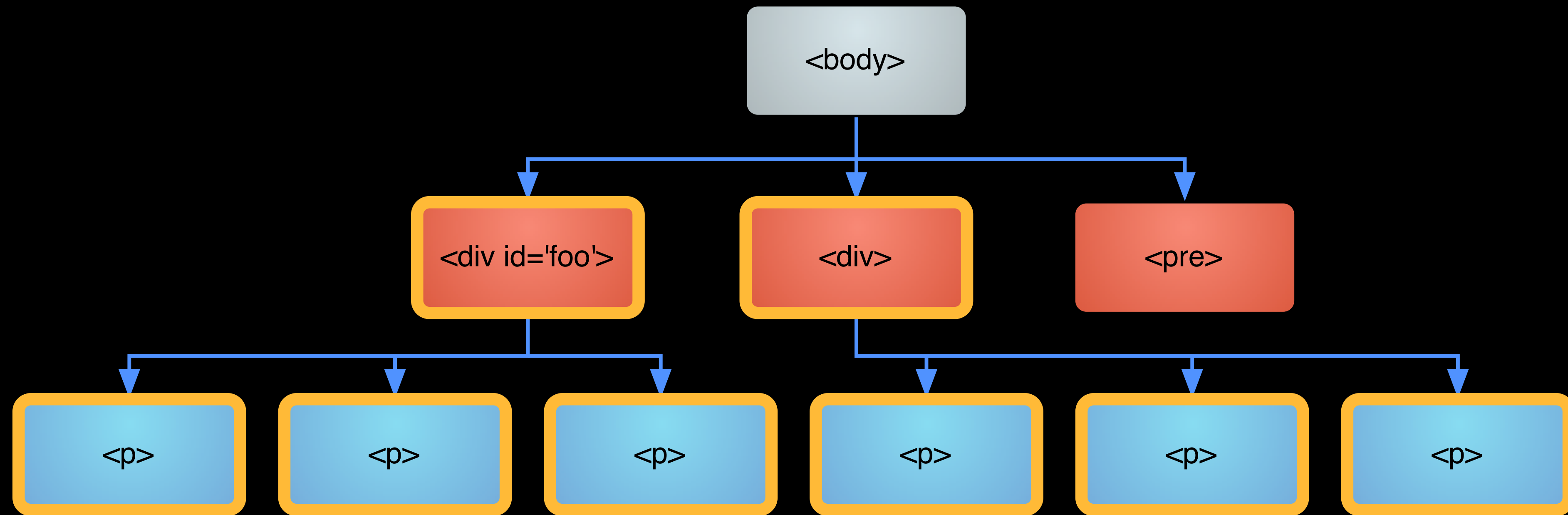
```
$(“div”).siblings().andSelf()
```

```
$(“div”).parent().andSelf()
```

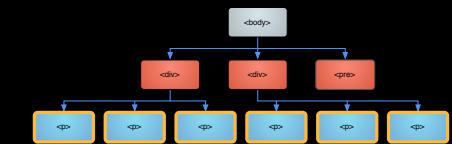
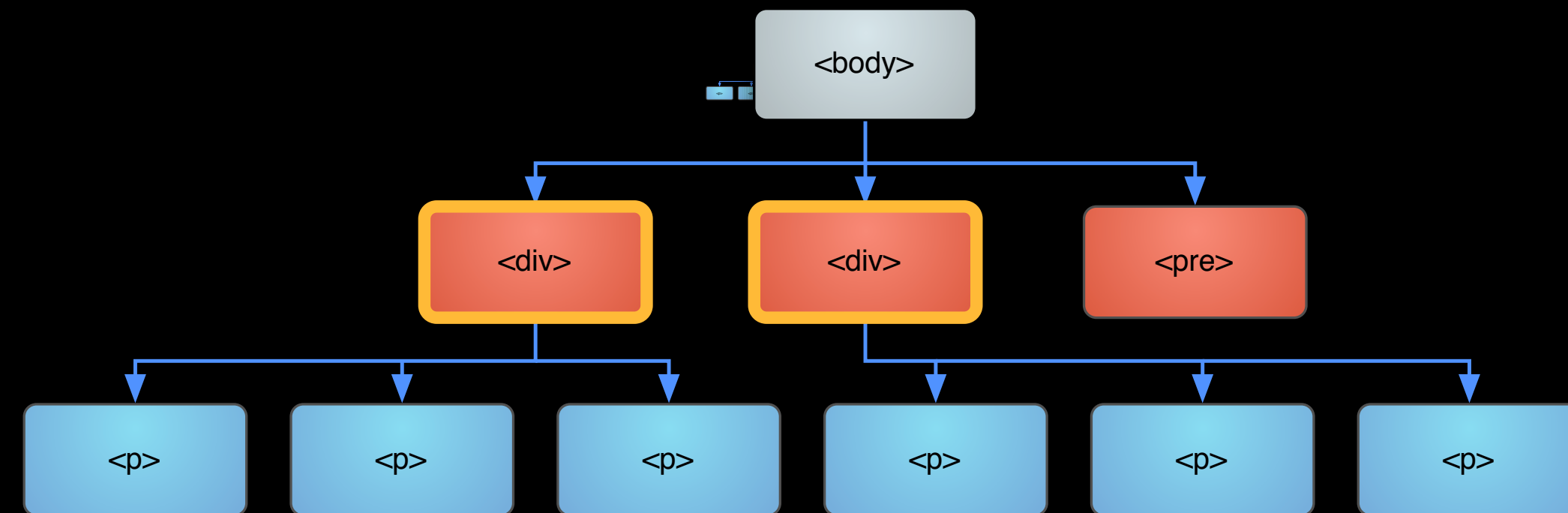
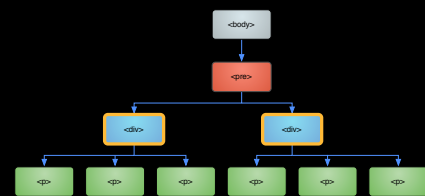
# \$("div").siblings().andSelf()



# \$("p").parent().andSelf()



# end()



`$(“div”).find(“div”).append(“pre”).end()`

# Lab 2: Traversal

- Select any text areas and their labels
- Any span's parent
- All of the form elements from a form that PUT's



# attributes

```
$("#div").attr("id") // returns id
```

```
$("#div").attr("id", "hello") // sets id to hello
```

```
$("#div").attr("id", function() { return this.name })
```

```
$("#div").attr({id: "foo", name: "bar"})
```

```
$("#div").removeAttr("id");
```

# classes

```
$(“div”).addClass(“foo”)
```

```
$(“div”).removeClass(“foo”)
```

```
$(“div”).toggleClass(“foo”)
```

```
$(“div”).hasClass(“foo”)
```

# other

```
$("#div").html()
```

```
$("#div").html("<p>Hello</p>")
```

```
$("#div").text()
```

```
$("#div").text("Hello")
```

```
$("#div").val()
```

```
$("#div").val("Hello")
```

noticing a pattern?

# manipulation

```
$("#div").append("<p>Hello</p>")
```

```
$("#<p>Hello</p>").appendTo("#div")
```

```
$("#div").after("<p>Hello</p>")
```

```
$("#<p>Hello</p>").insertAfter("#div")
```

# Lab 3: Manipulation

*Note: Use the Lab 2 File again*

- Add CSS4 to the list after CSS3
- Remove any images with Dogs
- Turn the ruby row red
- Add some default text to the input field

# 5 parts of jquery

DOM

EVENTS

EFFECTS

AJAX

PLUGINS

# document ready

```
$(document).ready(function() { ... })
```

Alias: `jQuery(function($){ ... })`



# bind

```
$(“div”).bind(“click”, function() { ... })
```

Alias: `$(“div”).click(function() { ... })`

“this”

refers to the element bound

e

```
$("div").click(function(e) { ... })
```

# corrected event object

Property	Correction
target	The element that triggered the event (event delegation)
relatedTarget	The element that the mouse is moving in (or out) of
pageX/Y	The mouse cursor relative to the document
which	<b>mouse:</b> 1 (left) 2 (middle) 3 (right)
	<b>keypress:</b> The ASCII value of the text input
metaKey	Control on Windows and Apple on OSX

# trigger

`$("div").trigger("click")`

Alias: `$("div").click()`

# triggerHandler

doesn't trigger the browser's default actions

# custom events

```
$(“div”).bind(“myEvent”, function() { ... })  
$(“div”).trigger(“myEvent”)
```

# hover

```
$("div").hover(function() { ... }, function() { ... })
```



# toggle

```
$("div").toggle(function() { ... }, function() { ... })
```



# live

```
$(“div”).live(“click”, function() { ... })
```

# 5 parts of jquery

DOM

EVENTS

EFFECTS

AJAX

PLUGINS

# Fades

```
$(“div”).fadeIn()
```

```
$(“div”).fadeOut(“slow”)
```

# slides

```
$(“div”).slideUp(200)
```

```
$(“div”).slideDown(“slow”)
```

# animate

```
$(“div”).animate({height: “toggle”, opacity: “toggle”})
```

```
$(“div”).animate({fontSize: “24px”, opacity: 0.5}, {easing: “expo”})
```

# Lab 4: Events and Effects

*Note: Use the Lab 2 File again*

- Fade out all of the divs
- Make each img grow when you mouseover them (and shrink again after you leave)
- Make clicking an li collapse the sub list

# 5 parts of jquery

DOM

EVENTS

EFFECTS

AJAX

PLUGINS



# make easy things easy

```
$(“div”).load(“some_url”);  
$(“div”).load(“some_url”, {data: “foo”},  
function(text) { ... });
```

it's easy to do it right

```
$.getJSON("some_url", function(json) { ... })
```

```
$.getJSON("some_url", {data: "foo"},  
function(json) { ... })
```

# it's consistent

```
$.get("some_url", function(text) { ... })
```

```
$.post("some_url", {data: "foo"},  
function(text) { ... })
```

# and powerful

## \$.ajax Options

- async
- beforeSend
- cache
- complete
- contentType
- data
- dataType
- error
- global
- ifModified
- jsonp
- processData
- success
- timeout
- type

# and powerful

## global ajax settings

```
/* No Ajax requests exist, and one starts */  
$("div.progress").ajaxStart(function() { $(this).show() });
```

```
/* The last Ajax request stops */  
$("div.progress").ajaxStop(function() { $(this).hide() });
```

```
/* Any Ajax request is sent */  
$("p").ajaxSend(function() { ... });
```

```
/* Any Ajax request completes (success or failure) */  
$("div").ajaxComplete(function() { ... });
```

```
/* Any Ajax request errors out */  
$("div").ajaxError(function() { ... });
```

```
/* Any Ajax request succeeds */
```

# 5 parts of jquery

DOM

EVENTS

EFFECTS

AJAX

PLUGINS

there are hundreds

which are important?



# jquery ui

- Draggables
- Droppables
- Sortables
- Selectables
- Resizables

Primitives



- Accordion
- Date Picker
- Dialog
- Slider
- Tabs

Widgets



<http://ui.jquery.com>

# jquery forms

AJAXIFY A FORM IN ONE EASY STEP:

```
$("FORM.REMOTE").AJAXFORMO
```

<http://www.malsup.com/jquery/form/>

# form validation

SPECIFY VALIDATION RULES IN YOUR MARKUP

[http://bassistance.de/jquery-plugins/  
jquery-plugin-validation/](http://bassistance.de/jquery-plugins/jquery-plugin-validation/)



# metadata plugin

SPECIFY METADATA FOR ELEMENTS IN MARKUP

```
<div data="{some: 'data'}">  
$("div").metadata().some // returns 'data'
```

[http://jqueryjs.googlecode.com/svn/  
trunk/plugins/metadata/](http://jqueryjs.googlecode.com/svn/trunk/plugins/metadata/)

# Event Driven JavaScript

# jQuery on Rails

# jQuery and RJS

# Rails 3



# Ajax and Rails

```
$.getJSON("/rails/action")
```

# Ajax and Rails

```
respond_to do |format|  
  format.json {  
    render :json => obj  
  }  
end
```

# link\_to\_remote

```
link_to_remote "Delete this post", :update => "posts",  
  :url => { :action => "destroy", :id => post.id }
```

```
link_to "Delete this post",  
  url(:action => "destroy", :id => post.id), :rel => "#posts"
```

```
$('a.remote').click(function() { $(this.rel).load(this.href) });
```

# form\_remote\_tag

```
<% form_remote_tag :url => ..., :update => "res" do -%>
```

```
<% form_tag :url => ..., :rel => "#res" do -%>
```

```
$('form').each(function() { $(this).ajaxForm({ target: this.rel }) })
```

# observe\_field

```
<%= observe_field :suggest,  
  :url => { :action => :find },  
  :frequency => 0.25,  
  :update => :suggest,  
  :with => 'q'  
%>
```

```
var lastTime = new Date  
$('#suggest').keyup(function() {  
  if(new Date - lastTime > 250) {  
    var field = $('#suggest');  
    var url = field.attr('rel');  
    field.load(url, {q: field.val()});  
  }  
});
```

# periodically\_call\_remote

```
periodically_call_remote(  
  :url => { :action => 'avgs' },  
  :update => 'avg',  
  :frequency => '20')
```

```
setInterval(function() {  
  $('#avg').load('/some/avgs');  
}, 20000);
```