

## HTML5 number类型文本框step属性的验证机制

这篇文章发布于 2012年12月7日, 星期五, 18:00, 归类于 [HTML相关](#)。 阅读 47274 次, 今日 5 次 [一条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com>  
本文地址: <http://www.zhangxinxu.com/wordpress/?p=2817>

我在下一盘很大的棋, 本文只是其中的一个棋子。

需要提前知道的:

1. 目前而言, 对 `step` 雄起的浏览器为IE10+, Chrome以及Opera浏览器。
2. 需要预先知道 `number` 类型 `input` 的一些基本知识。
3. HTML5 `step` 的验证机制可以应用在不支持的浏览器上。

### 问题引导, 威逼利诱

如下一小段含 `step` 属性的HTML代码:

```
<input type="number" step="2.1" min="1" />
```

在Chrome浏览器下长相应该是下面这个样子 (假设限宽70像素):



下面提问, 请问:

对于上面的 `number` 框, 下面几个数值哪些是有效的?

- A. 1.0
- B. 2.1
- C. 3.1
- D. 4.2

我的答案是:































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































在你大脑这样：



也有可能是这样子：



之后，我们得到了正确的答案：A. 1.0以及C. 3.1

可能有人有疑问？明明 `step` 是 `2.1`，但是 `2.1` 确实无效的数值呢？

keep on~~

## 实例页面，现身说法

如果您正在使用的是文章一开始提到的“雄起”浏览器，您可以狠狠地点击这里：[HTML5 number文本框step验证demo](#)

下为Chrome浏览器下载图：

N个 `number` 类型输入框，您可以随意设置值，点击最后的“提交”按钮，内置的HTML验证会判断您输入的数值是否是符合要求的。

例如，与上面的提问对应的输入框是倒数第二个框框，如下示意：

我们依次输入 `1.0`，`2.1`，`3.1`，`4.2` 看一下是否会出现提示，结果依次如下截图：

但是，同样是 `step=2.1` 上面一个框框，输入 `3.1` 却是无效的：

为何有此差异呢？

细心的你可能发现了其中的差异，就是 `min` 属性作的祟。

当 `min=1` 的时候，

```
min + step * 0 = 1 + 2.1 * 0 = 1
min + step * 1 = 1 + 2.1 * 1 = 3.1
min + step * 2 = 1 + 2.1 * 2 = 5.2
.
.
.
```

于是，`1.0` 以及 `3.1` 是有效值，而 `2.1` 无效的。

当 `number` 框没有 `min` 的时候，`min` 是无限小，实际验证的时候，`min` 等同于 `0`，因此：

```
<input type="number" step="2.1" />
```

的有效值是：`0`，`2.1`，`4.2`。

```
min + step * 0 = 0 + 2.1 * 0 = 0
min + step * 1 = 0 + 2.1 * 1 = 2.1
min + step * 2 = 0 + 2.1 * 2 = 4.2
.
.
.
```

`max` 跟 `step` 的有效值判断不来电的。

## 总结概括，小功告成

末日即将来临，没有多少时间扯废话，以下为demo页面测试得出的一些结论：

1. 输入非数值内容，如字母 `x`，值会被置空缺省处理 - 浏览器当作你没输入过。
2. 在 `number` 输入框的世界里，`1.00`和`1.0`和`1`是没有区别的。
3. 默认 `step` 为 `1`，因此，木有 `step` 属性值的时候，小数值是无效的（`.0` 除外）。

4. 输入数值的有效与否与 `step` 以及 `min` 属性相关，至于 `max`，是卖切糕的，不搭噶。
5. 输入值是否有效的计算公式是：

```
是否: (输入数值 - min值) ÷ step值 === 整数
```

6. 当没有 `min` 的时候，我们可以把 `min` 等同于 `0` 计算处理（实际 `min` 是无限小）。

7. IE10虽然支持 `number` 类型输入框，但是在UI表现上，却没有上下增减按钮。而且，其提示效果真是粗糙的令人喷舌！



8. Opera浏览器不能准确识别 `step=".1"`，或者应该是不认识，IE10以及Chrome浏览器可以准确识别为 `step="0.1"`，因此，从兼容角度来讲，还是写成 `0.1` 而不是 `.1`

9. 每个浏览器对值无效时候的提示文字都是不一样的：

IE10是：“你必须输入有效值”；

Opera是：此表单不允许数字 \*\*。此输入框值允许特定的数字。

Chrome是：值无效。

个人而言，更喜欢Chrome的简洁明了的表述。

欢迎补充，欢迎指出文章表述不准确的地方。

《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](#)。



» [Page Visibility\(页面可见性\) API介绍、微拓展](#)

» [HTML5 Boolean类型属性\(如required\)值的JS获取](#)

#### 猜你喜欢

- 小tip: 我是如何初体验uglifyjs压缩JS的
- 残念: IE10 IE7-IE9 type=email的完全抛弃
- jQuery html5Validate基于HTML5表单验证插件
- 翻译: 关于表单每个设计师都必须知道的10件事
- 伪元素表单控件默认样式重置与自定义大全
- 小tip: CSS动态实现文本框清除按钮的隐藏与显示
- 表单序列化、规则分离下的复杂表单计算
- HTML可访问性fieldset, legend元素及CSS布局应用
- HTML5按钮元素新属性formation, formenctype等简介
- CSS :focus-within伪类选择器及纯CSS下拉等应用举例
- HTML5 <template>标签元素简介

分享到: [微信](#) [QQ](#) [微博](#) [豆瓣](#) [知乎](#) [人人网](#) [开心网](#) [豆瓣](#) [知乎](#) [人人网](#) [开心网](#)

标签: [HTML5](#), [min](#), [step](#), [type=number](#), [表单](#), [验证](#)

#### 发表评论 (目前一条评论)

名称 (必须)

邮件地址(不会被公开) (必须)

网站

提交评论

1. 尹涛说道:  
2018年07月30日 17:54  
学习了, 6年后来抢个沙发  
[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 (103)
- » 未来必热：SVG Sprite技术介绍 (130)
- » 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 (80)
- » 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》女主角诚挚靠谱一起奋斗之人 (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

- 小tip：我是如何初体验uglifyjs压缩JS的
- 我忍：IE10 & IE7-IE9 type="email"的完全抛弃
- jQuery html5Validate基于HTML5表单验证插件
- 翻译：关于表单每个设计师都必须知道的10件事
- 伪元素表单控件默认样式重置与自定义大全
- 小tip: CSS动态实现文本框清除按钮的隐藏与显示
- 表单序列化、规则分离下的复杂表单计算
- HTML可访问性fieldset,legend元素及CSS布局应用
- HTML5按钮元素新属性:formaction,formenctype等简介
- CSS :focus-within伪类选择器及纯CSS下拉等应用举例
- HTML5 <template>标签元素简介