

jStyle, 基于JavaScript的高级CSS呈现引擎

jStyle (网站地址 <http://www.ccpup.com/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

- 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。
- 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]!
- 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。
- 易于扩展:
jStyle提供简单而有效的接口给开发者来进行扩展。

预览

在传统的网页应用中。为了呈现一些CSS标准之外的样式或者实现一些复杂的呈现逻辑, 往往需要编写大段的JavaScript代码来实现。

举个例子, 有表单form1, 输入控件input1, 现想在提交表单form1之前, 验证input的输入是否整数, 一般的代码如下:

```
<form id="form1" onsubmit="return checkInputs();">
<input id="input1" type="text" value="" />
</form>
<script type="text/javascript">
function checkInputs(){
    var input1=document.getElementById("input1");
    if(input1.value>" " && !input1.value.match(/^[^-+]?[0-9]+$/i))
    {
        alert("请输入整数");
        input1.focus();
        return false;
    }
    return true;
}
</script>
```

而使用jStyle实现同样的例子, 确实简单的多。有两种方式都可以实现。

第一种方式是设置input元素的属性, 代码如下:

1. `<input id="input1" jstyle="validation:{integer:true}" type="text" value="" />`
第二种代码更为简单，是使用JavaScript代码，如下：
2. `cs$("#input1").val({integer:true});`

基础概念

为了便于描述jStyle，首先介绍一下几个基础概念。

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。

样式序列化

样式序列化是指把静态和运行时产生的jStyle元素的样式序列化到元素的jstyle属性中,以方便某些应用在再次加载网页的时候可以重新应用元素已生成的jStyle样式。

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">
```

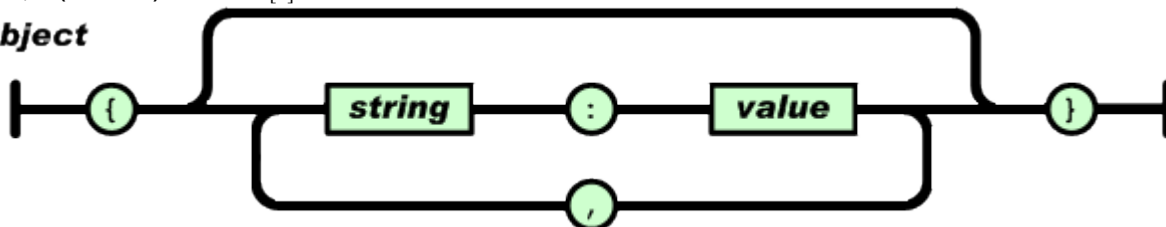
就表示对input元素指定名称为 validation的jstyle样式。

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。

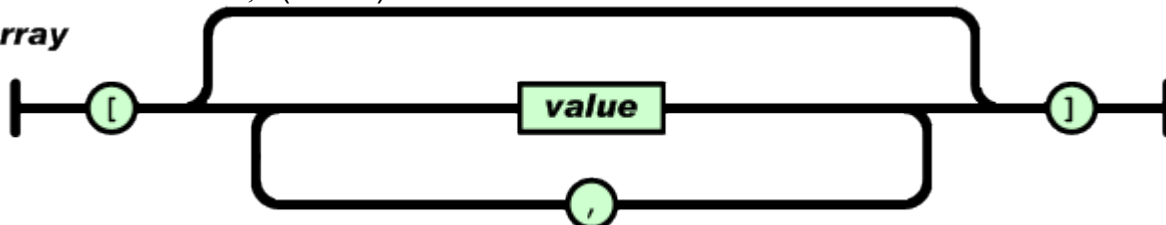
Object (对象)是一个无序的“名称/值”对集合。一个对象以“{”(左括号)开始,“}”(右括号)结束。每个“名称”后跟一个“:”(冒号);“名称/值”对之间使用“,”(逗号)分隔。^[3]

object



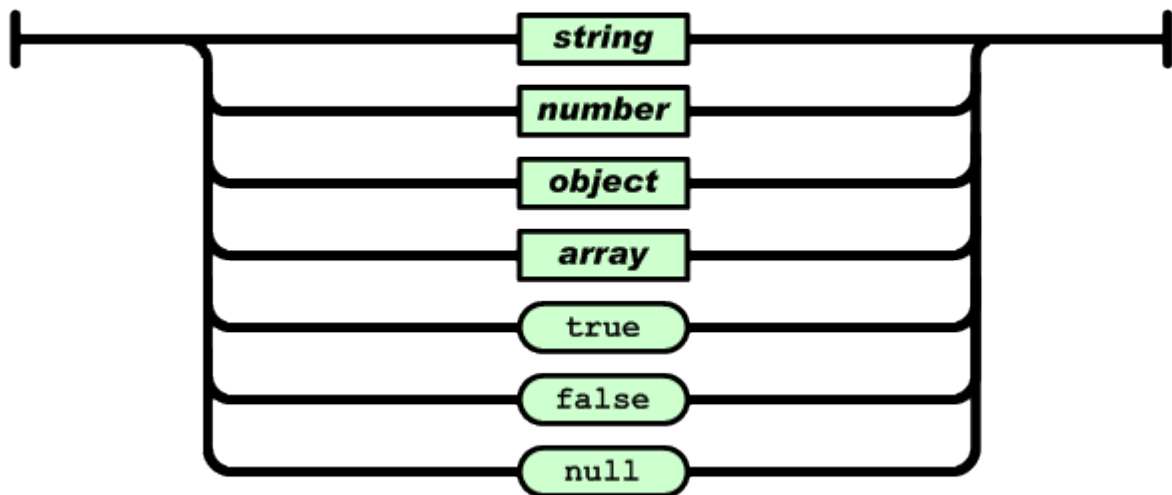
数组是值 (value) 的有序集合。一个数组以“[”(左中括号)开始,“]”(右中括号)结束。值之间使用“,”(逗号)分隔。

array



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。

value



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. 对所有有效的样式配置进行注册。
2. 从document.body开始，后根遍历整个文档树。
 1. 如果元素(假设为element1)（使用第一种方式）定义了jStyle样式，把它转换成jStyle元素。否则返回。
 2. 循环对每一个样式配置（假设为jstyle1），对element1应用样式。
 1. 对element1所有已定义了的jstyle1样式参数执行样式参数构建。
 2. 对element1执行呈现样式jstyle1操作
3. 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. 如果元素不是jStyle元素,把元素根据样式定义转换成jStyle元素，
 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。
 1. 对element1所有已定义了的jstyle1样式参数执行样式参数构建。
 2. 对element1执行呈现样式jstyle1操作。
4. 结束。

典型应用

1. 纯HTML实现的公式显示和编辑（开发计划中）

预览网址<http://www.ccpup.com/jstyle/js/formula/test.html>

2. HTML文档部分内容评论（开发计划中）

预览网址<http://www.ccpup.com/jstyle/js/partialcomments/test.php>

相关文档

- 入门演示文档 (英文版) http://www.ccpup.com/jstyle/docs/quick_start.html
- API 手册 (JsDoc) <http://www.ccpup.com/jstyle/docs/jsdoc/index.html>
- 扩展jStyle示例程序 <http://www.ccpup.com/jstyle/docs/jstyle.extend.js.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>