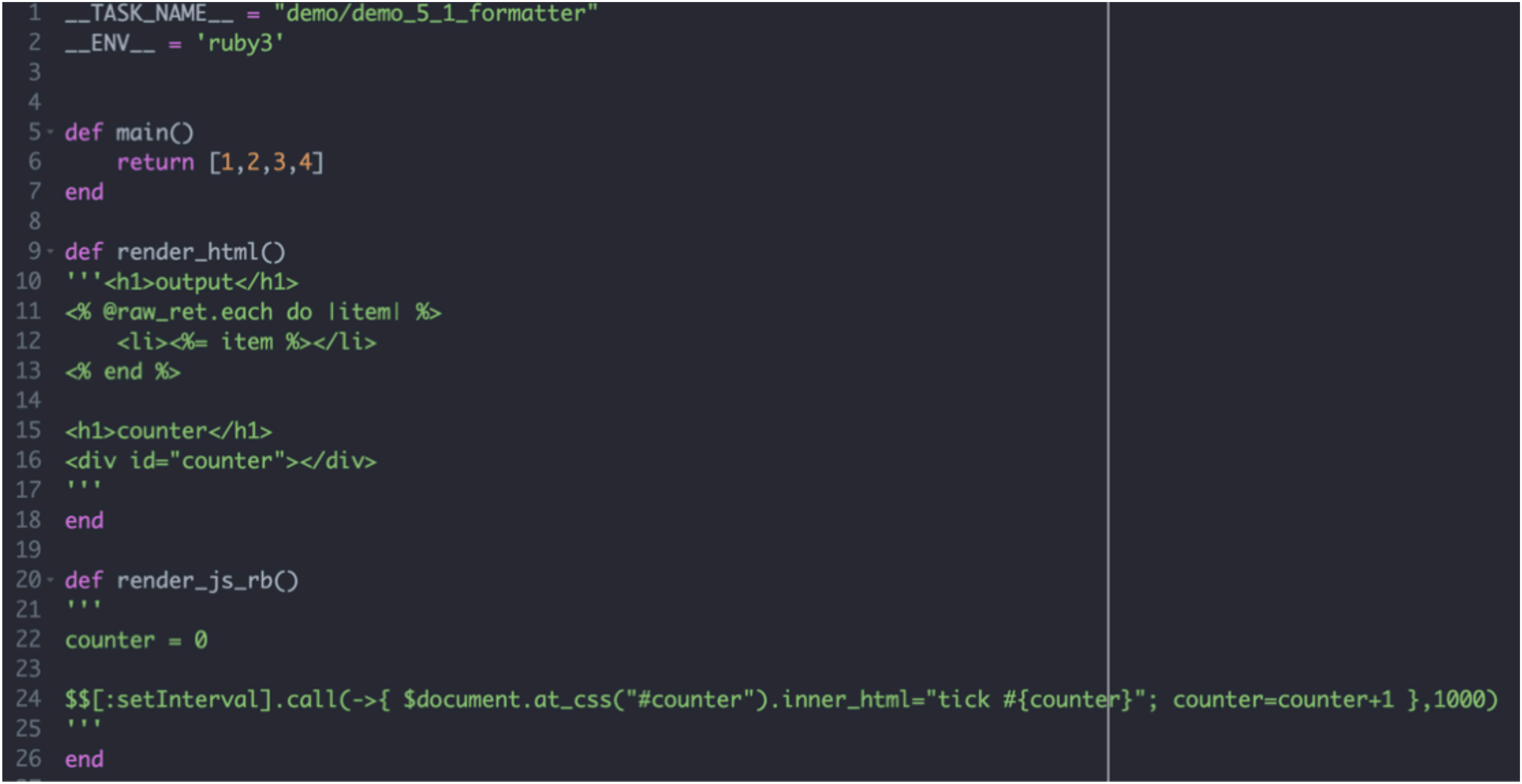
HTML Formatter



```ruby
__TASK_NAME__ = "demo/demo_5_1_formatter"
__ENV__ = 'ruby3'


def main()
    return [1,2,3,4]
end

def render_html()
'''<h1>output</h1>
<% @raw_ret.each do |item| %>
    <li><%= item %></li>
<% end %>

<h1>counter</h1>
<div id="counter"></div>
'''
end

def render_js_rb()
'''
counter = 0

$$[:setInterval].call(->{ $document.at_css("#counter").inner_html="tick #{counter}"; counter=counter+1 },1000)
'''
end
```

HTML Template

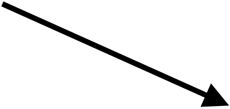
Ruby-JS Template

Backend Logic

Backend Logic

HTML

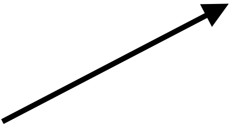
JSRB



Opal
Ruby-JS Compiler

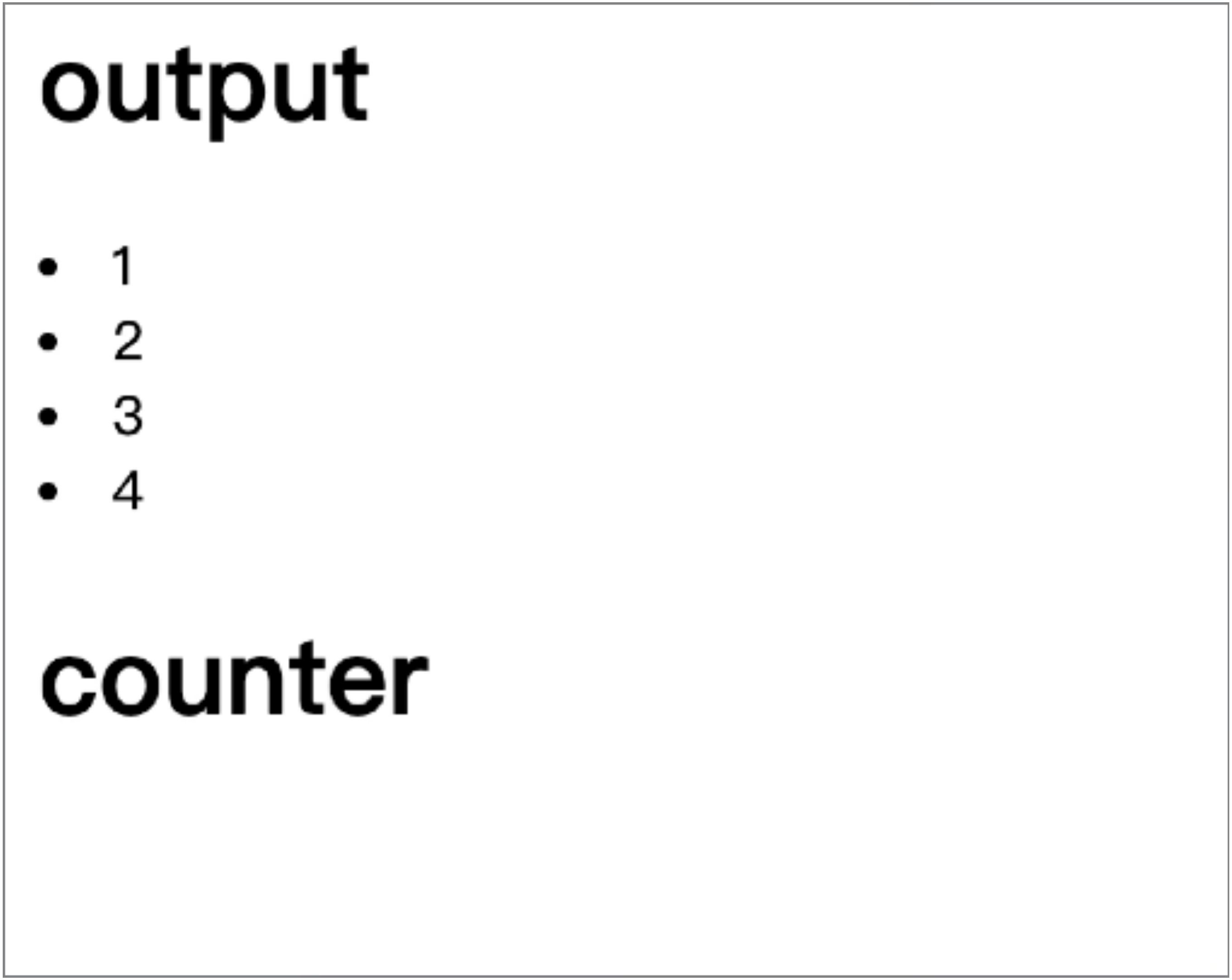
Javascript

→



HTML

Page



output

- 1
- 2
- 3
- 4

counter

output

- 1
- 2
- 3
- 4

counter

output

- 1
- 2
- 3
- 4

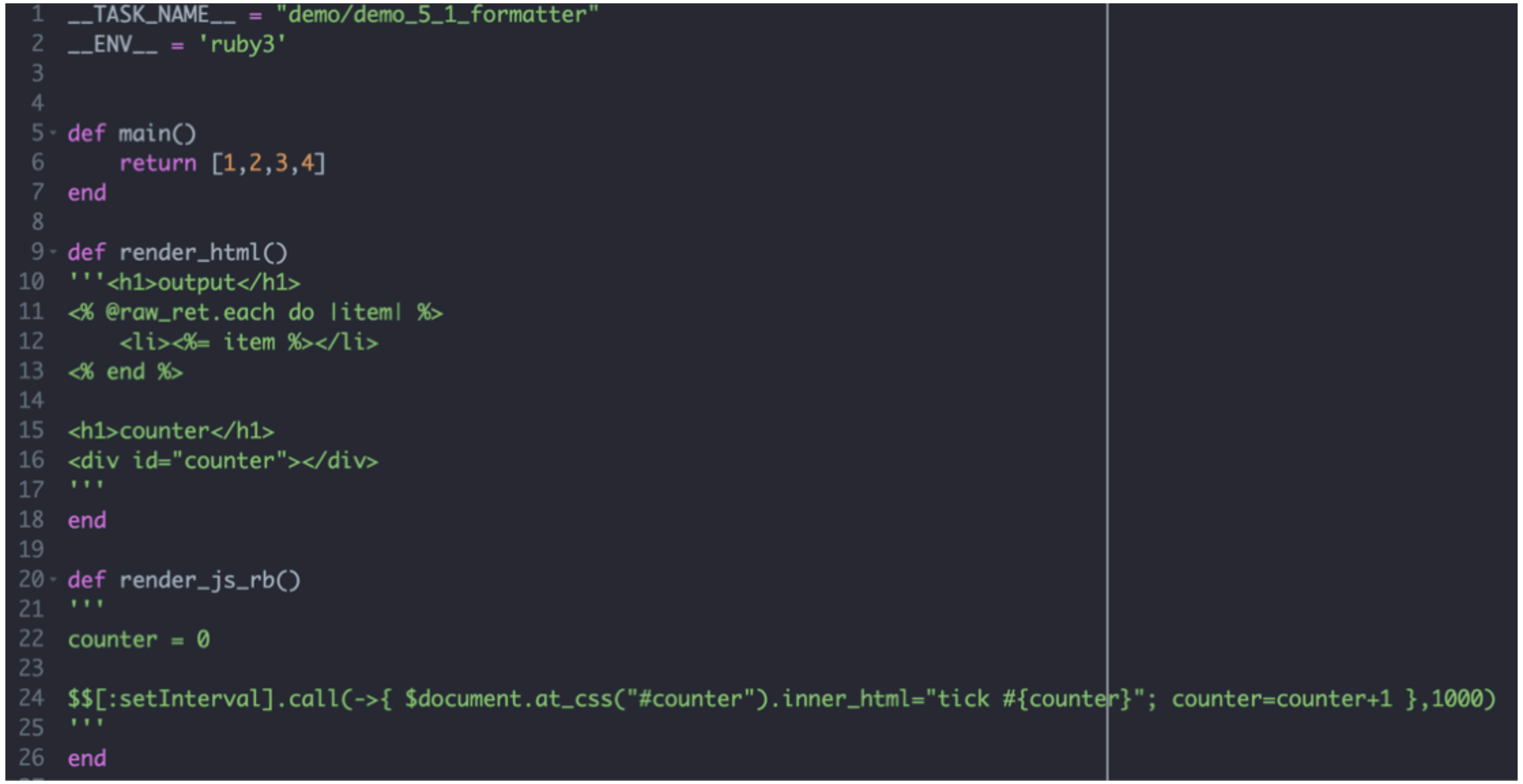
counter

HTML Formatter

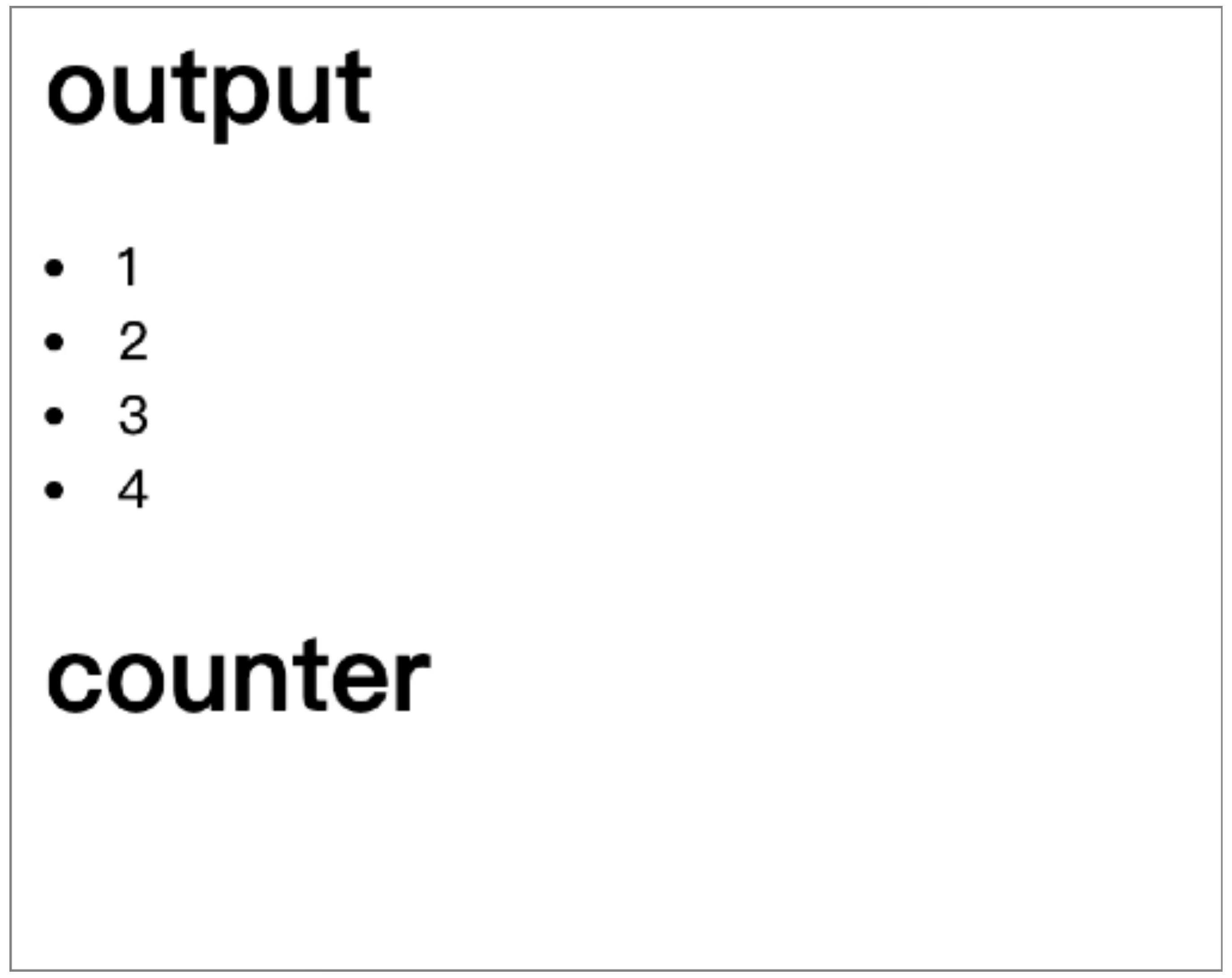
Backend Logic

HTML Template

Ruby-JS Template



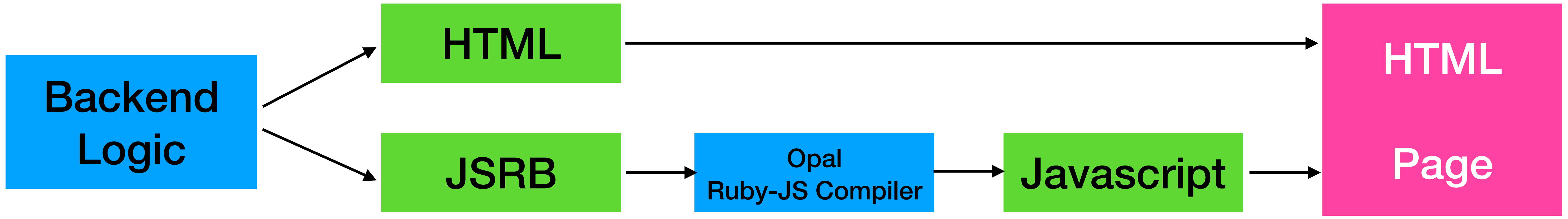
```
__TASK_NAME__ = "demo/demo_5_1_formatter"
__ENV__ = 'ruby3'


def main()
    return [1,2,3,4]
end

def render_html()
'''<h1>output</h1>
<% @raw_ret.each do |item| %>
    <li><%= item %></li>
<% end %>

<h1>counter</h1>
<div id="counter"></div>
'''
end

def render_js_rb()
'''
counter = 0

$$[:setInterval].call(->{ $document.at_css("#counter").inner_html="tick #{counter}"; counter=counter+1 },1000)
'''
end
```



output

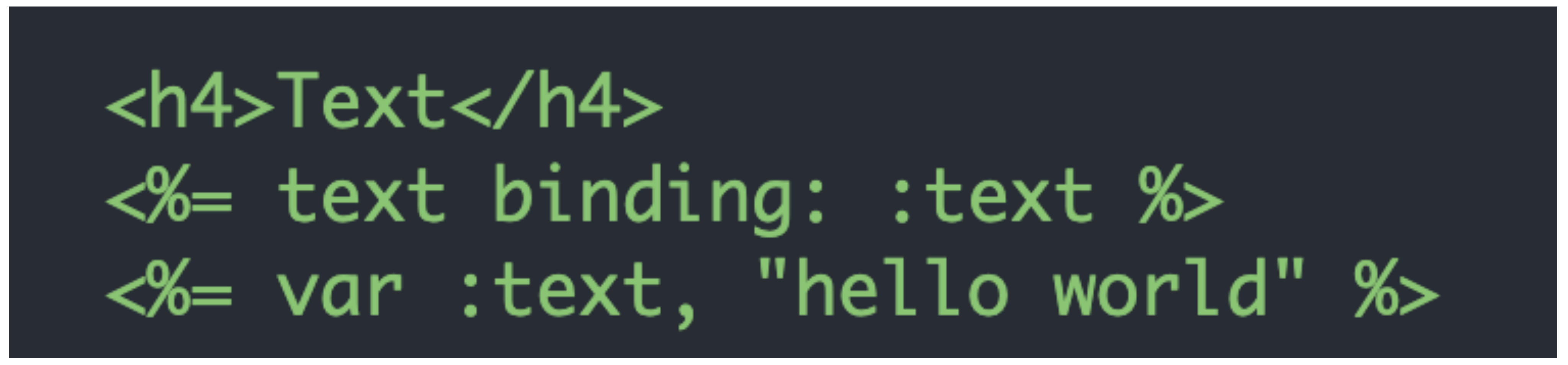
- 1
- 2
- 3
- 4

counter

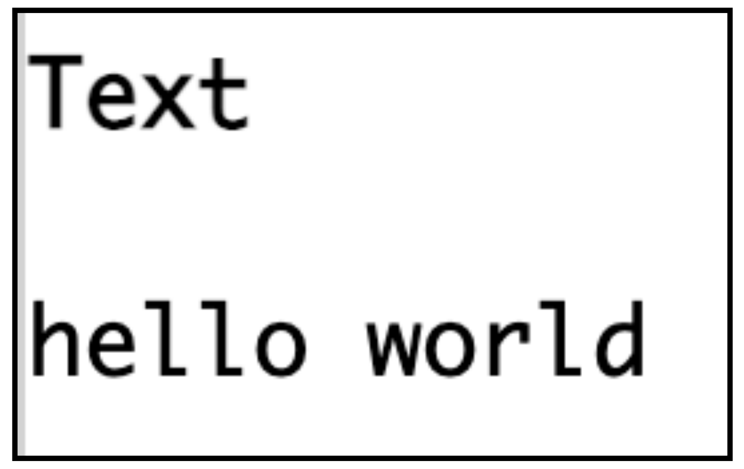


Widgets - Text

- Text 绑定了 Binding Var
- Binding Vars 可以被输入事件改变，或者被计算表达式改变，或者被代码逻辑改变



```
<h4>Text</h4>
<%= text binding: :text %>
<%= var :text, "hello world" %>
```



```
Text

hello world
```