

基于原生HTML的UI组件开发

这篇文章发布于 2016年01月22日, 星期五, 17:35, 归类于 [Web综合](#)。阅读 50744 次, 今日 3 次 [32 条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com>

本文地址: <http://www.zhangxinxu.com/wordpress/?p=5180>

本文属于波澜不惊, 平铺直叙版本。如果想看有有激情有思想的版本, 可以点击这里: “[顺势而为, HTML发展与UI组件设计进化](#)”。


一、基于HTML开发组件的设计思想

要想知道基于原生的HTML进行UI组件开发是什么鬼, 您可以狠狠地点击这里: [基于原生的HTML UI组件体验demo](#)

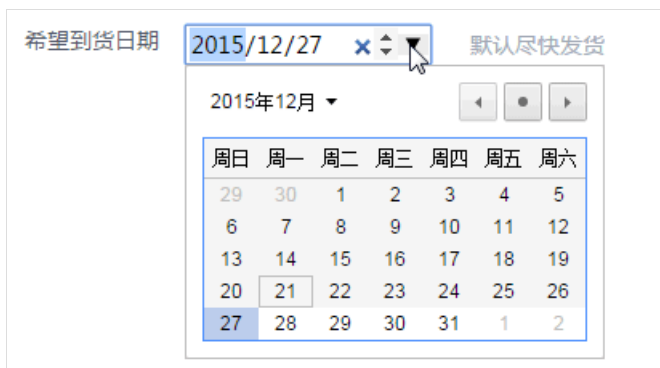
点击上面的demo, 进入一个平凡的静态页面, 引入眼帘的是一个普通的表单, 里面的UI都是系统默认的, HTML功能也是原生的。

例如:

- title提示



- 选择日期



- 点击提交的表单验证

完成您的购物订单 ⓘ

邮箱*

手机号* ! 请填写此字段。

您的款式* ☐ 男款 ☐ 女款

购买数量*

所在城市*

给商家留言

☐ 使用运费险0.4元

您总共需要支付 0 元

是否货到付款 ☐

希望到货日期 默认尽快发货

UI虽然原始，但是功能却是很健全的。

例如：

- 男女款式、城市以及运费险对价格的影响

您的款式* ☒ 男款 ☐ 女款

男款每件488元

购买数量*

所在城市*

运费：5 元

给商家留言

☒ 使用运费险0.4元

您总共需要支付 493.4 元

是否货到付款 ☐

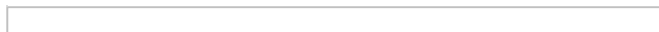
- 表单提交事件



下面，见证奇迹的时刻到了，点击demo页面（下图所示）的按钮进行某项目UI组件资源的加载和初始化：



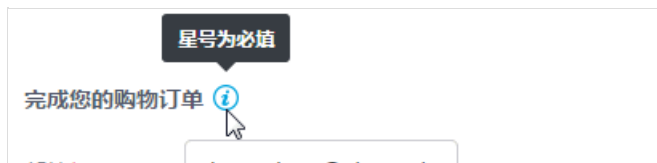
结果，一瞬间，上面原始粗糙的界面一下子变成了这样子：



妥妥的丑小鸭变成了白天鹅，包括之前原生的HTML功能。

例如：

- title提示



- 选择日期



- 点击提交的表单验证



而，最最重要，和最最神奇的事情是：我们仅仅是引入了UI组件的一些CSS和JS，对，仅仅是引入和一点初始化，没有动之前一点点一丝丝的业务JS.但是，之前的各种交互功能，却完全不受影响，反而体验更上两层楼！

请看下面的gif截图演示：

之所以这里的UI组件可以和业务相关JavaScript完全分离，同时可以无缝对接。就是因为设计理念上是基于原生的HTML实现的，让UI组件回归了其本质或者说本职工作——UI。

二、基于原生HTML的UI组件开发

解决2个疑问：

1. 为何可以基于原生HTML进行UI组件开发？
2. 如何基于原生HTML进行UI组件开发？

1. 为何可以基于原生HTML进行UI组件开发？

HTML中自带的很多原生的UI表现和UI组件的UI表现本质上是类似的，差别仅仅在于长相的粗糙和精致。

举个例子，title提示。浏览器默认使用 `title` 属性，长这样：



而设计师设计的tips组件是小黑风格，如下：



这两者其实是一个东西，作用是一样的，差别仅仅在于——UI。因此，我们完全可以基于原生的title属性实现我们的tips提示效果。利用相同的本质，改变的仅仅是呈现的样子。

再举个更有意思的例子，表单验证。在HTML5中，表单默认内置验证，基于 `type`，`required`，`pattern`，`max`，`min` 等HTML原生属性，且自带UI(下图是win下Chrome效果，其他系统其他浏览器长相不一样)：



下图则是team决定采用的UI形式：



同样的，验证的本质都是类似的，交互的形式也是类似的，不一样的仅仅是UI。因此，我们完全可以基于原生的HTML验证规则实现我们的表单验证效果。利用相同的本质，改变的仅仅是呈现的样子。

其他很多UI组件都是类似的。

当我们希望改变的仅仅是相貌的时候，重复利用之前的灵魂难道不是最合理的策略吗？

2. 如何基于原生HTML进行UI组件开发？

有2个要点：① API参数直接取自HTML；② 回调直接trigger原生事件；

我们拿上面提过的日期时间选择器举例说明下。

日期选择器，主流实现基本上是这样的：

```
<input id="date" type="text">
```

```
new DatePicker($("#date"), {  
  type: "date",  
  initDate: ..,  
  beginDate: ..,  
  endDate: ..,  
  onSelected: $.noop  
});
```

API参数，事件的回调全部源自JS参数。

而面向HTML的设计思想下的实现则是：

```
<input type="date" min="2015-11-11" max="2015-12-31">
```

```
new DateTime($("#[type=date]");
```

API参数全部取自HTML，JS代码仅仅就是全局初始化（一次覆盖所有时间类控件）。其中HTML中的 `type` 属性对应JS中的 `type` API，`value` 属性值对应 `initDate` 值，`min` / `max` 分别对应 `beginDate` / `endDate`。至于 `onSelected` 回调，则是通过trigger `input` 框原生的 `change` 时间实现。

于是，其他前端小伙伴在开发的时候，就可以按照原生的HTML属性和事件来开发就可以了，从而实现业务JS和UI组件基于0成本的无缝对接。

三、面向HTML的UI组件开发的好处

1. 语义化，可访问性，SEO等；
2. 学习成本低；
3. 专注HTML控件本身，而不是组件；
4. 可以一次性全局处理；

1. 语义化，可访问性


毕竟是基于原生HTML来开发的，这一块必定杠杠的。

2. 更低的学习成本

不需要记住千差万别的JS API名称，记住标准的HTML5属性即可，反过来对一些前端开发同学的HTML学习还起到了帮助作用。

而学习成本低对于跨团队合作非常有帮助。其他团队同学乐于使用你的东西，介入快，实现效果好，大家都开心。反之，API千差万别，每次使用都要去翻文档，肯定影响合作。

不过，实践下来，有一点学习成本我没考虑到，就是转换思维方式的学习成本。实际上只要面向元素的

HTML元素开发就可以了，但是有遇到小伙伴，还是按照老的思维方式，在生成的UI组件元素上做文章。

3. 专注HTML控件本身，而不是组件

举个例子，日期选择器，当日期修改了，我们要干嘛干嘛，直接：

```
$("input").change(function() {});
```

想要修改日期范围，直接：

```
$("input").attr({
  "min": "2015-12-27",
  "max": "2016-12-27"
});
```

UI组件会自动同步。没有任何组件相关的JS代码，也没有什么故弄玄虚，没有所谓的高屋建瓴，全是很简单基础的HTML操作。是不是这样的开发反而很省心，连小白用户也能上手？

于是乎，在多团队联合协作开发的时候，前端开发的进度并不会受UI组件开发影响，面向HTML，专心自身业务开发就可以了。

于是乎，实现了一个听上去很了不得的东西：前端分离。

不仅如此，厂子里有很多开发，负责内部项目，会写JS擅长业务功能实现，但是，UI这块是个软肋。OK，此时，我们这里面向HTML开发的UI组件体系就是其救星，对吧，直接引入CSS和JS，简单全局初始化一下（可能还有一些简单的微调），结果，页面立马高大上了，是不是很有用！

4. 可以一次性全局处理

传统实现，每个具体业务的脚本里面要参与UI组件的具体API参数设置。而面向HTML的实现，API落地与具体的业务页面，于是乎，只要在项目的common.js中全局初始化一下，如下拉 `Select.init()`，具体的业务JS文件（绝大多数情况下）中就无需再出现UI组件相关的JS代码。

UI层的JS代码和业务层JS代码分离，实现进一步的「前端分离」，去耦合。对于日后的维护、升级等必定大有裨益。

四、总结

本文主讲设计思想，至于具体的技术细节，以后有机会会慢慢分享。越是简单的成品越是需要足够的积累。

然而，现在的我再重新评估UI组件的实现，还是有一些遗憾的，主要遗憾在于，HTML层→数据层→展现层这三层概念实现的时候并没有理得很清楚。目前，HTML层和展现层没有任何问题，但是，数据层，并没有完整贯穿整个UI组件体系，导致，本UI组件体系不能很好地吸引对JSON数据有着偏执爱好的开发，以及应付潜在的极端需求。

不过，不要担心，明年，也就是16年，我会对组件设计进一步增强，首先，不考虑IE7浏览器，于是可以做的事情就更多了；其次，清晰的数据层作为中间层的代码重构等。

以上，希望本文的内容能够对大家有一点启示。



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



« 前端早读课专访

CSS font关键字属性值的简单研究 »

猜你喜欢

- 面向设计的半封装web组件开发(概要版)
- 以20像素为基准的CSS网页布局实践分享
- 基于CSS color属性的静态UI组件重构策略
- AMCSS（CSS属性模式）开发简介
- CSS3模拟window7炫酷界面效果展示
- 使用CSS实现Photoshop选区效果及应用
- 网页制作辅助工具-jQuery标尺参考线插件
- 是时候好好安利下LuLu UI框架了！
- HTML CSS列表元素ul,ol,dl的研究与应用
- HTML5新增的form属性简介
- HTML accesskey属性与web自定义键盘快捷访问

分享到:         0

标签: [html](#), [HTML5](#), [UI](#), [组件](#)

发表评论（目前32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. molaga.z说道:

2017年05月27日 11:25

关于组件化，除了原生html，还有shadow dom不知道可行性高不高

关注博主挺久，前端的东西收获不少，感谢

PS:一只做画面比后端还多的java小码农

[回复](#)



2.

看看说道:

2017年02月27日 16:51

空军建军节

回复


3.

wenrun说道:

2017年02月14日 15:50

checkbox在ie7下面使用[defaultChecked]进行处理时候，input的状态改变了，label的显示效果却没有变化，是要怎么进行重绘还是要如何处理?

回复

张鑫旭说道:

2017年02月14日 17:02

checked属性值也要同步修改，remove或者add~

回复


4.

luckyqiao说道:

2017年01月22日 14:28

样式乱了，求修复

回复


5.

indra说道:

2016年11月25日 18:21

样式乱了。。。求修复

回复


6.

sddfsf说道:

2016年10月12日 15:09

。。。开源了没?

回复


7.

xyzxyz说道:

2016年09月13日 08:53

大神，有个问题请教，原生datepicker快速点击下一月按钮，那个动画怎么实现的呢，或者按住不放的动画效果，是用canvas吗还是svg还是requestAnimationFrame

回复


8.

xxxx说道:

2016年09月12日 15:25

示例有问题啊，好像是样式不对

回复


9.

阿文说道:

2016年06月1日 16:26

啥时开源呢 我自己网站后台想用个



[回复](#)

10. **ManfredHu**说道:

2016年04月7日 14:36

类似polyfill的UI fix.我没理解错吧。
应该是一种趋势，统一浏览器原生丑丑的UI

[回复](#)



11. **qqxufo**说道:

2016年04月5日 14:38

看起来不错呀，可以偷学代码么==

[回复](#)



12. **steven**说道:

2016年03月21日 13:26

请问目前开源了么，还有就是input date 在微信端会先弹出原生的，其次又弹出第三方的组件选择器。

[回复](#)



13. **newbee**说道:

2016年03月18日 11:47

这个东西和pintucss是不是差不多呢，不过事件都是用的原生的。如果用这个的话，虽然不用背GregorianCalendar这种，但是是不是还要大量的背他的class，比如：ui_button,ui_button_blue，这种？？？
求解答~~~~~

[回复](#)



14. **刘东奇**说道:

2016年03月17日 15:15

基于原生html开发页面，这个思路真是太好了。不知道现在有没有博主说的这种只需要少量引用和初始化就能让原生html美化的产品可用

[回复](#)



15. **Daisy**说道:

2016年02月22日 16:17

如果能推动HTML原生控件往全部支持个性化定制的方向发展，自然是极好的！

[回复](#)



16. **赵某人**说道:

2016年02月18日 09:00

以后腾讯浏览器弄个网页渲染按钮得了.太黑科技了.

[回复](#)



17. **小李**说道:


2016年02月9日 16:11

颇受启发

[回复](#)



18. hhgnsc说道:
2016年02月6日 15:46
手机号没盖严实呀！好像还是能看清
[回复](#)





19. lianpen说道:
2016年02月4日 11:30
哥哥，脚本干毛加个密呀 如何阅读呀啊啊啊
[回复](#)

张鑫旭说道:
2016年02月4日 15:29
@lianpen 因为尚未开源，只能调用项目线上地址，是压缩版本。
[回复](#)



20. sandman说道:
2016年02月3日 11:54
demo页面的代码可以下载用吗?
[回复](#)



张鑫旭说道:
2016年02月3日 16:36
@sandman 目前尚未开源，非商业用途是可以的。
[回复](#)



21. sandman说道:
2016年02月3日 11:39
这个思路很赞！学习大牛！
[回复](#)





22. 陈小轩说道:
2016年01月28日 18:06
赞赞赞，以前写代码会先把业务逻辑写好，在UI，这个思路更明显，彻底，赞赞赞
[回复](#)



23. 北京soho说道:
2016年01月27日 16:59
您引入的ui里面的js和css是你自己写的吗?
[回复](#)

张鑫旭说道:
2016年01月27日 17:30
@北京soho 恩，欢迎交流~
[回复](#)



24.

reyhappen说道：

2016年01月26日 15:43

鑫哥，搞个表格操作的插件玩玩哪(实现单元格合并，拆分，在合并的基础上的插入新行、新列)

回复


25.

liuqingchi说道：

2016年01月26日 12:47

dojo 早就做到了，是不是多此一举呢？

回复


26.

dr说道：

2016年01月24日 11:23

学习了

回复


27.

松鼠闹IT说道：

2016年01月23日 17:21

越是简单的成品越是需要足够的积累。

说的相当不错。我一直都在寻找如何能简化美化UI的方式

回复


28.

懒人说道：

2016年01月22日 22:17

适合jq，不适用mvvm

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](#) ⁽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

- 面向设计的半封装web组件开发(概要版)
- 以20像素为基准的CSS网页布局实践分享
- 基于CSS color属性的静态UI组件重构策略
- AMCSS（CSS属性模式）开发简介
- CSS3模拟window7炫酷界面效果展示
- 使用CSS实现Photoshop选区效果及应用
- 网页制作辅助工具-jQuery标尺参考线插件
- 是时候好好安利下LuLu UI框架了！
- HTML CSS列表元素ul,ol,dl的研究与应用
- HTML5新增的form属性简介
- HTML accesskey属性与web自定义键盘快捷访问