

Hamsl can be used in command line, as a separate Ruby module, or in a Ruby on Rails application making Hamsl suitable for a wide range of applications.

External links

<u>Paradigm</u>	Template engine
<u>Designed by</u>	<u>Hampton Catlin</u>
<u>Developers</u>	<u>Natalie Weizenbaum</u> (https://github.com/nex3) (past), <u>Norman Clarke</u> (https://github.com/norman), <u>Matt Wildig</u> (https://github.com/mattwildig), <u>Akira Matsuda</u> (https://github.com/amatsuda), <u>Tee Parham</u> (https://github.com/teeparham) ^[1]
<u>Stable release</u>	5.1.2 / August 6, 2019 ^[2]
<u>Implementation language</u>	<u>Ruby</u>
<u>OS</u>	<u>Cross-platform</u>
<u>License</u>	<u>MIT License</u> and <u>Unspace Interactive</u> (http://unspace.ca/) ^[1]
<u>Filename extensions</u>	.haml
<u>Website</u>	<u>haml.info</u> (http://haml.info/)

History

Haml was originally introduced by [Hampton Catlin](#) with its initial release in 2006 and his work was taken ahead by a few other people.^[5] His motive was to make [HTML](#) simpler, cleaner and easier to use. Since 2006, it has been revised several times and newer versions were released. Until 2012 Natalie Weizenbaum was the primary maintainer of Haml, followed by Norman Clarke until 2015.^[5] Natalie worked on making Haml usable in [Ruby](#) applications, while the branding and design was done by Nick Walsh.^[5] Others who are currently in the maintenance team are Matt Wildig, Akira Matsuda, Tee Parham and Takashi Kokubun.^[1]

Version history

Version 2.2.0 was released in July 2009 with support for Ruby 1.9 and [Rails](#) 2.0 or above.^[2] Version 3.0.0 was released in May 2010, adding support for Rails 3 and some performance improvements. The fourth major version broke compatibility with previous versions, only supporting Rails 3 and Ruby 1.8.7 or above, and marked the switch to semantic versioning. Several amendments like increasing the performance, fixing a few warnings, compatibility with latest versions of Rails, fixes in the documentation and many more were made in the Haml 4 series.^[2] Version 5.0.0 was released in April 2017. It supports Ruby 2.0.0 or above and drops compatibility with Rails 3.^[2] A '[trace](#)'^[6] option, which helps users to perform tracing on Haml template, has been added.

Features

Four principles were involved in development of Haml.^[5]

User-friendly markup

[Markup language](#) is user-friendly if it adheres to following features:

- Easy to understand the language
- Easy of use (Implementation)

DRY

Markup language should adhere to the Don't repeat yourself ([DRY](#)) principle. It should:

- Avoid unnecessary repetitions
- Focus on clean code

Well-indented

Markup language with good [indentation](#) improves appearance, makes it easy to read for readers and also to determine where a given element starts and ends.

Clear structure

Markup language with a clear structure will help in code maintenance and logical understanding of final result. It is unclear whether Haml offers any differential advantage in this regard.

Examples

Haml markup is similar to CSS in syntax. For example, Haml has the same dot `.` representation for classes as CSS does, making it easy for developers to use this markup.

"Hello, World!"

Haml as a command-line tool

The following are equivalent as HAML recognises CSS selectors:

```
%p{:class => "sample", :id => "welcome"} Hello, World!
```

```
%p.sample#welcome Hello, World!
```

These render to the following HTML code:

```
<p class="sample" id="welcome">Hello, World!</p>
```

Haml as an add-on for Ruby on Rails

To use Haml with Ruby, the Ruby Gemfile should include this line:

```
gem 'haml'
```

Similar to eRuby, Haml also can access local variables (declared within same file in Ruby code). This example uses a sample Ruby controller file.^[7]

- file: app/controllers/messages_controller.rb

```
class MessagesController < ApplicationController
  def index
    @message = "Hello, World!"
  end
end
```

- file: app/views/messages/index.html.haml

```
#welcome
%p= @message
```

This renders to:

```
<div id="welcome">
  <p>Hello, World!</p>
</div>
```

Haml as a Ruby module

To use Haml independent of Rails and ActionView (http://guides.rubyonrails.org/action_view_overview.html), install haml gem, include it in Gemfile and simply import [`Usage: require 'haml'`] it in Ruby script or invoke Ruby interpreter with `-rubygems` flag.

```
welcome = Haml::Engine.new("%p Hello, World!")
welcome.render
```

Output:

```
<p>Hello, World!</p>
```

Haml::Engine (<http://haml.info/docs/yardoc/Haml/Engine.html>) is a Haml class.

Basic example

Haml uses whitespace indentation (two spaces) for tag nesting and scope. This acts as a replacement for the open-end tag pairs, making it DRY and cleaner. The following example compares the syntaxes of Haml and eRuby (Embedded Ruby), alongside the HTML output.

Haml

```
%div.category
  %div.recipes
    %h1=
recipe.name
    %h3=
recipe.category
    %div
      %h4=
recipe.description
```

ERB

```
<div
  class="category">
  <div
    class="recipes">
    <h1><%=
recipe.name %></h1>
    <h3><%=
recipe.category %>
</h3>
  </div>
  <div>
    <h4><%=
recipe.description
%></h4>
  </div>
</div>
```

HTML

```
<div class="category">
  <div class="recipes">
    <h1>Cookie</h1>
    <h3>Desserts</h3>
  </div>
  <div>
    <h4>Made from dough and sugar. Usually
circular in shape and has about 400 calories.
  </h4>
  </div>
</div>
```

Key differences are:

- Haml doesn't have both start and end for each element like eRuby
- eRuby syntax looks a lot like HTML and is thereby more HTML-like while Haml is more CSS-like.
- Haml uses indentation to nest tag elements whereas eRuby uses the same HTML representation

- In Haml properties like `class`, `id` can be represented by `.`, `#` respectively instead of regular `class` and `id` keywords. Haml also uses `%` to indicate a HTML element instead of `<>` as in eRuby.

Example with embedded Ruby code

Note: This is a simple preview example and may not reflect the current version of the language.

```
!!!
%html{ :xmlns => "http://www.w3.org/1999/xhtml", :lang => "en", "xml:lang" => "en"}
%head
  %title BoBlog
  %meta{"http-equiv" => "Content-Type", :content => "text/html; charset=utf-8"}
  %link{"rel" => "stylesheet", "href" => "main.css", "type" => "text/css"}
%body
  #header
    %h1 BoBlog
    %h2 Bob's Blog
  #content
    - @entries.each do |entry|
      .entry
        %h3.title= entry.title
        %p.date= entry.posted.strftime("%A, %B %d, %Y")
        %p.body= entry.body
  #footer
    %p
      All content copyright © Bob
```

The above Haml would produce this XHTML:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html lang='en' xml:lang='en' xmlns='http://www.w3.org/1999/xhtml'>
  <head>
    <title>BoBlog</title>
    <meta content='text/html; charset=utf-8' http-equiv='Content-Type' />
    <link href="/stylesheets/main.css" media="screen" rel="Stylesheet" type="text/css" />
  </head>
  <body>
    <div id='header'>
      <h1>BoBlog</h1>
      <h2>Bob's Blog</h2>
    </div>
    <div id='content'>
      <div class='entry'>
        <h3 class='title'>Halloween</h3>
        <p class='date'>Tuesday, October 31, 2006</p>
        <p class='body'>
          Happy Halloween, glorious readers! I'm going to a party this evening... I'm very
excited.
        </p>
      </div>
      <div class='entry'>
        <h3 class='title'>New Rails Templating Engine</h3>
        <p class='date'>Friday, August 11, 2006</p>
        <p class='body'>
          There's a very cool new Templating Engine out for Ruby on Rails. It's called Haml.
        </p>
      </div>
    </div>
    <div id='footer'>
      <p>
        All content copyright © Bob
      </p>
    </div>
  </body>
</html>
```

Implementations

The official implementation of Haml has been built for [Ruby](#) with plugins for [Ruby on Rails](#) and [Merb](#), but the Ruby implementation also functions independently. Haml can be easily used along with other languages. Below is a list of languages in which Haml has implementations:

- Ruby: [hamlit](https://github.com/k0kubun/hamlit) (<https://github.com/k0kubun/hamlit>)
- PHP: Fammel (<https://web.archive.org/web/20100303124319/http://wiki.github.com/dxw/Fammel/>), pHAML (<http://phaml.sourceforge.net/>), phamlp (<https://code.google.com/p/phamlp/>), phpHaml (<http://phphaml.sourceforge.net/>) (PHP 5), HAML-TO-PHP (<https://web.archive.org/web/20110417115303/http://haml-to-php.com/>) (PHP 5), Multi target HAML (<https://github.com/arnaud-lb/MtHaml>) (PHP 5.3)
- Javascript: [haml-js](https://github.com/creationix/haml-js) (<https://github.com/creationix/haml-js>)
- Python: [HamIPy](https://github.com/jessemiller/HamIPy) (<https://github.com/jessemiller/HamIPy>)
- Common Lisp: [cl-haml](https://github.com/Unspeakable/cl-haml) (<https://github.com/Unspeakable/cl-haml>)
- Dart: [Hart](https://github.com/lvivski/hart) (<https://github.com/lvivski/hart>)
- Java: [JHaml](https://github.com/raymyers/JHaml) (<https://github.com/raymyers/JHaml>)
- Lua: [LuaHaml](https://github.com/norman/lua-haml) (<https://github.com/norman/lua-haml>)
- .NET: [NHaml](https://code.google.com/p/nhaml/) (<https://code.google.com/p/nhaml/>)
 - ASP.NET: [MonoRail NHaml](https://web.archive.org/web/20080608081951/http://weblogs.asp.net/mirvodasergey/archive/2008/06/02/introducing-monorail-nhaml.aspx) (<https://web.archive.org/web/20080608081951/http://weblogs.asp.net/mirvodasergey/archive/2008/06/02/introducing-monorail-nhaml.aspx>)
- Perl: [Text::Haml](https://metacpan.org/module/Text::Haml) (<https://metacpan.org/module/Text::Haml>)
- Scala: [Scalate](https://web.archive.org/web/20100216131951/http://scalate.fusesource.org/index.html) (<https://web.archive.org/web/20100216131951/http://scalate.fusesource.org/index.html>)

See also


- [BBCode](#)
- [eRuby](#)
- [Markaby](#)
- [Ruby](#)
- [Ruby on Rails](#)
- [YAML](#)
- [Sass](#) – a similar system for CSS, also designed by Catlin.
- [Website Meta Language](#) – another template language with similar functionalities
- [Web template](#) – general concept of template to [HTML](#) expansion

References

1. ["GitHub"](https://github.com/haml/haml) (<https://github.com/haml/haml>). Retrieved 28 April 2018.
2. ["Changelog"](https://github.com/haml/haml/blob/master/CHANGELOG.md) (<https://github.com/haml/haml/blob/master/CHANGELOG.md>). Retrieved 28 April 2018.
3. ["Encoding"](http://haml.info/docs/yardoc/file.REFERENCE.html) (<http://haml.info/docs/yardoc/file.REFERENCE.html>). Retrieved 29 January 2016.
4. ["UTF encodings"](https://github.com/haml/haml/blob/88110b0607efca433c13bb1e339dcb1131edf010/lib/haml/util.rb) (<https://github.com/haml/haml/blob/88110b0607efca433c13bb1e339dcb1131edf010/lib/haml/util.rb>). Retrieved 7 February 2016.
5. ["About"](http://haml.info/about.html) (<http://haml.info/about.html>). Retrieved 28 April 2018.

6. "Trace Option in Haml" (<http://www.rubydoc.info/github/haml/haml/Haml/Options:trace>). Retrieved 16 February 2016.
7. "Using Haml" (http://haml.info/docs/yardoc/file.REFERENCE.html#using_haml). Retrieved 7 February 2016.

External links

- Official website (<https://haml.info>) 
 - Haml tutorial (<http://haml.info/tutorial.html>)
 - Learn Haml (basic) (<https://learnxinyminutes.com/docs/haml/>)
 - Haml (<https://github.com/haml/haml>) on GitHub
 - phpHaml (<http://phphaml.sourceforge.net/>), Haml implementation for PHP
 - Haml Google Group (<https://groups.google.com/forum/?hl=en#!forum/haml>)
 - Haml 1.0 (<http://weblog.rubyonrails.org/2007/1/19/haml-1-0>), announcement on the official Ruby on Rails weblog
-

Retrieved from "<https://en.wikipedia.org/w/index.php?title=Haml&oldid=1035215965>"

This page was last edited on 24 July 2021, at 10:54 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.