



slim template

SHORT DISCUSSION ABOUT SLIM TEMPLATE



◀ slim is... ▶

a template language whose goal is reduce the syntax to the essential parts without becoming cryptic.

<https://github.com/slim-template/slim>



slim's features

- ◀ elegant syntax ▶
- highly configurable
- high performance
- supported by rails, sinatra, other major frameworks
- supports markdown

<https://github.com/slim-template/slim>

slim vs some

< template engines >

(usage)



https://www.ruby-toolbox.com/categories/template_engines

erb vs haml vs slim

erb

ruby's built in template engine
uses `<%...%>` and `<%=...%>`
familiar to use (html-like syntax)
not whitespace dependent

haml

provides layer of abstraction above html
clean, straightforward syntax
dry, short, understandable codes
integrates ruby code

slim

faster than erb and haml, second to erubis
based on several engines like haml and jade
flexible to use
offers logic less mode (like mustache)
similar to haml, but does not require % before tag declarations

<http://www.vemployee.com/blog/revisiting-template-engines-in-ruby-on-rails/>

a website ran some tests...

```
Erb render: 41.091 [ms] (average)
Haml render: 43.508 [ms] (average)
Slim render: 40.387 [ms] (average) / Fastest
```

```
Erb render: 36.021 [ms] (median)
Haml render: 37.316 [ms] (median)
Slim render: 35.349 [ms] (median) / Fastest
```

<http://blog.crowdint.com/2013/07/17/view-engines-performance.html>

slim vs haml

```
#hello-id.hello-class
/ OR
div *{ :id => 'hello-id' , :class => 'hello-class' }
/ OR
/ YOU CAN ALSO GROUP ATTRs IN [], (), {}
div id= 'hello-id' class= 'hello-class'
/ OR
< div id= 'hello-id' class= 'hello-class' >
...
< /div >
/ OR

/ EXTRA: ATTRIBUTE MERGING
a class= [ 'hello-class-a' , 'hello-class-b' ]
a class= :hello_class_a, :hello_class_b
```

```
#hello-id.hello-class
```

...control code, output

(the same?)

```
/ SLIM
- variable integer if condition

= @variable
= puts 'string'
```

```
/ HAML
- variable integer if condition

= @variable
= puts 'string'
```


...tag nesting

```
/ SLIM
#parent-div
  #child-div content
```

```
/ EXTRA: INLINE TAGS
#parent-div: #child-div content
```

```
/ HAML
#parent-div
  #child-div content
```

slim's logic less (via plugin)

```
/ CONDITIONAL
/ if the object is not false or empty?, the content will show
- boolean_value
h1 'Hello World!'

/ ITERATION
/ if the object is an array, the section will iterate
- array
tr: td 'Hello World!'
```

i think that...

from what i've read, slim is faster, and (faster is better?)

< syntax-wise, there are many similarities since slim is somehow derived from haml >

but, even though slim is flexible in terms of creating the tags, it's also a downside since it leads to inconsistent code (especially when working in a team)

one thing that slim has over haml is that it's not really whitespace dependent.

< conclusion? >

if you prefer speed and flexibility, *slim is the way to go*

if you like code that is neat and DRY, *use haml*





fin

