Preprocesadores

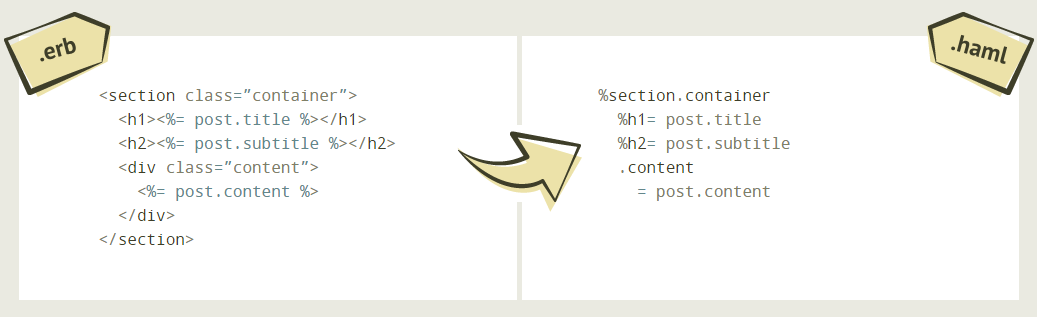
Un preprocesador es una herramienta que toma los valores introducidos y los transforma a otro estado. Puede ser que le des un archivo en SASS y lo transforma a CSS (o viceversa). El resultado de esa transformación es el archivo preprocesado.

Hay muchos tipos de preprocesadores y se pueden encontrar en esta página:

<http://riactive.mx/2013/09/16/preprocesadores-web/>

* HTML (HTML abstraction markup language):

Es un Markup language que se utiliza para limpiar y que sea simple el código HTML de cualquier documento web sin el uso de código en línea. Haml funciona como un reemplazo para sistemas de plantillas de página en línea tales como PHP, ASP y ERB, el templating language utilizado en la mayoría de aplicaciones de Ruby on Rails. Sin embargo, Haml evita la necesidad de forma explícita la codificación HTML en el template, porque él sí mismo es una descripción del HTML, con un código para generar contenido dinámico.



Principios básicos

Haml se basa en varios principios fundamentales para lograr lo que quieren que son:

* Marcado debe ser hermoso (Markup Should be Beautiful):

Marcado no debe utilizarse como una herramienta simple para navegadores para representar una página cómo el autor quiere que adopte. La representación no es lo único que la gente tiene que ver; tienen que ver, modificar y comprender así como el marcado. Así, el marcado debe ser tan fácil de usar y agradable como resultado prestado.

* Marcado debe ser seco (Markup Should be DRY):

HTML implica mayor repetición. Cada elemento se llama dos veces: una vez antes de su contenido y una vez después. ERB añade aún más repetición y caracteres innecesarios. HAML evita todo esto apoyándose en sangría, no texto, para determinar dónde comienzan a elementos y bloques de código y al final. No sólo tiene este resultado en las plantillas pequeñas, hace mucho más limpio que mirar el código.

* Debe ser marcado bien sangrada

Uno de los principales problemas de ERB es que no sólo no fomenta la código bien sangrada, activamente hace difícil o incluso imposible escribir. Esto nos lleva a HTML confusa, ilegible, cuando debería ser bien sangrada y reflejando la estructura subyacente del documento. Haml automáticamente correctamente formatos todas las etiquetas a eso él lo hace.

* Estructura HTML debe ser claro

XML y HTML son formatos basados en la idea de un documento estructurado. Esa estructura se refleja en su marcado y asimismo debe ser reflejada en meta-marcado como lógica de Haml porque Haml se basa en la muesca de elementos secundarios,esta estructura se conserva naturalmente, haciendo el documento mucho más fácil y más lógico para los seres humanos simples leer.

URL:<http://haml.info/>

* JADE:

Jade es un motor de plantillas de alto rendimiento muy influenciado por **HAML**, ha sido implementado con [**Javascript**](http://frontendlabs.io/tag/javascript/) para trabajar con [NodeJS](http://frontendlabs.io/tag/nodejs/). Adicionalmente es un pre-procesador de código [HTML.](http://frontendlabs.io/tag/html/)

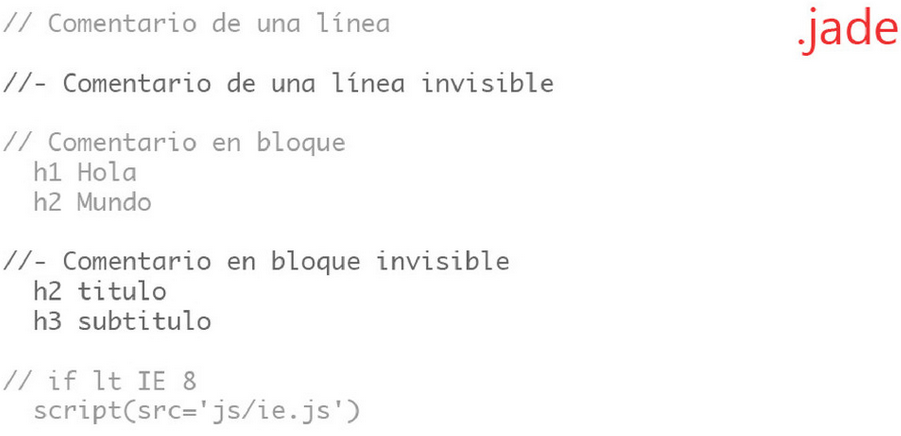
Ventajas:

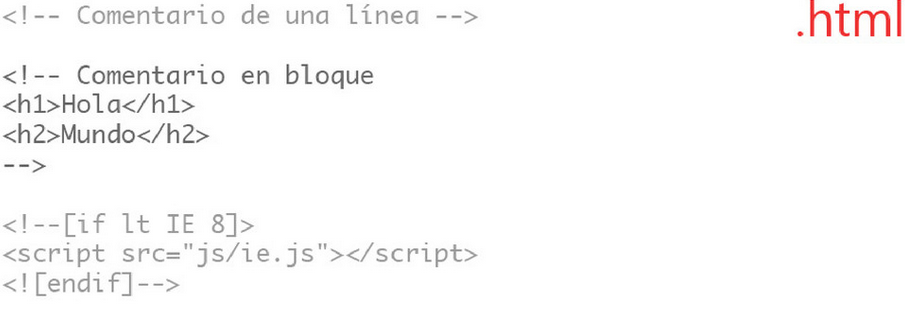
* Jade utiliza sangrías en vez de texto para determinar dónde empieza y en donde terminan los elementos y los bloques de código.
* Código limpio = algunas mejoras. Esto no solo da plantillas menos compactas sino que además hace que el código sea mas limpio para la vista humana.

Evita Bugs:

Jade nos mostrara una lista de errores encontrados y no generara ningun documento HTML hasta que lo hayamos resuelto.

Comentarios:





URLs: <http://frontendlabs.io/70--jade-language-node-template-engine-and-html-preprocessor>

<http://jade-lang.com/>

* Slim:

Slim es un lenguaje de plantillas, cuyo objetivo es reducir la vista sintaxis para las partes

esenciales sin llegar a ser críptico. Comenzó como un ejercicio para ver cuánto podría ser

eliminada de una plantilla estándar HTML (<,>, etiquetas de cierre, etc ...). A medida que más

personas se interesaron por Delgado, la funcionalidad creció y también lo hizo la flexibilidad de

la sintaxis.

Slim es un motor de plantillas rápido, ligero con soporte para Rails 3 y 4. Se ha probado en

gran medida en las principales implementaciones de ruby. Utilizamos integración continua

(travis-ci).

Sintaxis central de Slim es guiado por un pensamiento: "¿Cuál es el mínimo necesario para

hacer este trabajo".

A medida que más personas han contribuido a Slim, ha habido adiciones de sintaxis

influenciados por su uso de Haml y Jade. El equipo de Slim está abierto a estas adiciones

porque sabemos la belleza está en el ojo del espectador.

Slim utiliza Templo para analizar / compilación y también está integrado en la inclinación, por lo

que se puede utilizar junto con Sinatra o estante llano.

La arquitectura del templo es muy flexible y permite la extensión del proceso de análisis y

compilación sin mono-parches. Esto es utilizado por la lógica menos plugin y el plugin traductor

que proporciona I18n. En el modo de lógica menos puede utilizar Slim si te gusta la sintaxis

delgado para construir su HTML, pero no quieren escribir Ruby en sus plantillas.

¿Por qué utilizar Slim?

Slim le permite escribir plantillas muy mínimos que son fáciles de mantener y prácticamente

garantiza que usted escribe HTML bien formado y XML

Slim también es estético y lo hace mucho más divertido de escribir plantillas. Ya que se puede

utilizar Slim como un gota en el reemplazo de todo el marco principal se puede empezar

fácilmente.

La arquitectura Slim es muy flexible y permite escribir extensiones y plugins de sintaxis.

Slim fue desarrollado desde el principio con el rendimiento en mente. Los puntos de referencia

se realizan para cada confirmación en http://travis-ci.org/slim-template/slim. No te fíes de los

números? Eso es como debe ser. Por favor, intente la tarea rake referencia a ti mismo!

Sin embargo, en nuestra opinión debería usar Slim debido a sus características y sintaxis.

Acabamos de garantizar que Slim no tiene un impacto negativo en el rendimiento de la

aplicación.

Ejemplo:



URL:<http://slim-lang.com/index.html>

* Twig:

¿Qué es Twig?

Twig es un motor de plantillas construido en PHP. ¿Y qué es eso de un motor de

plantillas? Pues es una herramienta para eliminar la duplicación en HTML mediante la

reutilización de plantillas.

A la hora de construir una página web solemos tener en cuenta evitar la duplicación de

código en los controladores. ¿Pero qué pasa con los ficheros HTML? Si lo piensas,

tienen toneladas de código duplicado, como pueden ser los menús inferior y superior, el

sidebar lateral, la etiqueta <head> y todo su contenido, los ficheros js que incluimos al

final de la plantilla… Cada página de nuestra web debe incluir todos esos elementos, y

si no usamos un gestor de plantillas nos veremos obligados a copiarlos y pegarlos en

cada una de nuestras páginas.

Pues bien, Twig soluciona este problema permitiéndonos usar herencia en nuestras

plantillas, de forma muy similar a cómo se hace en la programación orientada a objetos.

Lo único que tenemos que hacer es identificar aquellos elementos de nuestra web que

sean comunes a todas las páginas, para de esa forma separarlos en una plantilla

propia, y luego diseñar el resto de plantillas simplemente heredando de la “plantilla

padre”.

Ejemplo:

1<!DOCTYPE html>

2<html>

3 <head>

4 <title>Mi web</title>

5 <meta name="viewport" content="width=device-width, initial-scale=1.0">

6 <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">

7 <link href="css/styles.css" type="text/css" rel="stylesheet">

8 <link rel="icon" type="image/x-icon" href="img/favicon.ico" />

9 </head>

1 <body>

0 <nav class="navbar">

1 <ul>

1 <li><a href="http://miweb.com/quienes-somos">Quiénes somos</a></li>

1 <li><a href="http://miweb.com/calendario">Calendario</a></li>

2 <li><a href="http://miweb.com/contacto">Contacto</a></li>

1 </ul>

3 </nav>

4 <div class="cuerpo">

1 {% block body\_content %}

5 {% endblock %}

1

6 </div>

1 <footer class="footer">

7 <ul>

1 <li><a href="http://miweb.com/aviso-legal">Aviso legal</a></li>

8

1 <li><a href="http://miweb.com/politica-privacidad">Política de privacidad</a></li>

9 </ul>

2 </footer>

0

2 <script src="js/jquery-2.1.0.min.js"></script>

1 </body>

2</html>

Resultado

1{% extends 'base.html.twig' %}

2

3{% block title %}

4 Mi web - Contacto

5{% endblock %}

6

7{% block body\_content %}

8 <!-- Formulario de contacto de la web -->

9{% endblock %}

* Handlebars:

URLs:<http://handlebarsjs.com/builtin_helpers.html>

<http://handlebarsjs.com/reference.html>