

粉丝群第1期CSS小测点评与答疑

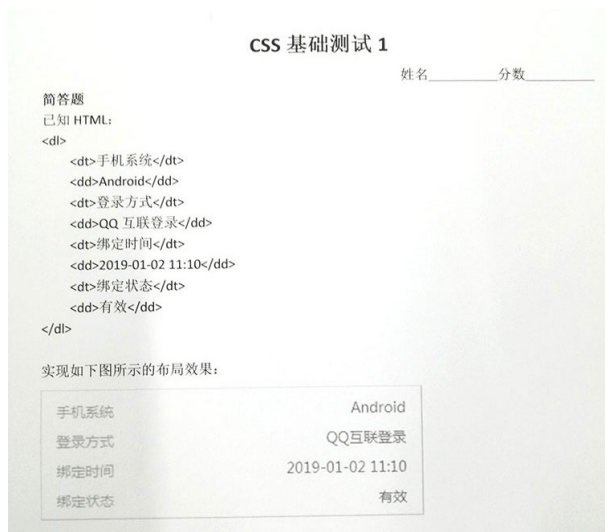
这篇文章发布于 2019年01月12日, 星期六, 20:35, 归类于 [CSS相关](#)。阅读 1923 次, 今日 36 次 [11 条评论](#)

by zhangxinxu from <https://www.zhangxinxu.com/wordpress/?p=8375>

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、CSS小测第一期题目

题目如下图, 一个很简单也很常用的布局:



左右排布的列表, 左侧是标签信息, 右侧是描述信息。

在微信粉丝群里面发出题目后, 总共收到了将近50人的回答, 完整回答列表可以参见此issues (发现有几个回答已经被原作者删掉了, 其实没必要的): <https://github.com/zhangxinxu/quiz/issues/1>

也欢迎关注这个github项目。

今天上午在B站进行了答疑直播, 本文是整理后的文字版。

二、格式与缩进的问题

Github提交信息天然支持Markdown语法, 对于CSS代码, 很多人直接就一段文字往里面复制, 结果文字密密麻麻, 让我看得很吃力。这其实就跟团队协作的代码一样的, 要有让别人可以轻松阅读你代码的意识。Github天然支持开发语言代码高亮, 大家可以用起来, 例如CSS可以使用下面的语法:

```
```css
/* 你的CSS */
```
```

现在看代码都已经高亮了, 那是我周五晚上一个一个编辑过的。😁

然后很多人不注意缩进, 看起来也非常难受:

虽然都是小事，但……怎么说呢，那些在职场上走得更高更远的人，往往是把那些小事都做到尽善尽美的人。

由于每个行为每个个体的行为是不可预料的，所以下一次发布小测题的时候我会强调一下代码排版和格式的问题，否则苦的还是自己。

三、答题点评之值得商榷的实现

1. 通配符reset的问题

不少回答出现了下面的CSS：

```
* { padding: 0; margin: 0; }
```

这是一直偷懒的CSS reset写法，我个人是不推荐这种写法的，一来带来比较多的资源开销，而且这些开销完全是没有必要的，有种为了几棵树砍掉整个森林的感觉，在所有的HTML标签中，默认有padding值的屈指可数，完全没有必要使用通配符进行重置；二来，对于有些元素，默认的 `margin` 属性是有用的，比方说表单元素中的单选框和复选框，其默认margin值可以和后面文字保持合理的间距，如果使用通配符重置，就会挤成一坨，阅读体验并不好。

2. 容器定高的问题

有人在实现的时候，容器写死了高度，实际开发的时候我们不能保证需求会不会变动，比方说增加一个条目的数据。如果我们的容器高度是定死的，那必然增加新条目的时候会出现布局上的bug，会降低容错性和可维护性，因此，避免定高。

3. 50%的问题

可能有将近1/3的人，在实现这种布局的时候，都是 `dt` 和 `dd` 宽度各占50%，无论是左右浮动，`flex` 布局或者使用 `inline-block` 排列。

示意如下：

```
dt, dd{
    width: 50%;
}
dt{
    float: left;
}
dd{
    float: right;
    text-align: right;
}
```

如果是应付这条题目呢，可以说是勉强及格，但是如果放在实际开发中，则局限性就会比较大。由于 `d` 标签中的内容是动态的，可能会出现长段的类似于备注这样的选项，此时 `dd` 50%宽度就会有问题，在移动端，我们显示的宽度可能就只有300多像素，于是，布局可能会表现成这样：



而且根据我的测试，一般容器的宽度小于290像素，时间也会掉下来，因为，50%并不是一个实际开发中的最佳取值，可以调整下。比较好的做法是，左侧安全宽度，右侧自动分配剩余空间。

4. absolute实现的问题

这是一个使用dd标签进行绝对定位的案例：



直播的时候讲这个案例，遇到点突发状况，dd标签标签元素全部挤成一团，我起初以为是选择器用的不当，修改了一番，发现还是没有任何效果，妈呀，这么多人看着呢，有点尴尬呀。后来打开控制台一看样式解析，当场吐血三升，top属性值的单位居然没有写！完全没有想到会这里会缺个单位[捂脸]。

使用dd标签一个一个绝对定位，最大的问题在于，维护性太糟糕了，如果要新增一个条目必然要重新再写一段CSS，如果这里有20行数据，难道还要再增加20多个语句？

实际上，dd标签绝对定位是可以使用的，但是，没有必要使用top进行一个一个定位，如下修改：



其他那些洋洋洒洒的CSS都是多余的，就下面这几行足够了，兼容性非常OK。

```
dl {  
  border: 1px solid #000;  
  position: relative;  
}  
dd {  
  position: absolute;  
  right: 0;  
  margin: -1.2em 0 0 0;  
}
```

为什么有用，这里不展开，《CSS世界》这本书的绝对定位部分有解答，有兴趣可以买一本看看。

5. relative实现的问题

写这个定位方法的回答已经被原作者删除了，我觉得很可惜，首先就这一位作者想到了relative定位，说明其思路易于常人，非常开阔，是个有创造力的人；再者采用relative定位的想法其实挺好的，有时候 `relative` 定位非常管用，有种天衣无缝配合的舒畅。说可惜还有另外一个重要原因是仿佛看到了一个内心敏感的人，我担心以后的小测回答，他不会再回答任何问题，因为怕暴露自己实现的不足，让别人看到自己的短处，这其实是很糟糕的。快速成功的捷径之一，就是勇于犯错，主动犯错，发现自己的不足，并加以改正，绝对提高的非常快。如果你什么都压在心里，放弃了这次别人给你指正的机会，永远意识不到自己哪方面还有所欠缺，何来的学习，成长与提高呢？

说的有点多了，希望是自己想多了！

回到这里，`relative` 定位有时候非常管用，只是这里不太合适：

```
dd {  
  position: relative;  
  top: -20px;  
}
```

最大的问题在于 `relative` 元素定位的时候原本占据的空间他是依然保留的，于是会留下一片空白区域。这里，我们可以把 `top` 改成 `margin-top` 效果就可以了。

6. 左右浮动的问题

还有的实现是直接左右浮动，没有定宽：

```
dt{
```

```
float: left;
clear: left;
}
dd{
float: right;
clear: right;
}
```

这个方法要比左右各50%宽度要好一些，适用的场景要更广一些。

先说些题外话，在《CSS世界》中提到过，没有任何理由使用 `clear:left` 和 `clear:right`，直接使用 `clear:both` 代替。

直播的时候，我想强调一下这个观点，然后就把后面 `clear:right` 改成 `clear:both`，结果布局错位了，当时我就慌了，难道我的研究结论有问题？刚刚又测了一下，发现是被误导了，这里的 `clear:right` 压根就是多余的，没有任何作用。所以，更好的写法是这样：

```
dt{
float: left;
clear: both;
}
dd{
float: right;
}
```

但是，如果把极端情况考虑在内，单纯的左右浮动啊，容易产生错位的问题，例如，我们右侧的描述内容较多的时候：



这就引出了下面一个比较重要的点，一个好的布局实现，应该要能应付各种极端场景。

5. 有必要考虑极端内容

对于文本内容而言，所能出现的极端场景，包括下面三种：

1. 文字内容很多；
2. 连续的一串英文字符；
3. 没有文字内容。

如果本文的小测题是一道面试题的话，最终对候选人的评价最加分的不是用了什么新技术，也不是用了什么稀奇古怪的奇巧淫技，而是能否预知到可能遇到的场景并在代码层面做好容错处理。这不仅可以体现出足够的开发经验，还能体现出全局意识，以及非常重要的基本功。

我们一个一个来。

首先是连续英文字符。这个简单，我们可以使用 `word-break` 属性：

```
word-break: break-all;
```

其次是没有文字内容。这个问题实际上是开发的锅，在内容输出的时候，如果没有数据，应该范围“暂无”，或者“-”这样的缺省信息，但是，多年的经验告诉我，从内容输出和呈现上，一定不要相信后台开发人员，我们自己一定要留一手，否则出现了布局问题，报告单是提到前端这里的。

我们可以这么处理：

```
dd:empty::before {
  content '-';
  color: #999;
}
```

这样，即使dd标签里面没有输出任何文字，也会有字符占位，这样，布局就非常稳固。

最后是文字内容很多，这样就要靠布局策略了。

四、答题点评之比较好的实现

这个布局比较好的实现是这样子的：左侧固定宽度，宽度足够安全，右侧自动填满剩余空间。

根据我多年的开发经验，左侧的标签描述虽然也有动态特性，但是内容却是产品经理决定的，不是用户输入的，因此，个数可控，不要担心会超过四个字，中文就有这个特性。因此，我们可以放心大胆地把左侧dt标签占据空间设定为 5

em。一定要使用 em 单位，不要使用 px 或者 rem，这样，无论容器的字号大小是多少，左侧宽度都不会空间不足，非常弹性，容错性很强。

然后，剩下的就是让右侧dd标签内容宽度自动填满剩余空间。

方法很多。

1. dt标签绝对定位

这是其中一个人的解答，代码虽少，却是非常棒的解答：

```
dt { position: absolute; }
dd {text-align: right; }
```

但是，如果文字内容很多，则dd标签里面内容会和dt发生重叠：



所以，我们加个 5em 大小的左 margin 就可以了。

```
dt { position: absolute; }
dd {text-align: right; margin-left: 5em; }
```

效果如下截图：



此方法兼容性非常好，低版本IE浏览器也支持，但 position:absolute 元素的层叠顺序比较高，如果页面布局复杂，出现了元素层叠的场景，则此方法需要斟酌下，因为很可能会增加布局的复杂度（额外的 z-index 进行层级控制）。

2. Flex布局实现

Flex布局也能实现我们想要的效果，代码如下：

```
dl {
  display: flex;
  flex-wrap: wrap;
}
dt {
```

```
width: 5em;
}
dd {
width: calc(100% - 5em);
text-align: right;
}
```

效果如下截图：



关于Flex布局，如果不了解，可以参见我之前写的文章：[“写给自己看的display:flex布局教程”](#)。

Flex布局的优点是布局的呈现傻白甜，很好理解。不足就是会存在些许兼容性问题，在一些老旧的Android手机上。

3. Grid布局实现

Grid布局实现是容错性最强，语义最佳的方法，其最大的优点是，左侧的标签描述文字，就是5个汉字，6个汉字，布局依然坚挺。

代码如下：

```
dl {
display: grid;
grid-template-columns: auto 1fr;
grid-column-gap: 1em;
}
dd {
text-align: right;
}
```

效果如下截图：



关于Grid布局，如果不了解，可以参见我之前写的文章：[“写给自己看的display:grid布局教程”](#)。

如果没有兼容性方面的限制，则是最佳实现。

4. float浮动实现

有人就用了下面的方法实现，对说明对CSS基础知识还是比较了解的。

```
dt {
width: 5em;
float: left;
}
dd {
text-align: right;
overflow: hidden;
}
```

关键是这里的 `overflow:hidden`，文字内容再多也不会浮动环绕，具体原理可以见这篇文章[“CSS深入理解流体特性和BFC特性下多栏自适应布局”](#)。

效果如下截图：

此方法兼容性很不错，直到IE7浏览器都支持，IE6不支持，如果要兼容IE6，可以试试加一句 `_display: inline-block`。

5. 借助原生流体特性实现

这个方法是我最最后补充的没有人提到的方法，是最最简单的实现：

```
dd {  
    margin: -1.5em 0 0 5em;  
    text-align: right;  
}
```

就结束了。

宽度自适应，字号大小自适应，文字个数自适应。

效果如下截图：




此方法兼容性最强，上至IE6浏览器都支持，代码最少，各种优点也都有，也不需要掌握什么flex布局，grid布局，也不需要了解什么BFC之类概念，就一个简简单单的 `margin` 属性就搞定了，是实际项目开发中的最佳实现。然而，至少在答题的这波人中，却没人知道这个方法，什么原因呢？


总结一下

以上五种方法，虽然代码是不同的，但是最终实现的效果却是一模一样的。

眼见为实，您可以狠狠的点击这里：[CSS小测1比较好的5种布局实现demo](#)

demo页面可以调整容器的宽度和字号大小，我们可以看到所有5个布局都表现良好，如下GIF录屏示意：

宽度调节(240-414): 

字号调节(12-16): 

1. absolute实现

| | |
|------|--------------------|
| 手机系统 | Android |
| 登录方式 | QQ互联登录 |
| 绑定事件 | 2019-01-02 00:01 |
| 绑定状态 | 成功 |
| 备注 | 如果需要修改绑定，请联系HR进行修改 |

2. flex实现

| | |
|------|--------------------|
| 手机系统 | Android |
| 登录方式 | QQ互联登录 |
| 绑定事件 | 2019-01-02 00:01 |
| 绑定状态 | 成功 |
| 备注 | 如果需要修改绑定，请联系HR进行修改 |

3. grid实现

| | |
|------|--------------------|
| 手机系统 | Android |
| 登录方式 | QQ互联登录 |
| 绑定事件 | 2019-01-02 00:01 |
| 绑定状态 | 成功 |
| 备注 | 如果需要修改绑定，请联系HR进行修改 |

4. float实现

| | |
|------|--------------------|
| 手机系统 | Android |
| 登录方式 | QQ互联登录 |
| 绑定事件 | 2019-01-02 00:01 |
| 绑定状态 | 成功 |
| 备注 | 如果需要修改绑定，请联系HR进行修改 |

5. 流体特性实现

| | |
|------|--------------------|
| 手机系统 | Android |
| 登录方式 | QQ互联登录 |
| 绑定事件 | 2019-01-02 00:01 |
| 绑定状态 | 成功 |
| 备注 | 如果需要修改绑定，请联系HR进行修改 |

实现完整代码如下：


```

/* 公共部分 */
dl {
    line-height: 1.5;
    margin: 0; padding: 10px;
    border: 1px solid #ccc;
    background-color: #fff;
}
dd {
    word-break: break-all;
    text-align: right;
    margin-left: 0;
}
dd:empty::before {
    content: '-';
    color: #999;
}
/* absolute实现 */
dt {
    position: absolute;
}
dd {
    margin-left: 5em;
}
/* flex实现 */
dl {
    display: flex;
    flex-wrap: wrap;
}
dt {
    width: 5em;
}
dd {
    width: calc(100% - 5em);
}
/* grid实现 */
dl {
    display: grid;
    grid-template-columns: auto 1fr;
    grid-column-gap: 1em;
}
/* float实现 */
dt {
    width: 5em;
    float: left;
}
dd {
    overflow: hidden;
}
/* 流体特性实现 */
dd {
    margin: -1.5em 0 0 5em;
}

```

五、关于粉丝群、小测与直播

上周新建了一个粉丝群，周三发小测题目，每周依次是CSS、DOM和JS，周六上午答疑，答疑方式是直播加群聊。我估计下周就会群满，到时候想进来估计也进不来了。想入群的可以加我好友：zhangxinxu-job，申请信息“入群”，最好带上自己的姓名，方便我备注。

直播地址是：<https://live.bilibili.com/21193211>

答疑直播时间为每周六上午10:00-11:00，有时候睡懒觉可能会延后一点。

入群和直播都是免费的，如果你觉得有所收获，想要表示感谢，可以关注我的公众号，或者买本我的《CSS世界》就可以了。

《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](#)。



« 分享三个纯CSS实现26个英文字母的案例

常见的CSS图形绘制合集 »

猜你喜欢

- 写给自己看的display: grid布局教程
- 写给自己看的display: flex布局教程
- HTML CSS列表元素ul,ol,dl的研究与应用
- 我是如何对网站CSS进行架构的
- word-break:break-all和word-wrap:break-word的区别
- 借助wbr标签实现连续英文字符的精准换行
- 小tips: 纯CSS实现打字动画效果
- CSS样式分离之再分离
- CSS 相对/绝对(relative/absolute)定位系列 (三)
- CSS 相对/绝对(relative/absolute)定位系列 (四)
- :after伪类+content内容生成经典应用举例

分享到:

0

标签: absolute, css reset, css相关, dl, flex, float, grid, relative, word-break, 宽度自适应, 小测, 布局

发表评论 (目前11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.

大嘴说道:

2019年01月16日 10:57

今天过来看大神的微博, 先去自己写了一遍例题, 再看大神的讲解, 很棒, 感觉打开了很多思路

回复


2.

webxy说道:

2019年01月15日 16:29

对方账号异常, 无法添加朋友。

回复

张鑫旭说道:

2019年01月16日 22:49

再试试吧, 每天有很多人加的。

回复


3.

lulu说道:

2019年01月15日 10:19

你微信账号异常加不了好友了大哥

回复

张鑫旭说道:

2019年01月16日 22:49

再试试吧, 每天有很多人加的。之前应该是加的人太多。

回复


4.

mfk说道:

2019年01月14日 09:31

既然说到用 calc 的话, dt和dd宽度相当于固定了。可能直接对dt和dd应用display:inline-block应该也是可以的。这种方式可能还要对“换行符也占空间”再处理一下。

回复

mfk说道:

2019年01月14日 09:38

此外, 对dl加display:table。对dt和dd使用display:table-cell, 应该也是可以的。

回复


5.

董懂说道:

2019年01月14日 09:22

添加不上, 一直提示“操作太快了, 请休息一下……”

回复

张鑫旭说道:

2019年01月16日 22:56

再试试吧, 之前是加的人一下子太多。

回复



6. d说道:

2019年01月12日 23:31



参加了这次答题，提交了一个类似第四个浮动的实现，但是当换行后的情况没有考虑全面，没有触发bfc使内容对齐。记得直播的时候提到过flex有一种自动填充剩余空间的属性，看总结里用的还是计算宽度，是flex并没有合适这一情况的属性吗？

回复

张鑫旭说道:

2019年01月12日 23:40



有，但是对DOM结构有要求，需要是dl>dt+dd，dl>dt+dd，...的结构。

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 (74)
- » 超级强大的SVG SMIL animation动画详解 (42)
- » 小tips: 纯CSS实现打字动画效果 (40)
- » 粉丝群第1期CSS小测点评与答疑 (36)
- » 找到适合自己的前端发展方向 (35)
- » 让所有浏览器支持HTML5 video视频标签 (32)
- » 分享三个纯CSS实现26个英文字母的案例 (28)
- » 好吧，CSS3 3D transform变换，不过如此！ (24)
- » 未来必热：SVG Sprite技术介绍 (23)
- » CSS/CSS3 box-decoration-break属性简介 (23)

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

- 写给自己看的display: grid布局教程
- 写给自己看的display: flex布局教程
- HTML CSS列表元素ul,ol,dl的研究与应用
- 我是如何对网站CSS进行架构的
- word-break:break-all和word-wrap:break-word的区别
- 借助wbr标签实现连续英文字符的精准换行
- 小tips: 纯CSS实现打字动画效果
- CSS样式分离之再分离

- CSS 相对/绝对(relative/absolute)定位系列 (三)
- CSS 相对/绝对(relative/absolute)定位系列 (四)
- .after伪类+content内容生成经典应用举例