

CSS蛋疼应用之：数据上报和HTML验证

这篇文章发布于 2018年12月6日, 星期四, 00:45, 归类于 [CSS相关](#)。阅读 3217 次, 今日 16 次 [2 条评论](#)

by zhangxinxu from <https://www.zhangxinxu.com/wordpress/?p=8258>

本文可全文转载, 个人网站无需授权, 只要保留原作者、出处以及文中链接即可, 任何网站均可摘要聚合, 商用请联系授权。

一、纯CSS实现数据上报

举个例子, 要跟踪并统计某个按钮的点击事件:

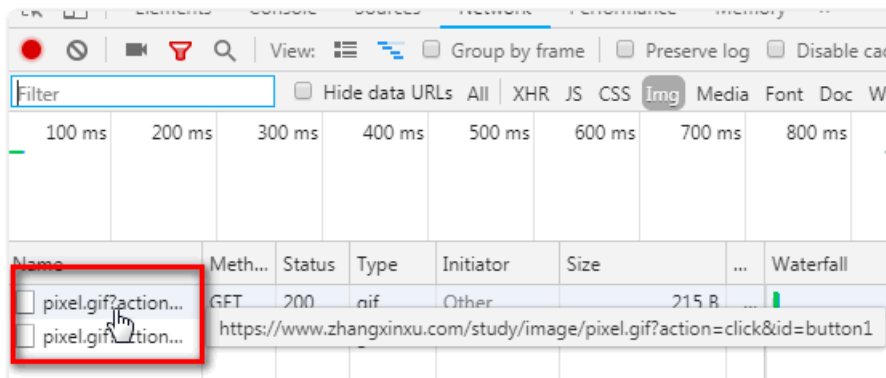
```
.button-1:active::after {
    content: url(./pixel.gif?action=click&id=button1);
    display: none;
}

.button-2:active::after {
    content: url(./pixel.gif?action=click&id=button2);
    display: none;
}
```

此时, 当我们点击按钮的时候, 相关行为数据就会上报给服务器, 这种上报, 就算把JS禁掉也无法阻止。

您可以狠狠地点击这里: [纯CSS跟踪统计上报点击行为demo](#)

点击demo页面的两个按钮, 可以看到发出了如下的请求:



当然, 我们可以统计的不仅仅是单击行为, hover行为, focus行为都可以统计, 当然, 还有很多其他方面的。例如:

1. 不支持CSS3浏览器比例统计

```
.any-element {
    background: url(./pixel.gif?css=2);
    background-image: url(./pixel.gif?css=3), none;
}
```

例如, 我的Chrome发出的图片请求地址就是:

<input type="checkbox"/>	pixel.gif?css=3	GET	200	gif	some-css-t...	(from memory)
	https://www.zhangxinxu.com/study/image/pixel.gif?css=3					

类似的，我们可以检测支持其他一些CSS属性的比例，要比单纯看浏览器的占比要精准的多。因为同样是Chrome浏览器，不同用户版本可能不一样，要想准确知道某些CSS新特性支持情况，这种CSS上报方法要更准确。

可以使用 `@supports` 规则。

```
.any-element {
  background: url(/pixel.gif?grid=0);
}
@supports (display: grid) {
  .any-element {
    background: url(/pixel.gif?grid=1);
  }
}
```

2. retina屏幕占比统计

要么上报0，要么上报1，最后可以知道retina屏幕的比例。

```
.any-element {
  background: url(/pixel.gif?retina=0);
}
@media screen and (-webkit-min-device-pixel-ratio: 2) {
  .any-element {
    background: url(/pixel.gif?retina=1);
  }
}
```

例如，我家里的window普通显示屏：

Name	Meth...	Status	Type	Initiator	Size
<input type="checkbox"/> pixel.gif?retina=...	GET	200	gif	some-css-t...	(from memory c
<input type="checkbox"/> pixel.gif?	GET	200	gif	some-css-t...	(from memory c
	https://www.zhangxinxu.com/study/image/pixel.gif?retina=0				

类似的，可以检测宽屏设备比例等。

3. 是否支持某字体

例如，用户是否电脑是否安装了思源黑体：

```
@font-face {
  font-family: anyFontName;
  src: url(/image/pixel.gif?font=unmatch&id=s_h_s);
}
.element-with-text {
  font-family: 'Source Han Sans CN', 'anyFontName';
}
```

这个要看font请求，如果你的浏览器没有安装思源黑体，则会尝试加载anyFontName这个字体，于是发起

了请求，如下图所示：

如果安装，则没有上报。

//zxx: 对于一些常见中文字体在CSS中的英文名称，可以参见我之前的研究成果：“[CSS font-family常见中文字体对应的英文名称](#)”。

以上1,2,3三个CSS上报数据的小按钮，您可以狠狠地点击这里亲自体验一番：[一些CSS跟踪上报示意demo](#)

页面截图如下：

我们还可以借助浏览器原生行为简化我们上报成本，比方说表单验证出错，用来统计用户注册或者其他重要表单操作的成功率。

CSS如下：

```
.track:invalid {
    background: url(/pixel.gif?action=regist&status=invalid);
}
.track:valid {
    background: url(/pixel.gif?action=regist&status=valid);
}
```

每当表单提交的时候，我们给form元素添加类名 `.track`，此时，其会自动上报表单是否填写完全成功，`:invalid` 和 `:valid` 都是标准的原生的CSS伪类选择器，我们无需自己写验证逻辑。

JS如下：

```
forms.addEventListener('submit', function (event) {
    event.preventDefault();
    // 上报成功与否
    this.classList.add('track');
    // 这个是不影响原生表单的提交行为，实际开发多走Ajax
    if (this.reportValidity()) {
        this.submit();
    }
    setTimeout(function () {
        this.classList.remove('track');
    }.bind(this), 0);
});
```

HTML这里也需要novalidate属性配合下：

```
<form novalidate></form>
```

眼见为实，您可以狠狠地点击这里：[表单验证出错与数据上报demo](#)

不过，这种方法有个缺陷，无论成功与失败，只能上报一次。

表单的 `reportValidity()` 方法虽然很省力，但是会呼起浏览器原生的提示，过不了设计师那一关的，因此，恩，大家high一下即可。

二、纯CSS实现HTML验证与提示

再说另外一个纯CSS应该，那就是进行HTML验证，并且直接在页面上输出来。

举个简单例子，图片如果空 `alt` 属性，则高亮这个图片：

```
img[alt=""] { outline: 2px solid red; }
```

又或者 `_blank` 链接的 `rel` 属性没有加 `noopener` 和 `noreferrer`。

```
a[target="_blank"]:not([rel="noopener noreferrer"]) { outline: 2px solid red; }
```

诸如此类。

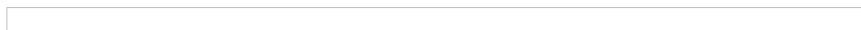
然后配合伪元素在页面上显示提示的文字：

```
script[src]:not([async]) {  
  display: block;  
}  
script[src]:not([async])::after {  
  content: '试试给带有[src="..."]的<script>元素添加[async]属性';  
}
```

然后，就有清闲的有志之士还专门搞了个通过CSS验证检测HTML合法性的CSS库，名为construct.css，项目地址为：<https://github.com/t7/construct.css>

这里有个demo，有兴趣的可以进去瞅瞅。

使用很简单，你的项目引入这个CSS，那些使用有问题的HTML就会标记并提示，类似下面截图：



然而，有些提示嘛，咳咳，有些牵强附会了，大家当新闻八卦看看就好了。

三、为什么说蛋疼应用

本文提到的两个CSS的新奇应用，怎么说呢，有点像手工耿哥的各种手工作品，很有新意，很有特色，可惜，并不实用。



比方说一开始的数据上报，局限很大，很多行为你无法统计；另外维护非常不方便，把一大堆URL地址塞到CSS中，回头要是改个什么东西，很啰嗦的。CSS是偏前前端的玩具，而数据上报偏后的前端更合适，实际开发肯定不会这么玩的，人力资源利用完全没有最大化嘛。

还有那个HTML验证，这年头，还有谁关心HTML语义，结构啥的嘛，小程序，Vue这种HTML结构都是自定义的，和标准越走越远了，没有市场，就没有价值。而且里面很多验证，有些想得太多，空div为什么就不能存在呢？我就想不通了。

因此看来，本文介绍的这两个CSS应用，只适合远观，不适合亵玩。

以上！

《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](#)。



« [HandBrake](#)乃Web MP4视频免费压缩工具不二之选

[CSS font-feature-settings 50+关键字属性值完整介绍](#) »

猜你喜欢

- 博闻强识: 了解CSS中的@ AT规则
- 玩转HTML5 Video视频WebVTT字幕使用样式与制作
- 隐私相关-了解HTML5 Do Not Track API
- CSS3 box-shadow兼容loading效果兼IE10+ CSS Hack介绍
- 小tip: CSS动态实现文本框清除按钮的隐藏与显示
- CSS font-family常见中文字体对应的英文名称
- 真正了解CSS3背景下的@font face规则
- 设备像素比devicePixelRatio简单介绍
- 近期手机网页项目一些杂碎心得分享
- 视网膜New iPad与普通分辨率iPad页面的兼容处理
- CSS reset的重新审视 - 避免样式重置

分享到: [+](#) [QQ](#) [微信](#) [微博](#) [贴吧](#) [收藏](#) [0](#)

标签: :invalid, :valid, @supports, css相关, devicePixelRatio, doNotTrack, novalidate, reportValidity, track, 伪元素, 思源黑体

发表评论 (目前2条评论)

名称 (必须)

邮件地址(不会被公开) (必须)

网站

提交评论

1. **refined-x.com**说道:
2018年12月15日 13:17

自黑典范^^

[回复](#)



2. **cshenger**说道:
2018年12月8日 03:17

打破零惨案，知道一点都是好的，万一哪天可以说出来装个b呢

[回复](#)



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](#) ⁽¹⁷⁸⁾
- » [粉丝群第1期CSS小测点评与答疑](#) ⁽¹¹²⁾
- » [未来必热: SVG Sprite技术介绍](#) ⁽¹¹¹⁾
- » [HTML5终极备忘大全 \(图片版+文字版\)](#) ⁽⁸⁵⁾
- » [让所有浏览器支持HTML5 video视频标签](#) ⁽⁸³⁾
- » [Selectivizr-让IE6~8支持CSS3伪类和属性选择器](#) ⁽⁸⁰⁾
- » [CSS3下的147个颜色名称及对应颜色值](#) ⁽⁷⁸⁾
- » [小tips: 纯CSS实现打字动画效果](#) ⁽⁷²⁾
- » [写给自己看的display: flex布局教程](#) ⁽⁶⁹⁾
- » [分享三个纯CSS实现26个英文字母的案例](#) ⁽⁶⁹⁾

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 scrollToView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

- [博闻强识: 了解CSS中的@ AT规则](#)
- [玩转HTML5 Video视频WebVTT字幕使用样式与制作](#)
- [隐私相关-了解HTML5 Do Not Track API](#)
- [CSS3 box-shadow兼容loading效果兼IE10+ CSS Hack介绍](#)
- [小tip: CSS动态实现文本框清除按钮的隐藏与显示](#)
- [CSS font-family常见中文字体对应的英文名称](#)
- [真正了解CSS3背景下的@font face规则](#)
- [设备像素比devicePixelRatio简单介绍](#)
- [近期手机网页项目一些杂碎心得分享](#)

- 视网膜New iPad与普通分辨率iPad页面的兼容处理
- CSS reset的重新审视 - 避免样式重置