

jStyle, 基于 JavaScript 的高级 CSS 呈现引擎

目录

jStyle, 基于 JavaScript 的高级 CSS 呈现引擎	1
jStyle 的特点	1
基本概念	1
使用方式	2
运行机制	4
入门演示	5

jStyle, 是一个基于 JavaScript 的高级 CSS 呈现引擎, 是一个高性能、简单易用的 JavaScript 引擎, 它的语法灵活而简约, 可以帮助你快速而简便地开发具有复杂界面和逻辑的网页应用。同时, jStyle 的设计基于简单有效的理念, 其实现机制和 API 都很简单易懂, 任何非必要的程序功能和部件都已被舍弃, 因此其他开发者可以很方便的对 jStyle 进行样式扩展, 给使用者提供更多的样式可用。jStyle 体积非常小, 经过压缩的脚本才 14K(不包含 Sizzle, 版本 1.00)。

jStyle 的特点

- I 简单易用:
使用 jStyle, 你可以很简便的开发出具有复杂界面和逻辑的网页应用。
- I 支持 CCS3 选择器:
jStyle 使用支持 CCS3 的 Sizzle^[1] 作为 CSS 选择器的引擎。因此也支持 CSS 3 选择器^[2]!
- I 跨浏览器支持:
jStyle 支持的浏览器有: IE 6.0+, FF 2+, Safari 3.0+, Opera 9.0+, Chrome。
- I 易于扩展:
jStyle 提供简单而有效的接口给开发者来进行扩展。

基本概念

为了便于描述 jStyle, 首先介绍一下几个基本概念。

jStyle 样式

jStyle 样式是 jStyle 的一个基本单位。一个 jStyle 样式对应一组(一个)特定的界面和逻辑功能。例如 jStyle 样式 validation 对应于对 HTML 元素的录入进行验证的一组功能。

为了简化描述, 下文中对“ jStyle 样式”和“样式”混用, 均表示“ jStyle 样式”

样式参数

样式参数是所属样式的一组参数。可以通过设置样式参数来对样式进行具体的设置。例如可以通过设置元素的样式 validation 的参数 required 为 true 来指定该元素的验证样式规则为必须录入。

样式定义

样式定义是对元素 jStyle 样式的语法描述。引擎会根据元素的样式定义来应用样式。关于样式定义请参考下文的 jStyle [使用方式](#) 部分。

jStyle 元素

已经定义过 jStyle 样式的元素称为 jStyle 元素，引擎在呈现 jStyle 元素样式的时候，会把样式定义转换为标准的 JavaScript Object 格式，赋予 jStyle 元素名为"jStyle"的属性。我们可以在 JavaScript 中通过 element1.jStyle 来获取元素 element1 的样式定义。注意，有些样式为了避免前后不一致，会在应用样式后删除原样式定义。

注册样式

每个已定义的样式，都有一个内置函数 register();该函数用于在浏览器初始化样式的时候执行特定的操作来使得样式对应的功能生效。例如为了使得 validation 样式生效，需要在初始化 validation 样式的时候，注册 Forms 提交事件，用来捕获元素验证的反馈结果。

样式参数构建

对于已定义的样式的参数，均有一个内置函数 builder(style, element);该函数用于对已经定义了本样式参数的元素 element，应用参数定义 style.应用参数定义的时候，可能会根据元素已有的本参数的定义来做出相应的动作。例如，对于样式 validation，如果某一个元素 element1 有样式定义 jstyle="validation({required:true})",那么 jStyle 就在构建样式参数 required 的时候，执行本样式参数的构建函数 builder(true, element1)，在 element1 的旁边显示一个红色叹号图标，提示用户 element1 必须输入值。

呈现样式

每个已定义的样式，都有一个内置函数 render(style, element);该函数用于对已经定义了本样式的元素 element，应用样式定义 style。应用样式定义的时候，可能会根据元素已有的 jStyle 属性来做出相应的操作。例如，对于样式 CCS，如果某一个元素 element1 有样式定义 jstyle="css:{display:'none'}",那么 jStyle 就在呈现样式的时候，执行本样式的函数 render({display:'none'}, element1)，把 element1 改为不显示。

样式配置

样式配置是指对一特定 jStyle 样式的全局定义。它描述了样式的名称，样式的参数，样式怎么样构建参数，怎么样呈现样式，怎么样注册样式等等。

使用方式

jStyle 支持三种方式来进行样式定义。

第一种方式

通过在 HTML 代码中对指定元素设置名称为"jstyle"的属性来进行样式定义。

例如：

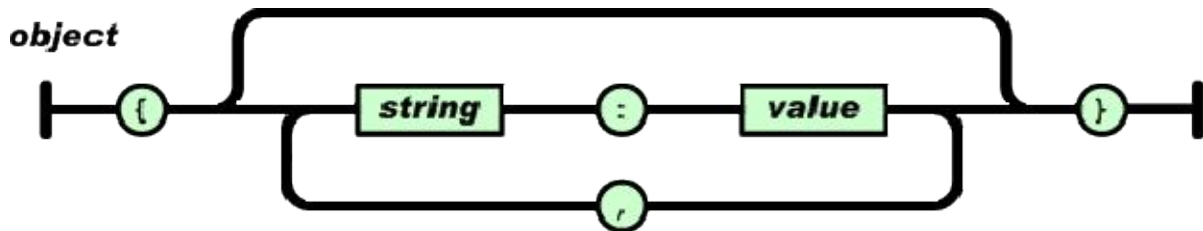
```
<input jstyle="validation :{required:true,email:true}" type="text">
```

就表示对 input 元素指定名称为 validation 的 jstyle 样式。

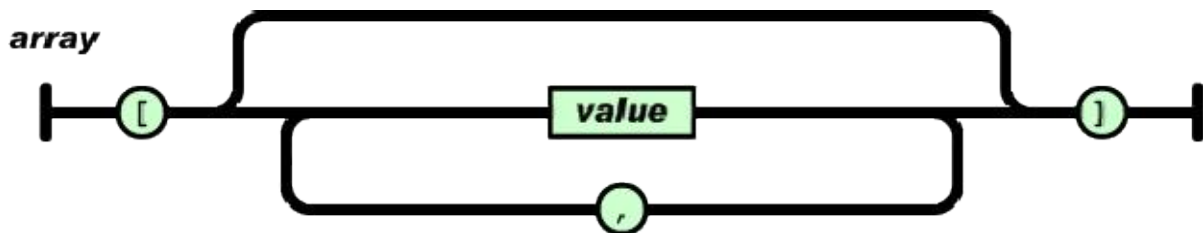
注意，这里的 jStyle 描述符的语法为，jstyle="style1: Object,style2: Object,..."。

其中，Object 表示标准的 JavaScript 对象。style1,style2 为 jStyle 样式名称。

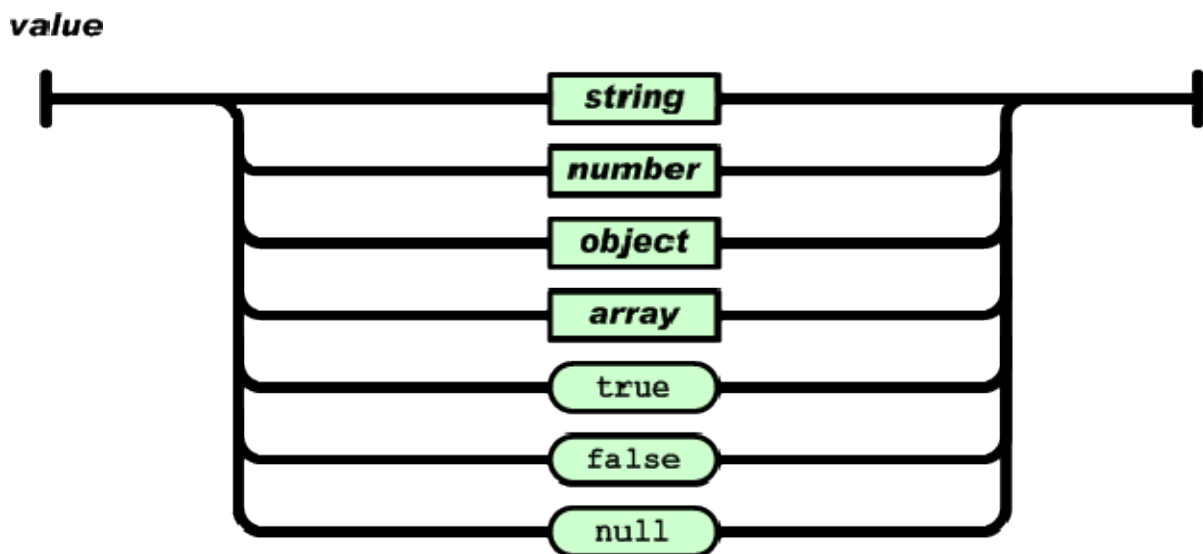
Object (对象)是一个无序的“名称/值”对集合。一个对象以“{”（左括号）开始，“}”（右括号）结束。每个“名称”后跟一个“:”（冒号）；“名称/值”对之间使用“,”（逗号）分隔。[3]



数组是值（value）的有序集合。一个数组以“[”（左中括号）开始，“]”（右中括号）结束。值之间使用“,”（逗号）分隔。



值（value）可以是双引号包括起来的字符串（string）、数值(number)、true、false、 null、对象（object）或者数组（array）。这些结构可以嵌套。



上例中，样式属性设定{required:true,email:true}，表示 input 元素必需输入，输入必需为 email 格式。这是 jStyle 主要的样式定义方式。

通过这种方式来定义样式是最为高效的，它不需要用户编程，只需要在 html 中设定相应的 jStyle 描述符，就可以实现 jStyle 所定义的复杂界面和逻辑，因此可以大大提高网页开发的效率。

第二种方式

通过使用 JavaScript 函数 `cs$(selector,stylepath,jstyle)`来定义样式。

函数中：

参数 selector 的格式为 {String 或者 DOMELEMENT 或者 DOMELEMENT 数组 }，如果是 String 格式，应该为 CCS3 的选择器格式。这里的 DOMELEMENT，代表 HTML 文档 Dom 树上的节点元素。

参数 stylepath 的格式为 {String}，自定义要设定 jStyle 样式的路径，例如 "validation.required"表示对

validation 样式的 required 参数进行设置。

参数 jstyle 的格式为 JavaScript Object, 表示对样式路径 stylepath 的设置。

例如:

```
cs$("input","validation.required",true);
```

表示对所有 input 元素都需要验证其必须输入。

```
cs$("#input1","render.dialog.url","dialog_input.html");
```

表示对设置 id 为 input1 的元素的 render 样式的样式参数 dialog 的 url 属性为"dialog_input.html"。

第三种方式

通过使用 JavaScript 函数 cs\$(selector).aStyle(arguments)来定义样式。

函数中:

参数 selector 的格式为 {String 或者 DOMElement 或者 DOMElement 数组 }, 如果是 String 格式, 应该为 CCS3 的选择器格式。

函数 aStyle 对应 jStyle 的一个样式的名称。

参数 arguments 表示函数 aStyle 的参数列表。它可能有一个或者多个参数, 每个参数的格式均为 JavaScript Object。参数之间用","分割。

例如:

```
cs$("input").validation({required:true,email:true});
```

表示对所有 input 元素都需要验证其必须输入, 还要验证输入是否为 email。

```
cs$("input").classes("+","class1");
```

表示对所有 input 元素都添加类"class1"。

运行机制

jStyle 的运行机制基于简单有效的理念来设计。

jStyle 使用两种方式来应用样式。

A. Window 加载完成时应用样式.

在 window 加载完后, jStyle 引擎会对使用第一种方式 (参考[使用方式](#)) 来对定义了 jStyle 样式的元素应用样式。过程如下:

- 1 找出所有 (使用第一种方式) 定义了 jStyle 样式的元素。
- 2 把元素转换成 jStyle 元素。
- 3 循环对每一个样式配置 (假设为 jstyle1), 应用 jStyle 元素的样式。
 - 3.1 在所有 jStyle 元素中找出所有定义了 jstyle1 样式的元素
 - 3.2 循环对每一元素 (假设为 element1), 应用样式 jstyle1。
 - 3.2.1 对 element1 所有已定义了的 jstyle1 样式参数执行样式参数构建。
 - 3.2.2 对 element1 执行呈现样式 jstyle1 操作。
 - 3.3 注册样式 jstyle1.
- 4 结束.

B. 运行时应用样式.

jStyle 也支持在运行时使用第二种、第三种方式, 通过执行 JavaScript 代码来对元素 (假设为元素 element1, 假设样式定义为 style1) 应用样式定义 (参考[使用方式](#))。过程如下:

- 1 设置 element1 为 jStyle 元素
 - 1.1 如果元素不是 jStyle 元素, 把元素根据样式定义转换成 jStyle 元素,

- 1.2 如果元素已经是 jStyle 元素，就把 style1（覆盖性）附加到元素的 jStyle 属性上去。
- 2 找出 style1 所对应的样式配置（假设为 jstyle1）。
- 3 对 element1 应用样式 jstyle1。
 - 3.1 对 element1 所有已定义了的 jstyle1 样式参数执行样式参数构建。
 - 3.2 对 element1 执行呈现样式 jstyle1 操作。
- 4 结束。

入门演示

这里有 jStyle 引擎使用的入门演示文档（英文版），供给各位参考。

链接：http://www.ccpup.com/jstyle/docs/quick_start.html

参考：

[1]. Sizzle website <http://sizzlejs.com/>

[2]. CCS 3 Selectors <http://www.w3.org/TR/css3-selectors/>

[3]. Introducing JSON <http://www.json.org/index.html>