

# 以20像素为基准的CSS网页布局实践分享

这篇文章发布于 2016年03月22日, 星期二, 22:59, 归类于 CSS相关。阅读 36150 次, 今日 8 次 32 条评论

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

本文地址: <http://www.zhangxinxu.com/wordpress/?p=5264>

## 一、一切从line-height行高说起

想想看, 你CSS构建页面的时候, 默认的字大小和行高分别是多少?

下面是我统计的一些数据:

- 新浪微博: 12px/1.5
- 腾讯微博: 12px/1.75
- 淘宝网/天猫/腾讯微云: 12px/1.5
- 京东: 12px/150%

计算下来, 基本行高要么是18像素, 要么是21像素~

我想, 大多数小伙伴应该没关注过这方面的细节, 基本行高大致就可以, 取 1.5 方便计算, 1.75 呢似乎也无伤大雅。

当下的网页早已告别当初就只能浏览信息的那种状态, 页面结构更加复杂, 大段描述文字的情况只占少数, 因此, 行高的角色有点从阅读体验层面转换为更方便的计算或者其他什么角色。

这里的“其他什么角色”看似平淡无奇的一句话, 其实才是本文的主角。在本文, 行高担任了网页垂直格栅基准的角色!



我们以前可能听过网页格栅布局, 老实讲, 我对水平格栅一点兴趣也没有, 几百篇CSS文章我也从来没有介绍过, 因为跟自己的布局理念不符; 但是, 我今天倒是要提一提垂直格栅!

页面内容往往是自上而下瀑布式的呈现, 内容千变万化, 元素的高度也是变幻莫测, 因此对于垂直方向, 所谓格栅, 几乎是无稽之谈。确实是这样, 但是, 局部的格栅有时候会让我们的页面变得更加规范, 以及可以让我们的工作更加轻松。

而这一切, 就要从行高说起。我们以前写页面, 都是设置字体大小以及行高值, 确定单行文本所占据的高度内容; 而我们这里, 则逆向思维, 我们希望页面基本文字所占据的高度是 20 像素, 则我们的行高应该是?

现在是大屏时代, 假设我们的默认字号大小是 14 像素, 我们计算下:  $20/14 \approx 1.4285714285714286$ , 四舍五入的结果, 于是得到:

```
body {  
  line-height: 1.42857;  
  font-size: 14px;  
}
```

不好意思，给大家下套子了，注意了，在CSS中，行高计算的时候，一定不要向下四舍五入，而要向上。类似上面的代码，虽然 `14*1.42857` 近乎就是20像素，但是，不好意思，最后还是以19像素的高度呈现，在Chrome浏览器下就是如此！

因此，实际的设置应该是：

```
body {  
  line-height: 1.42858;  
  font-size: 14px;  
}
```

于是，我们就得到了一个20像素为基准的网页布局环境了，此时，普通一行文字的高度就是20像素，那又有什么好处呢？

单一起来看，20像素的高度单元和21像素似乎没什么差别，但是，如果放在一个完整的体系里面，价值就能很好地体现了！

## 二、20像素为基准的20\*20像素的小图标策略

基本上每一个网站都离不开小图标，国际通用的图形语言，对于一个网站而言，无论是体验还是辨识度都是必不可缺的。

目前而言，绝大多数网站还是处于12像素字体时代，设计师设计的图标都是按照16\*16像素规格设计的偏多；不太专业的设计师可能会14~20像素之间摇摆。

反正不管怎样，最后（加上sprite工具盛行）我们图形在网页中的尺寸基本上就是 `16px*16px`：

```
.icon-hi {  
  display: inline-block;  
  width: 16px; height: 16px;  
}
```

当然，`17px*18px` 也是很常见的：

```
.icon-hi {  
  display: inline-block;  
  width: 17px; height: 18px;  
}
```

这种一小图标真实尺寸构建CSS代码的方式有3个比较大的问题：

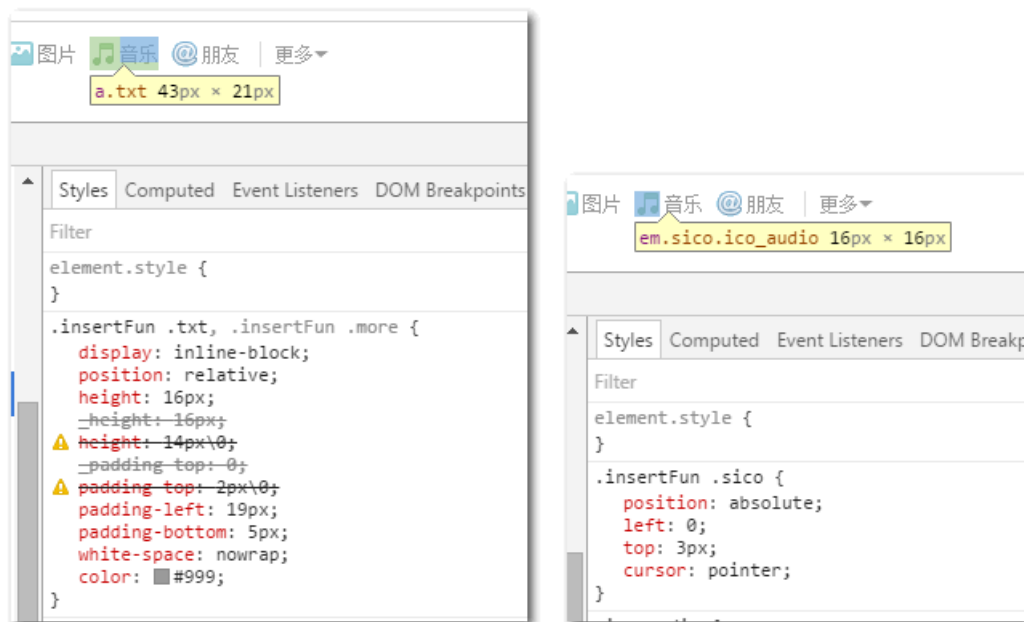
1. 与后面文字的对齐
2. 点击区域大小
3. 重复冗余的CSS代码

### 1. 与后面文字的对齐

由于 `vertical-align` 属性的兼容性，以及 `vertical-align:middle` 并不是严格意义的垂直居中，因此，小图标+文字的对齐，基本上都要针对不同浏览器加个hack补丁；在加上，如果你的图标尺寸一会

儿16像素，一会儿18像素，显然，没法通过全局一个设置使得整站的小图标和文字都对齐良好！

例如，腾讯微博这里，图标就是16像素尺寸，然后，一些五花八门的处理：



很多补丁，很多CSS处理，里面图标使用了absolute绝对定位，这倒是处理兼容挺不错的方法，但是，显然不具有普遍适用性。

## 2. 点击区域大小

有时候，我们的小图标直接就是点击按钮，此时，如果你的尺寸就是16像素\*16像素，会不会点不准的概率就上升了，如果图标是20px\*20px呢？

## 3. 重复冗余的CSS代码

当下类似[grunt-spritemith](#)的小图标合并工具基本成了前端团队的标配，而根据我的观察，基本上，大家都是设计师给的图标直接扔到文件夹里面进行合并，于是，产出的代码基本上就是 `width / height / background-position` 的模式，然而，可能里面70%宽高都是16像素，20%是18像素，还有10%是其他小尺寸，也就是，其实很多CSS代码是可以合并的，然而，都浪费了。

如下生成less代码截图（源自真实项目）：



以上这些问题实际上一个对策就可以搞定，就是所有图标统一按照20px\*20px的标准处理！



你想啊，我们网页文字基础高度是20像素，图标也是20像素高，天然对齐，问题1解决；20\*20的点击区域对吧，显然比16\*16要大，问题2解决；所有图标都是20\*20的尺寸范围内，所有 `width / height` 都可以合并，大大减少CSS代码，问题3也搞定了！

如下图CSS生成模板示意：



———低调的分隔线———

然而，实际上的处理要比上面说的复杂和深奥的多！

## 图标和文字天然对齐

按照我们直观的认知，两个元素都是20像素高，都在自己的垂直范围内居中，那这两个元素应该是在一个水平线上的。实际上真的是这样吗？是的，但是，注意这里的但是，是有条件限制的！

在“[CSS深入理解vertical-align和line-height的基友关系](#)”一文中，其中就已经提及过：

The baseline of an ‘inline-block’ is the baseline of its last line box in the normal flow, unless it has either no in-flow line boxes or if its ‘overflow’ property has a computed value other than ‘visible’, in which case the baseline is the bottom margin edge.

中文直译就是：

‘inline-block’的基线是正常流中最后一个line box的基线，除非，这个line box里面既没有line boxes或者本身‘overflow’属性的计算值而不是‘visible’，这种情况下基线是margin底边缘。

翻译成白话就是：

如果inline-block水平元素‘overflow’不是‘visible’，或者里面没有内联元素（图片、文字之类），则整个元素的基线就是自身的下边缘；否则，基线就是里面最后一行图文元素的基线（✌️这是我们需要的）。

有点不太理解？没关系，不是本文的重点。你只要知道，我们要想20像素高的图标和20像素高的文字天然对齐显示，需要满足这两个条件：

1. `overflow` 属性值除了 `visible` 都不行；
2. 里面需要有不加修饰的文本内容；

所以，下面两种情况都是淘汰的！

```
.icon {  
  display: inline-block;  
  width: 20px; height: 20px;  
  background: ...  
  overflow: hidden;  💀  
}
```

```
<i class="icon"></i>  💀  
  
.icon {  
  display: inline-block;  
  width: 20px; height: 20px;  
  background: ...  
}
```

第一种情况是 `overflow:hidden` 拖后腿了；第二种情况是 `<i>` 标签里面是空大屁，基线还是元素底边缘而不是里面的文字（如果有）。

因此，要想实现小图标天然对齐，我们不能有 `overflow:hidden` 同时HTML标签内部有文本内容，我靠，好多限制，貌似很烦啊，然而，经过本人的实践，是可以有CSS代码段满足各种场景的对齐效果的，如下：

```
.icon {  
  display: inline-block;  
  width:20px; height:20px;  
  background: ...;  
  white-space:nowrap;  
  letter-spacing: -1em;  
  text-indent: -99em;  
  color: transparent;  
  /* IE7 */
```

```

    *text-indent: 0;
    *zoom: expression( this.runtimeStyle['zoom'] = '1', this.innerHTML = '\3000');
}
.icon:before {
    /* 伪元素插入空格文本 */
    content: '\3000';
}

```

您可以狠狠地点击这里：[小图标文字对齐的终极解决方案demo](#)

结果，无论是空标签HTML，还是内置可访问性提示文字的HTML，都是对齐效果棒棒哒！

```

<i class="icon"></i>

<a href="javascript:" class="icon">删除</a>

```



而且，就算文字的字号大小变化，例如 14px → 16px 图标和文字也是对齐良好的，因为，对齐的本质是图标元素里面的文字和后面的文字对齐，文字和文字对齐，自然是天然对齐的，千古难题就这么有了解决方案，以前的CSS hack啊，什么 vertical-align 控制，还有 margin 负值偏移都是浮云了！

### 图标20\*20尺寸扩展grunt工具

设计师设计的图标都是16px~20px不等，CSS代码都是Grunt工具生成的，我们很难简单控制让所有图标都是20\*20的区域大小。除非，我们对所有的小图标在导出的时候，手动扩展画布到20px\*20px。

亲，什么年代了，又不是搞艺术品，手工劳作年代过去了，直接上工具。

我基于GM搞了个20像素以下小图标自动扩展为20像素大小图片的Grunt工具：<https://github.com/zhangxinxu/canvasExpand>

精力有限，这个小项目还没细整就扔上去了，可以看到，很多模板生成的文字我还没来得及改。

window用户记得要安装ImageMagick.exe，安装时候记得勾上那个全局变量什么东西的。

如果有什么问题，欢迎……不要来打扰我，忙，自己想办法，么么哒~~

### 图标的重心像素级处理

有些图标，虽然设计师给的尺寸是标准的，没有多余像素，但是，可能图标本身的形状并不是对称的，尤其上下，这就会导致图标的重心会有些偏上或者偏下，导致和后面的文字呈现的时候，虽然真实尺寸是对齐的，但是视觉感觉却是不在一条线上。如果要求很高，可以让设计师或UI工程师自己微调下，一般1像素就够了，当然，是调整图片，例如，重心低的，下面多1像素高度的透明区域。

## 三、40像素高度的UI组件体系

前面提到过，20像素的基准行高要在体系中才能大放异彩，而这个体系的另一个非常重要的成员就是页面的基本UI组件们！

所有的按钮均是40像素高度，所有的输入框都是40像素高度，所有的下拉框、时间选择框都是40像素高度；



上图截自“[基于原生HTML的UI组件开发](#)”一文那个展示前端分离的例子：[QQ公众平台UI组件下的前端分离demo](#)

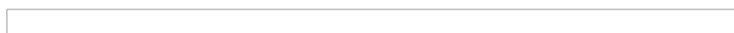
由于我们的基准文字高度是20像素，因此，左边文字距离顶部的间距，就是标准的10像素！



这就使得我们网页中无论是大模块之间的间距，还是小的文字和空间之间的间距；无论是水平的间距还是垂直的间距，全部都是标准的5像素的倍数。从而让我们所有的大小模块的实际高度都是10的倍数(padding-top + line-height\*行数 + padding-bottom)。

换句话说，我们以20像素为基准进行布局和UI组件设计，使得我们的网页间距标准化了，无形之中会让我们页面的排版更专业，同时也让zxx.lib.css的利用率提升了。

如果我们进一步深究按钮或者输入框的实现细节，你会发现，其CSS实现本身就是按照20像素为基准的策略进行实现的：



//zxx: 擦，一看代码，发现按钮直接行高控制的，失策失策，我回去就调整下，这其实是有问题的，适用性大了折扣，因为没有遇到按钮中有图标的设计情境，所以没有暴露出来。更好实现应该和上面输入框一行，20像素行高，使用padding实现最终的40像素高度。

## 四、结束语

看似是简单的要求以20像素为基准，实际上根植于体系中并有一套完整的解决方案。

不过，跳出内容本身，换个角度讲，本文的内容其实挺无聊的。

人家淘宝天猫就是基础高度18像素，不是这里推荐的20像素，但照样年销售额3万亿，照样上市，照样造就了一大批土豪。

所以，大家大可不必过分在意一点点的CSS细节，CSS对于产品的商业价值有，但是到了一定阶段以后，实际上就有限，或者说很难直观的体现，或者说性价比就不高了。今天还有小伙伴问我，`absolute` 绝对定位元素 `display:none` 和 `visibility:hidden` 两个隐藏的渲染性能差异，这个问题的商业价值比本文内容还要小1万倍，以现在浏览器的渲染性能以及我们实际的开发需求，就算有差异，有价值吗，肯定没有！

大环境如此浮躁，你会发现，自己遇到的困境并不是技术成长遇到了瓶颈，而是根本不需要你这方面进一步的技术成长，来，干点收益更明显的事情！

我想，很多前辈技术博客断掉了，怕也是人在职场，身不由己！

我能不能走出一条不一样的路呢？



《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](#)。



### 猜你喜欢

- humans.txt-网站相关人员信息记录的idea
- 面向设计的半封装web组件开发(概要版)
- 基于原生HTML的UI组件开发
- 基于CSS color属性的静态UI组件重构策略
- 翻译 - CSS继承详解
- AMCSS（CSS属性模式）开发简介
- jQuery之replace字符串替换实现不同尺寸图片切换
- 拜拜了,浮动布局-基于display:inline-block的列表布局
- 去除inline-block元素间间距的N种方法
- 基于vw等viewport视区单位配合rem响应式排版和布局
- SVG图标颜色文字般继承与填充

分享到: 1

标签: font-size, line-height, UI, 图标, 标准, 组件, 规范

### 发表评论（目前32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. yuehao说道:

2018年04月19日 10:56

前辈在这条路上披荆斩棘，获得的回报却不成正比，心里膈应，这也是人之常情。  
前辈只管做自己认为是对的事情吧，时间会记录你做的一切。

[回复](#)



2. callcter说道:

2017年02月17日 15:13

学长吊呀，就是因为学长这么努力深入的研究，我们能够知其然并知其所以然，加油！

[回复](#)



3.

洪泽扬说道：

2016年11月25日 08:51

脑洞了一下，20\*20标准的网站会会比没有标准的网站多耗多少流量

回复


4.

纵向发展的小月叶说道：

2016年07月18日 10:13

感谢你的研究，作为水设计师+水前端需要设计标准，我现在在慢慢摸索中，果然腾讯的前辈都摸索出标准了，22333

回复


5.

诺颜说道：

2016年07月17日 17:45

可以转载吗？楼主

回复


6.

林木说道：

2016年05月18日 17:02

那个“QQ公众平台UI组件”又开源吗？感觉很好看的，想研究研究。

回复


7.

z说道：

2016年05月16日 22:54

坚持确实不容易，欣赏你

回复


8.

travis说道：

2016年05月9日 15:28

1.88元

回复


9.

总裁说道：

2016年04月27日 10:51

对于一个html、css、js出身的前端，深有体会，赞！

回复


10.

Tison说道：

2016年04月12日 10:11

淡定，虽然文章末尾说没什么用，但是有的时候在面对设计师苛刻的挑剔的时候，处理不好这个细节真的很造孽呢。理解了baseline这个点再来处理这些问题，瞬间就像诈了尸一样开光啊！

回复


11.

litian说道：

2016年04月7日 18:05

最后一句话，你能，希望你能。现在大家都是结果导向，实际就是一切向钱看，不过希望博主能感性一点，能不一样。

回复





12.

3920001说道：

2016年04月1日 01:12

张含韵那个P的图，我说你到底有多喜欢啊，看了你很多的文章，原先是想静静的学习，不想评论的，看到那个P的图片，不能忍了，真是笑到我了

回复

苏说道：

2016年04月14日 15:39

我也是。。。真是毫无ps痕迹。我很想知道博主老婆一般用什么眼神看张含韵的

回复
13.

MUKI说道：

2016年03月31日 20:04

看到最後那三行 .. 不禁潸然淚下 OTL

回复
14.

sandman说道：

2016年03月29日 17:36

终于碰到了大神解决这个难题！！！！感谢！

回复
15.

whq731说道：

2016年03月29日 17:28

最后的配图  
“我秃了，也变强了”

回复
16.

levomm说道：

2016年03月28日 21:08

我一般是用vertical-align微调几像素来解决这个图标的对齐问题的，是不是太补丁了

回复
17.

levomm说道：

2016年03月28日 21:07

你可以写有商业价值的呀

回复
18.

邪恶的刘大说道：

2016年03月28日 14:18

当然能走出一条不一样的路，那就是情怀啊。不断优化，不仅仅是优化速度……

回复
19.

qinyebo说道：

2016年03月25日 22:57

全方位发展，CSS、javascript、NodeJS、服务端等同步学习，同步学习是做得到的

回复

20.

middle.Lin说道：

2016年03月25日 20:09

感谢

回复


21.

刘东奇说道：

2016年03月25日 14:48

我们跟着你走！

回复


22.

贝伦哈尔特说道：

2016年03月25日 09:52

一如既往的受益匪浅

回复


23.

律讯小前端说道：

2016年03月24日 10:16

谢谢大神批评受教了，我绕进死胡同了

回复


24.

粉丝张说道：

2016年03月23日 17:42

鑫总博客更多久，我就跟多久。每期必看，受益匪浅

回复


25.

末末说道：

2016年03月23日 09:21

看了鑫总博客很多年，学习了很多，确实能坚持下来的不多，特地微信支持了50rmb

回复

张 鑫旭说道：

2016年03月23日 10:16

@末末 感谢！

回复


26.

保健表哥说道：

2016年03月23日 09:03

希望博主可以再更新blog 20年，虽然写作风格我不是很喜欢，但是为了看干货还是忍住看完。

回复


27.

L9m说道：

2016年03月22日 23:52

从这里看见匠人精神了

回复



xiaof说道：

2016年04月10日 17:36

加油！

[回复](#)



xiaof说道：

2016年04月10日 18:26

我想看看博主的多重回复是怎么做的，可以无限嵌套吗，用的是jquery.tmpl吗，谢谢

[回复](#)



28. zhishaofei3说道：

2016年03月22日 23:30

挺好看的

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

#### 今年热议

- » [《CSS世界》女主角诚寻靠谱一起奋斗之人](#) <sup>(76)</sup>
- » [不借助Echarts等图形框架原生JS快速实现折线图效果](#) <sup>(64)</sup>
- » [看，for..in和for..of在那里吵架！](#) <sup>(60)</sup>

- » [是时候好好安利下LuLu UI框架了！](#) <sup>(47)</sup>
- » [原来浏览器原生支持JS Base64编码解码](#) <sup>(35)</sup>
- » [妙法攻略：渐变虚框及边框滚动动画的纯CSS实现](#) <sup>(33)</sup>
- » [炫酷H5中序列图片视频化播放的高性能实现](#) <sup>(31)</sup>
- » [CSS scroll-behavior和JS scrollIntoView让页面滚动平滑](#) <sup>(30)</sup>
- » [windows系统下批量删除OS X系统.DS\\_Store文件](#) <sup>(26)</sup>
- » [写给自己看的display: flex布局教程](#) <sup>(26)</sup>

## 猜你喜欢

- [humans.txt-网站相关人员信息记录的idea](#)
- [面向设计的半封装web组件开发\(概要版\)](#)
- [基于原生HTML的UI组件开发](#)
- [基于CSS color属性的静态UI组件重构策略](#)
- [翻译 - CSS继承详解](#)
- [AMCSS（CSS属性模式）开发简介](#)
- [jQuery之replace字符串替换实现不同尺寸图片切换](#)
- [拜拜了,浮动布局-基于display:inline-block的列表布局](#)
- [去除inline-block元素间间距的N种方法](#)
- [基于vw等viewport视区单位配合rem响应式排版和布局](#)
- [SVG图标颜色文字般继承与填充](#)