

基于active,checked等状态类名的web前端交互开发

这篇文章发布于 2016年10月30日, 星期日, 00:00, 归类于 Web综合。阅读 20612 次, 今日 6 次 31 条评论

by zhangxinxu from <http://www.zhangxinxu.com/wordpress/?p=5716>

本文全文转载需购买版权(500¥), 摘要引流则免费, 具体参见[这里](#)

一、热身开始

在我们进行web前端开发的时候, 恩, 换种接地气的说法, 在我们进行网页制作的时候, 一定会存在各种交互效果, 比方说, 选中高亮, 点击显示更多, 选项卡切换等等。

我可以拍着胸脯说, 绝对多数前端人员在实现这些交互效果的时候, 是没有什么规范或者准则的。比方说点击显示更多, 有些人可能直接jQuery `$.show()` 方法, 有些人可能会写个类似 `.more-show` 的类型控制显示, 具体使用什么方式, 用什么样的命名, 完全要看网上搜到的代码是如何实现的, 以及看当时的心情。

代码如产品, 要想做出好产品一定要细心打磨, 同样的, 如果要想写的好代码也需要细心打磨。所谓打磨, 就是让一些“糙”的地方变得流畅。

好产品的意义在于用户和企业受益, 好代码的意义在于自己的寿命以及同事关系。

交互效果的实现方式随性, 就是代码“糙”的地方, 我们其实可以约定一套通用的规范或者准则, 这一块变得流畅。虽然说从产品的角度讲, 意义不是很大, 但是对于我们开发人员自己或者一起合作的同事而言, 却是很有裨益的。因为, 可以降低我们日后维护的成本, 同时有其他一些非常受用的益处。需求总是要变更了, 迭代总是要进行的。好的代码准则可以让你不怕需求变更, 不要加班修“霸哥”。和产品成为好盆友, 每天可以早早回去陪女朋友看电视, 岂不很好!

我经过这些年的实践和提炼, 对于交互效果的实现提炼了一套实践准则, 至少在我目前这个阶段, 觉得是很有裨益的, 这里跟大家分享下。

如果用一句话概括下这个交互实现准则就是: 完全基于active,checked等状态类名的web交互开发

二、进入正题: 基于active,checked等状态类名的web交互开发

那什么是“基于active,checked等状态类名的web交互开发”呢?

我们拿最简单的“点击显示更多”效果举例, 比方说下面这种现在见的比较多的查看更新信息的效果:



默认状况下, 容器会设定一个高度, 例如 `80px`, 然后 `overflow:hidden`, 类似下面的CSS:

```
.box {  
  height: 80px;  
  overflow: hidden;
```

```
}
```

此时要实现点击展开效果，其实很简单，比较偷懒的做法就是JS一步搞定：

```
more.onclick = function () {  
    box.style.height = 'auto';  
};
```

也有喜欢用CSS类名控制的小伙伴，此时会发挥自己在CSS命名上的天赋，来一个 `.height-auto` 的类名：

```
.height-auto {  
    height: auto;  
}
```

于是，JS代码变成：

```
more.onclick = function () {  
    box.className = box.className + ' height-auto';  
};
```

其实从产品角度讲，上面两种实现都无伤大雅，都是不错的实现，但是从代码层面讲，则均有不足之处。

1. JS直接控制样式的不足

由于我们网页大部分的样式都是CSS控制的，一旦JS也直接参与样式控制，CSS和JS就存在交叉关系，用一个软件术语描述就是“耦合了”，一旦两个事物发生耦合，势必会增加日后的维护成本。举个例子，产品上线后，设计师突然希望这里的展开不要这么的硬，希望可以有点CSS3动画效果。请问，此时，你当如何维护，是不是CSS既要修改（增加动画控制），JS也要修改（设置 `height` 为 `auto` 不会触发CSS3动画），看，要修改两个地方。如果我们一开始所有的样式都值交给CSS控制，只需要改一个CSS文件就可以了。同时，考虑到，很多公司写CSS的和写JS的不是同一个人，如果CSS和JS同时修改，岂不是要动用两个人力参与，这用人的成本和周期就上去了。虽然，计算下来，都是小成本，但是，但我们埋头某项目或者某问题的时候，被这种小需求插入，其实是很损耗精力的。如果这类小打小闹（尤其不是自己本职的）总能避免，工作岂不会轻松很多。

2. 命名语义或随意的问题

类名样式语义本无可厚非，但是，对于JS交互效果实现而言，语义化就是问题所在。语义化的类名往往代表了自身是有CSS样式的，例如，上面的 `.height-auto`，但是，由于我们这种添加类名的行为是在JS代码中完成的，因此，本质上，

```
more.onclick = function () {  
    box.className = box.className + ' height-auto';  
};
```

和

```
more.onclick = function () {  
    box.style.height = 'auto';  
};
```

几乎没有任何区别。这个点很重要，大家再仔细想想，这两者是不是没有本质的区别。还是举例说明，同样的，设计师突然希望这里的展开不要这么的硬，希望可以有点CSS3动画效果。虽然说，从技术的角

度讲，我们只需要修改CSS代码就可以了，但是，这个 `.height-auto` 命名...如果我们把里面的样式改成了CSS3动画相关内容，是不是就牛头不对马嘴了？是不是要去JS中把这个类名改成 `.height-animate` 之类的。看，最后还是改了两处。

另外还有一个看上去不是问题的问题，那就是，一个页面往往会有很多很多的交互效果，如果没有交互效果都有一个对应的类名控制，那岂不是JS文件中有很多样式控制的类名。很显然的，这东西一旦多了之后啊，自然也就不好维护了。

—————低调的分隔线—————

但是，如果我们采用“基于active,checked等状态类名的web交互开发”，就不会有上面JS和样式耦合的问题。如何做呢？很简单，就是，我们控制样式的类名使用 `.active`，`.checked` 等这种状态类名，例如：

```
more.onclick = function () {  
    box.className = box.className + ' active';  
};
```

并且，网页所有的页面交互都使用这种状态类名，根据我自己的实践，就算所有交互都使用 `.active` 这同一个类名都是可以的，而且事实上是推荐这么做的。

注意，下面这条准则很重要，很重要，是本文的精髓——

.active等JS交互类名自身绝对...绝对不能有CSS样式

再说一遍，自身无样式，就是一个状态标识符，用来和其他类名发生关系，然后让其他类名的样式发生变化。这种关系可以是父子，兄弟或者自身。

不急，我们先看一个例子，您可以狠狠地点击这里：[基于.active类名的显示更多交互demo](#)

默认是这样的：



当我们点击更多按钮后，会以CSS3动画的方式展开剩余所有文字信息，如下动图：



下面我们看下关键的CSS代码，这个 `.active` 是如何自身无样式的：

```
.box {  
    max-height: 80px;  
    transition: max-height .25s;  
    overflow: hidden;
```

```
}  
.box.active {  
  max-height: 200px;  
}  
.active > .more {  
  display: none;  
}
```

可以看到，当我们点击按钮后，盒子变高，以及更多元素隐藏，全部都是通过和 `.active` 发生关系后发生的，而不是 `.active` 自己的样式。例如，我们盒子实现的 `200px` 以内任意高度的动画效果，是通过 `.box.active` 组合类名触发的，用中文解释就是 `.box` 元素同时有 `.active` 状态的时候，样式如何如何...

同样，`.active > .more` 表示，当我的爸爸状态为 `.active` 的时候，我隐藏。

`.active`本质上扮演的角色不是类名样式，而是一个标识符！

—————低调的分隔线—————

有人可能会疑问，为什么要这样处理呢？

还记不记得 `.height-auto` 命名问题所在，由于这个类名本身牵扯样式，因此，本质上，还是把CSS样式牵扯到了JS文件中。但是，如果是一个完全没有独立样式的状态标识符，则 `.active` 在JS中扮演的角色也就成了一个完完全全真真切切的状态占位符了！也即是说，虽然我们的JS文件中出现了类名，但是，就如同 `.JS_xxx` 类名作为DOM选择器一样，本身适合样式分离的，实现了真正意义上的样式和行为分离！

比方说，技术总监突然觉得这里展开更多有动画效果太啰嗦，要直接展开，要更硬朗，需要进行改动怎么办？你会发现，我们根本无需劳烦JS开发工程师，直接页面仔改一下CSS文件就可以了，例如，回到一开始的 `height:auto`：

```
.box {  
  height: 80px;  
  overflow: hidden;  
}  
.box.active {  
  height: auto;  
}
```

三、高潮开始：基于active状态类名的web交互开发更炸裂的优点

采用“基于active状态类名的web交互开发”除了上面提到的“CSS样式和JS行为分离”有利于日后维护的优点外，还有很多其他炸裂的优点：

1. 省了很多想命名的脑细胞

妈妈再也不用担心有些单词不知道怎么翻译啦~~

2. CSS和JS代码可读性更强了

一旦在CSS或JS中看到 `' .active '`，大家都知道，页面这块内容要变形了；

3. JS代码量更少了

例如，我们在全局，或者顶层局部定义这么一个变量：

```
var ACTIVE = 'active';
```

则，由于我们所有的交互都只要这一个类名，因此，当我们JS压缩的时候，其压缩率要高很多，也更好维护。加入说页面中有10个UI交互效果，原先是10个不能压缩的字符串，现在变成了1个，代码量自然要少。

4. 类名压缩成为了可能

我几乎从未见到国内的网页产品HTML中的类名是有压缩的，如果 `.aaa-bbb-ccc` 这种类名可以压缩成1~2个字符，那岂不是CSS和HTML压缩率会更喜人，同时，我的[CSS命名最佳实践（无ID，无层级，无标签）](#)唯一的HTML中className会比较多解决的问题也解决了。

类名压缩最大的阻碍，在我看来，就是我们小伙伴在实现JS交互效果的时候不注意，常常把带有CSS样式的类名混在JS文件中，并且命名随意，并且，会把类名字符串进行分隔处理，尤其一些网上的UI组件，类似：

```
var classNameRoot = 'swipe-slide-';
```

然后，通过这个类名前缀，拼接其他类名，你说，这该如何准确压缩。

但是，如果大家遵循“基于active状态类名的web交互开发”，JS文件中只有不参与CSS样式的类名选择器，以及不参与CSS样式的交互效果占位符，则我们就可以通过简单的设置不过滤的类名，实现我们的类名压缩效果了。例如在 `config.js` 中：

```
{
  "compressClassName": true,
  "ignoreClassName": ["active", "disabled", "checked", "selected"]
}
```

让我们的类名压缩工具不压缩这些类名，则我们完全不要关心JS，也就是无需去截自页面使用的JS资源，只要安安心心压缩CSS文件中的类名，同时对我们页面中使用的类名进行压缩就可以了，一个想实现很多年的东西就通过这么个实现准则约束给实现了。

实现了？对，实现了。

由实习生完成的我们阅文集团新的招聘官网已经可以外网访问了，不过还有些细节需要打磨，所有老站还没做301跳转，现在我是悄悄的告诉大家地址的，大家这几天低调传播，什么时候高调传播可以关注我的微博：<http://join.yuewen.com/>

整个招聘站的HTML类名（如果CSS有使用）全部进行了2字符的压缩（2字符显得更规整，我心里舒服），如下截图：



经测试，CSS大小可以降低10%，HTML则受标签多还是内容多影响，也是有明显的体积减小的。

四、缓缓收尾：为何以前没有这样的实践准则

“基于active状态类名的web交互开发”在多年之前实际上是很难实践下去的，因为，该准则依赖的CSS关系符选择器，而几乎所有的CSS关系符选择器都是从IE7浏览器开始才支持完全的。

例如：`.active.xxx` 之前在IE6下是有bug的，顺序没放好，可能会发生 `.active` 冲突的情况；相邻父子关系 `.active > .xxx` IE7浏览器才开始支持，由于我们网站几乎所有普通交互都使用这同一个 `.active` 类名，因此，不能简单使用父子选择器 `.active .xxx` 有一定概率会发生冲突，使用相邻父子

关系 `.active > .xxx` 约束则无此问题；相邻兄弟选择器 `.active + .xxx` 以及普通兄弟选择器 `.active ~ .xxx` 全部都是IE7浏览器才开始支持。也就是，浏览器的发展催着这这种更好实践的成型和普及。不好意思，说错了，应该还没普及，我是希望可以从本文开始慢慢普及。

我想，现在基本上很少有公司还要兼容IE6的了，如果你的产品还要做这样的事情，我只能扼腕而叹了。

五、缓缓收尾：我常用的一些状态类名

我简单回顾了下，我使用的状态类名最多的就是 `.active` 以及 `.disabled`，在单复选框UI组件使用过 `checked`，自定义下拉使用过 `selected` 等，其他几乎鲜有使用。

另外，还有其他一些状态类名，当然，是我见别人用过的，有 `.on`，以及 `.in` 和 `.out`。前面的 `.on` 等同于我习惯使用的 `.active`，因为 `.on` 的状态含义更广，例如单选框选中用 `.on` 表示就挺好，但是如果是 `.active` 则似乎则是另外的意思了，所以，如果大家希望偷懒，建议可以试试 `.on` 这个状态标识符，真正的可以贯穿所有交互的状态类名；后面的 `.in` 和 `.out` 是用来标示CSS3 `animation` 动画状态的，也属于状态类名。

等等。

六、最终收尾

中国有句成语，就做“立竿见影”，比喻效果很明显。

如果从降低维护成本的角度讲，折腾那些node前端工具流所节约的开发和维护成本确实是立竿见影，相比之下，本文的这点东西（如果不是后面的类名压缩撑腰）节约的开发和维护成本简直就是不值一提。

这其实是一件很悖论的东西。从企业的角度讲，自然是希望员工能够把好钢用在刀刃上，以最小的力带来最大的收益，优先解决最为瓶颈的东西上，所以，如果大家比较看重职级，抬头，薪资这样的东西，建议更多精力花在这些“立竿见影”的事情上，本文的这些内容呢，大可不必较真。但是，从工程师的角度讲，指有责任心的工程师，会对技术和代码品质很有追求，会化很多精力去磨代码，这其实是很棒的一件事情，就好比本文所说的这些代码准则。然而，问题在于，工程师花了很多精力做这样的事情，其带来的价值往往并不是立竿见影的那种，需要时间来证明。从企业的角度讲，这样的工程师并不一定是最想要的工程师，尤其中小企业。企业属于商业机构，不是学术交流场所，看重利益，所以，很多做技术的同学一定要认清这一点，如果你在意公司对你的认可，做出立竿见影的成绩来说话；如果你觉得自己技术很不错，但升职加薪的不是我，想想看是不是自己的认知层出现了偏差，是不是认为代码最棒的就应该工资最高。当然，如果你就是希望技术纯青，对世俗这些不care，则建议你要一直坚定的走下去，每天成长每天提高，通过技术给企业给同事给同行带来更加沉稳更加厚重的收益，假以时日，你一定可以走出和别人不一样的康庄大道，要知道，决定一个人最终成功的不是智商，而是毅力。这些话既是鼓励也是自勉。

对了，对于前文提到的我自己写的类名压缩工具，目前暂不开放，除了精力有限的原因外，最主要原因是.....想了想，归根结底还是精力有限，嫌麻烦。你看我10月份就2篇文章，忙得连鱼缸的鱼都没时间喂。

恩，就这些，欢迎交流！



《CSS世界》签名版独家发售，包邮，可指定寄语，点击显示购买码

（本篇完） // 想要打赏？点击[这里](#)。有话要说？点击[这里](#)。



« 巧用浏览器CSS属性值的不兼容向下兼容hack技巧

HTML可访问性fieldset,legend元素及CSS布局应用 »

猜你喜欢

- 精简高效的CSS命名准则/方法
- 了解web前端领域的undefined behavior未定义行为
- 翻译-高质量JavaScript代码书写基本要点
- 页面可用性之outline轮廓外框的一些研究
- CSS+JavaScript实现页面不同布局的切换
- HTML5 DOM元素类名相关操作API classList简介
- CSS样式分离之再分离
- CSS的样式合并与模块化
- 自己以前架构CSS/JS时候挖过的一些坑
- 高富帅seajs使用示例及spm合并压缩工具露脸
- Chrome下语音识别实现页面快速引导以及重要交互

分享到:  0

标签: active, className, 交互, 压缩, 命名, 样式分离, 规范

发表评论（目前31条评论）

<input type="text"/>	名称 (必须)
<input type="text"/>	邮件地址(不会被公开) (必须)
<input type="text"/>	网站
<div></div>	
<input type="button" value="提交评论"/>	

1. **wl22**说道:
2017年09月7日 15:29
我一直是用w-ac,w-di,w-first,w-last这样的类控制。
[回复](#)



2.

风儿寻说道：

2017年06月30日 14:05

为什么我写了一遍，没有展开效果呢

回复


3.

小小说道：

2017年01月5日 14:21

大神，你用的那个压缩工具呢？ 求教！

回复


4.

韩长寿博客说道：

2016年11月27日 17:47

不错支持

回复


5.

小儿来二斤酱牛肉说道：

2016年11月17日 17:27

楼主的想法挺好，用一个类来做状态，通过父子 兄弟关系来 改变一些交互效果，但是对于一下简单的交互，例如展开，隐藏等，楼主还是用到了js来控制（增加click事件），这算不算“耦合”，对于这些 简单的交互效果，我觉得 不用js来做 更好，完全由html标签和css样式 来控制交互。
我是用 这个标签 来做，通过控制checked,来关联 父子 和 兄弟样式，然后把他透明度设置为0 来实现点击切换，

回复


6.

旧时长安说道：

2016年11月15日 17:08

这是第一次评论，之前在写效果的时候，自己也用到了本文所提到的方法，但是没有系统的整理过，今天，看了这篇文章，感觉思路更清晰了

回复


7.

墨染说道：

2016年11月15日 11:22

写得非常好，，要是早点知道就可以省很多麻烦了，，有时候写某些效果纠结的不行，，用这个就好多了

回复


8.

浅唱姘華说道：

2016年11月12日 14:12

受益颇多

回复


9.

文艺小青年说道：

2016年11月11日 16:01

鱼哥，你的产量越来越少了！！！！

回复


10.

yanghaha说道：

2016年11月8日 17:24

好产品的意义在于用户和企业受益，好代码的意义在于自己的寿命以及同事关系。



[回复](#)

11. **simon_tal**说道：

2016年11月8日 16:21

换成 IE8 胡歌都不见了 哈哈！

[回复](#)



12. **yp**说道：

2016年11月8日 11:29

偷偷用IE7/8测新的招聘官网一下，你懂的~

[回复](#)

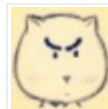


13. **xieguanglei**说道：

2016年11月7日 22:23

拨云见日！

[回复](#)



14. **猫哥**说道：

2016年11月7日 16:09

不单独在 active 上写样式是点睛之笔！

类名压缩这块有些不一样的看法，因为类名可能同时有 CSS、JS和模板在引用，所以使用场景上会有很多限制，配置也会越发复杂。从成本收益上看，遵循一定的命名规范即可，我猜测 gzip 的压缩收益可能会更好。

[回复](#)



15. **cshenger**说道：

2016年11月4日 22:31

然而看了下网站，你们居然不招前端？！差评>_<

[回复](#)



16. **清冰**说道：

2016年11月4日 15:31

有个项目的购物车就用了ckecked，初衷是为了方便通过标识选中的商品调整页面显示的相关数据，还是张大大考虑的深远，讲解的透彻，赞~

[回复](#)



17. **zsusyt**说道：

2016年11月3日 10:08

学习了。

[回复](#)



18. **盖大楼**说道：

2016年11月2日 10:39

学习了，看完了总结了下。自我感觉active就是一个标志，在不同的父级下，active代表的含义不同，可以使改变任何状态。压缩类名是不是就是，在js中用一个var ac='空格active'，ac代替要出现active的地方。

有个地方大神写错字了：三_3.加入说页面中有10个UI交互效果。

我自己模仿写了一个demo：active在不同的父级box及box2下有不同的效果：

box：下拉更多；box2：字体变红；

用ac代替active，但是定义变量时要在“active”前面加个空格。



```
.box{padding: 10px; font-size: 14px; max-height: 90px; overflow: hidden; transition: all .25s;width: 260px;float: left; }
.box2{float: right;}
.box>p,h3{width: 260px;}
.more{background:linear-gradient(to bottom, hsla(0,0%,100%,0), hsla(0,0%,100%,.9) 50%, hsla(0,0%,100%,1)); display:block ; width: 270px; height: 80px;line-height: 80px; text-align: center; position: absolute; top:70px; left: 0px;color: black; text-decoration: none;}
.box.active{max-height: 200px;}
.box2.active{color:red;}
.active .more{display: none; }
```

flv.js-B站HTML5播放器内核开源

项目地址: <https://github.com/Bilibili/flv.js>

一个纯JS的不依赖Flash的HTML5 Flash视频 (FLV)播放器!

本项目受hls.js启发, 需要依赖Media Source Extensions才能生效。

[更多↓](#)

字体变色

```
var box=document.getElementById('box');
var box2=document.getElementById('box2');
var more=document.getElementById('more');
var ac="" active";
if(box&&box2&&more){
more.onclick=function(){
box.className+=ac;
box2.className+=ac;
};
}
```

[回复](#)

19. 会飞的胖子说道:

2016年11月1日 16:04

之前是在.on上面直接进行css, 没有考虑到耦合的问题, 看了文章后才发觉这样的更改问题还是很多, 懒, 比较多用的是class“.on”

[回复](#)



20. 麦尼玛桑说道:

2016年11月1日 10:35

选项卡就用这种方式实现的, 但是没像大神一样想这么深入系统, 膜拜

[回复](#)



21. erlking说道:

2016年11月1日 00:34

类名压缩的概念真是耳目一新, 孤陋寡闻貌似没见过有这类的自动化插件, 期待大湿放出

[回复](#)



22. ziven27说道:

2016年10月31日 18:36

两个字符的如果不加前缀很容易和你的zxx.lib.css冲突。比如.dn, .di。

对于这种不能独立使用的样式名, 我之前习惯加一个s_ (son_) 的前缀。

这样就可以放心的使用.s_on了。

有一个疑问就是, 我可以直接使用下滑线作为css名称的前缀吗?

比如: .box_on{ } .box_on{ }

这样就变成了_on, 这种命名直接给人的感觉就是前面需要一个父级或者兄弟的选择器。就是没有详细测试过下滑线前缀, 不敢放心的用。

[回复](#)



张鑫旭说道：

2016年10月31日 19:47

zxx.lib.css中的CSS也会被压缩的，不会存在冲突

[回复](#)



ziven27说道：

2016年11月2日 10:04

是有什么特别的css类名的压缩机制吗？zxx.lib.css中本身的css名称就已经足够短了啊

[回复](#)



23. dals说道：

2016年10月31日 16:48

路过

[回复](#)



24. XXig说道：

2016年10月31日 15:16

受益匪浅，偷偷的说阅文集团新的招聘官网，缩小窗口轮播图旁边会留白哦

[回复](#)



25. meepo说道：

2016年10月31日 10:36

学习。

旭哥文章满满的干货。

[回复](#)



26. 7P_LonG说道：

2016年10月31日 10:08

on,off,loading,个人还会用上这些，当然只是脑中觉得方便，没想到这么深。受教了。

[回复](#)



27. cc长空说道：

2016年10月30日 14:30

我常用的状态类是`.Act`，首字母大写来跟其他普通类区分。另外`Act`莫名的感觉更帅一点

[回复](#)



28. RDD说道：

2016年10月30日 10:08

大赞，确实是优秀的解决方式！各种程度上都受教了！

[回复](#)



29. 前端-小强说道：

2016年10月30日 09:32

赞，向张老师学习。简单快速完成任务，也许是对产品快速迭代有更多帮助，但是多花点时间在思考上面，有没有更合理的方法，而不是一味的简单完成任务，对于整个项目的维护会更有帮助。



最新文章

- » 常见的CSS图形绘制合集
- » 粉丝群第1期CSS小测点评与答疑
- » 分享三个纯CSS实现26个英文字母的案例
- » 小tips: 纯CSS实现打字动画效果
- » CSS/CSS3 box-decoration-break属性简介
- » CSS :placeholder-shown伪类实现Material Design占位符交互效果
- » 从天猫某活动视频不必要的3次请求说起
- » CSS vector-effect与SVG stroke描边缩放
- » CSS ::backdrop伪元素是干嘛用的?
- » 周知: CSS -webkit-伪元素选择器不再导致整行无效

今日热门

- » 常见的CSS图形绘制合集 (190)
- » 未来必热: SVG Sprite技术介绍 (119)
- » 粉丝群第1期CSS小测点评与答疑 (115)
- » HTML5终极备忘大全 (图片版+文字版) (93)
- » 让所有浏览器支持HTML5 video视频标签 (86)
- » Selectivizr-让IE6~8支持CSS3伪类和属性选择器 (82)
- » CSS3下的147个颜色名称及对应颜色值 (79)
- » 视区相关单位vw, vh..简介以及可实际应用场景 (76)
- » 写给自己看的display: flex布局教程 (76)
- » 小tips: 纯CSS实现打字动画效果 (76)



今年热议

- » 《CSS世界》女主角诚寻靠谱一起奋斗之人 (76)
- » 不借助Echarts等图形框架原生JS快速实现折线图效果 (64)
- » 看, for..in和for..of在那里吵架! (60)
- » 是时候好好安利下LuLu UI框架了! (47)
- » 原来浏览器原生支持JS Base64编码解码 (35)
- » 妙法攻略: 渐变虚框及边框滚动动画的纯CSS实现 (33)
- » 炫酷H5中序列图片视频化播放的高性能实现 (31)
- » CSS scroll-behavior和JS scrollIntoView让页面滚动平滑 (30)
- » windows系统下批量删除OS X系统.DS_Store文件 (26)
- » 写给自己看的display: flex布局教程 (26)

猜你喜欢

- 精简高效的CSS命名准则/方法
- 了解web前端领域的undefined behavior未定义行为
- 翻译-高质量JavaScript代码书写基本要点
- 页面可用性之outline轮廓外框的一些研究
- CSS+JavaScript实现页面不同布局的切换
- HTML5 DOM元素类名相关操作API classList简介
- CSS样式分离之再分离

- CSS的样式合并与模块化
- 自己以前架构CSS/JS时候挖过的一些坑
- 高富帅seajs使用示例及spm合并压缩工具露脸
- Chrome下语音识别实现页面快速引导以及重要交互