Jade笔记

一．基础

1.什么是jade

模板引擎 用js实现 供node使用

2.安装

npm install jade -g

3.测试

jade -h

4.使用

1.标签中有大段的块内容

在标签后面加.

每段前面加 |

2.使用（）来分割属性

3.一个id和多个class

a#download-btn.btn.blue-btn

4.循环 each VAL[,KEY] in OBJ

知识点

* 在 Express 中调用 jade 模板引擎
* jade 变量调用
* if 判断
* 循环
* Case 选择
* 在模板中调用其他语言
* 可重用的 jade 块 (Mixins)
* 模板包含 (Includes)
* 模板引用 (Extends)

在express中调用jade

Var express = require(‘express’) ; 引入express

Var Http = require(‘http’); 引入http

Var app = express();

App.set(‘view engine’, ‘jade’); 设置模板引擎

App.set(‘views’, \_dirname); 设置模板相对路径(相对当前目录)

* 1. App.get(‘/’, function(req, res){ res.render(‘test’)}); 调用当前路径下的test.jade模板

Jade变量调用

#(表达式)

=表达式

!=表达式

注意：符号 - 开头在 jade 中属于服务端执行的代码

= 会转义内容，不想被转义可以加 !=

变量转义

- **var** welcome = 'Hello <b>Wrold</b>'

**p** welcome <p>Hello World</p> html标签不赋值

p= welcome <p>Hello &lt;b&gt;Wrold&lt;/b&gt;</p> 特殊符号转义 能正确输出赋予的值

p!= welcome <p>Hello <b>Wrold</b></p> 特殊符号不转义 能正确输出赋予的值

if判断

- var user = { description: '我喜欢猫' }

- if (user.description)

h2 描述

p= user.description

- else

h1 描述

p 用户无描述

省略写法

- var user = { description: '我喜欢猫' }

#user

if user.description

h2 描述

p.description= user.description

else

h1 描述

p.description 用户无描述

循环

（1）

- var array = [1,2,3]

Ul

- for (var i =0; i < array.length; i++)

Li hello #( array[i] )

- }

（2）数组、 json 数据

ul

each val, index in ['西瓜', '苹果', '梨子']

li= index + ': ' + val

ul

each val, index in {1:'苹果',2:'梨子',3:'乔布斯'}

li= index + ': ' + val

Case （case 不支持case 穿透）

- var friends = 10

case friends

when 0

p you have no friends

when 1

p you have a friend

default

p you have #{friends} friends

简略写法：

- var friends = 1

case friends

when 0: p you have no friends

when 1: p you have a friend

default: p you have #{friends} friends

在模板中调用其他语言

可重用的 jade 块 (Mixins)

（1）

声明可重用的块

Mixin list

Ul

Li first

Li second

Li third

调用

+list()

（2）可以给这个重用块指定参数

mixin pets(pets)

ul.pets

- each pet in pets

li= pet

+pets(['cat', 'dog', 'pig'])

1. Mixins 同时也支持在外部传入 jade 块

Mixin article(title)

.article

.aricle-warpper

H1= title

//- block 为jade关键字，代表外部传入的块

If block

Block

Else

P 该文章没有内容

调用 ： +article(‘hello world’)

<div class="article">

<div class="article-wrapper">

<h1>Hello world</h1>

<p>该文章没有内容</p>

</div></div>

+article(‘hello jade’)

P 这里是外部传入的块

P 再写两句

结果：

<div class="article">

<div class="article-wrapper">

<h1>Hello Jade</h1>

<p>这里是外部传入的块</p>

<p>再写两句</p>

</div></div>

1. Mixins 同时也可以从外部获取属性

mixin link(href, content)

a(class!=attributes.class, href=href)= content

+link('/foo', 'foo')(class="btn")

<a href="/foo" class="btn">foo</a> 结果

模板包含 (Includes)

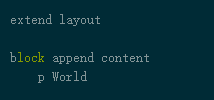
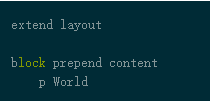
模板引用 (Extends)

通过 Extends 可以引用外部的 jade 块 （会覆盖原来代码）

1. 如何追加代码片段

Append : 在原有代码之后追加

Prepend:在原有代码之前追加

注释

Jade 支持两种注释：单行注释和多行注释。

每种注释支持两种模式：输出到源文件和不输出到源文件。

// 输出扫编译后的文件

//- 不会输出扫编译后的文件

修改style

A( style = { color: ‘red’, background: ‘green’ } )

文本

Jade 支持三种文本输出方式：单行文本、管道文本和多行文本

