

**Let's Talk about**

**Ph1ex,**

**and more**

**Let's Talk about**

**ERB**

Let's Talk about rendering our things in

**HTML**

# Who here...

make some noise if you do

who here

 **viewhelpers in  
rails?**

make some noise

who here



make some noise

who here



# ViewComponent?

make some noise

who here



# Ruby & Rails

make some noise (you should be really loud here, otherwise the point I am trying to make doesn't work.)



Rails view layer  
feels a bit ... 🙄



gem "phlex"

Phlex::HTML

```
render List.new
```

```
class List < Phlex::HTML  
end
```

```
class List < Phlex::HTML
  def template
    ...
  end
end
```

```
class List < Phlex::HTML
  def template
    ul(class: "pretty") do
      ...
    end
  end
end
```

```
class List < Phlex::HTML
  def template
    ul(class: "pretty") do
      li { "Helvetica Ruby" }
      li { "Friendlyrb" }
    end
  end
end
```



So what?



 Extract methods

 Inheritance

 Include Modules

 DSL

# Expressive Ruby

**Example**

**clarifying logic**

```
<% @title = "search" %>
<% if @error_msg %>
  <div class="errorExplanation">
    <p><%= @error_msg %></p>
  </div>
<% end %>
<%= link_to t("advanced_search"), advanced_search_path %>
<% if @yanked_filter %>
  <% @subtitle = t('.subtitle', h(params[:query])) %>

  <% if @yanked_gem.present? %>
    ...
```

```
class Searches::Show < Phlex::HTML
  def template
    error_messages
    if @yanked_filter
      yanked_search
    else
      header
      aggregations
      if @gems.any?
        search_results
      else
        suggestion_results
      end
    end
  end
end
...
end
```

**Example**

**Neat table API**

```
render Table.new(rows: products) do |table|
  table.column("Name") { |product| link_to(product.name, product) }
  table.column("Description") { |product| truncate(product.desc) }
end
```

Example

Go crazy with  
forms

```
class ChangePasswordForm < ApplicationForm
  validates :password_rules ...

  def template ...

  def save ...
end
```



# phlex is



fun



fast



ruby

**phlex.fun**

**@roliruby.social**

**roliruby.com**