你的页面的语义化真的好么?

我们喜欢使用 标签。它们已经存在了几十年，这几十年来，当需要将一些内容包裹起来达到（添加）样式或者布局目的的时候，它们成为首选元素。查看线上站点时，看到像下面这些内容的情况依旧很常见:

```
<div class="container" id="header">
    <div class="header header-main">Super duper best blog ever</div>
    <div class="site-navigation">
        <a href="/">Home</a>
        <a href="/about">About</a>
        <a href="/archive">Archive</a>
    </div>
</div>
```

WoW，差不多每个网站都有很多的 div 标签。但是，它有效。我的意思主要是，它具有你需要的结构。并且，我确定在你完成样式添加之后，它看起来会像你想要的那个样子。然而，它有些严重的问题:

1. **可访问性差**:
   许多浏览器非常智能，会尽力解析页面结构，以帮助用户按照页面制作者的意图来引导用户，并为用户提供简单的跳转链接来指引他们到自己关心的页面部分。然而， 标签并没有真正传递有关文档结构的任何有用信息。世界上最聪明的浏览器仍然不是人类，不能指望其解析 class 和 id 属性，或能够识别全世界开发人员命名块元素的奇怪和狂野的方式。我可以识别到 class="article-header-level-2"是一个副标题，但是机器不能。

2. **可读性差**:
   要阅读此代码，你需要仔细扫描类名，从 样板之间挑选出来。一旦你（的代码）深入几个层次，跟踪哪个 结束标记与哪个<div ...>开始标记对应，那就变得很棘手了。你开始非常依赖 IDE 功能，例如着色不同的缩进级别或突出显示匹配的标记以跟踪您的位置，而在较长的文档中，它可能需要在这些功能之上进行大量的滚动。
3. **一致性差**:
   开始新的工作或转移到新项目，并且必须从头学习代码库中使用的让人抓狂的标记，那可能会令人很沮丧。如果每个人都有标准化的方法来标记 web 文档中常见结构，那么在不熟悉代码库的情况下，都可以很容易的浏览 HTML 文件并快速处理它应该展示的内容。

如果真有一个标准的话，他就是我们的 HTML5

HTML5 并不新奇。这是轻描淡写；最初的工作草稿于 2008 年 1 月（11 年前）发布，以征求公众意见，并于 4 年半前，2014 年 10 月份成为一个全面 W3C 的推荐。所以，就像它已经存在了一段时间。

HTML5 的主要进步之一是引入了一组标准化的语义元素。术语“语义”指的是单词或

事物的含义，因此”语义元素“是用于以更有意义的方式标记文档结构的元素，这种方式可以清楚地表明它们的用途和它们在文件中服务的目的是什么。而且重要的是，由于它们是标准化的，定义文档的这些元素可以被每个人使用并理解，包括机器人。

对于 div 的建议：
强烈建议大家将 div 元素视为最后采取的元素，在没有其它元素适合的（情况下）。使用更合适的元素而不是 div 元素可以使读者更容易访问，并且更容易为作者提供可维护性。

我将语义块元素分为两类：**主要结构和内容指标。这些不是标准的条款或者其它条款；我在这篇文章中做了一些（区分）。但我认为这种区分足够有用。**

主要结构

有一个超级常见的模式，可在互联网上的网站，教程甚至 CSS 库中找到，并且有充分的理由。我们经常将最顶层的页面划分为三个区域：页眉、主页和页脚，然后根据需要将这些区域划分为多个区域。我在上面的例子中包含了这个来证明这点：

```
<div class="container" id="header">...</div>
<div class="container" id="main">
    ...
    <div class="article-section">...</div>
    ...
</div>
<div class="container" id="footer">...</div>
```

我已经看过（并且使用过）这种模式很久了，以这种方式构造文档非常有意义，既可以读取 HTML，又可以更加简单地在 CSS 中设置页面样式。页眉和页脚元素页可以使用 PHP 或 Rails/ERB 等语言中的部分模版来更易于使用，因为你可以在整个站点中包含常见的页眉和页脚部分：

```
<?php include 'header.php'; ?>

<div id="main">...</div>

<?php include 'header.php'; ?>
```

所以这就是事情：每个人都认为这是一个很好的模式。这包括 WHATWG 和 W3C 的人员，他们将模式标准化为 HTML5 中的四个新元素，名称非常清晰：<header>, <main>, <footer>和<section>。

<header> 和 <footer>

<header> 和 <footer>元素基本上是双胞胎：它们在规范中的定义非常相似，并遵循相同的规则，关于它们被允许使用的位置，唯一区别在于它们的语义目的：页眉在事物的前面，页脚在事物的末尾。对于事物，我的意思不仅仅是页面的： 这对元素的设计用于文档的任何部分，代表一大块内容，具有明确的开头和结尾。这可以包括表格，文章，文章部分，社交媒体网站上的帖子，卡片等。

页眉和页脚在语义上接近 sectioning root 或 sectioning content 元素。像<body>, <blockquote>, <section>, <td>,<aside>等许多其它元素；如果你想了解完整的列表，就点击上面的链接。辅助技术可以使用这些元素和其它元素生成文档大纲，这可以帮助用户更轻松的访问它。在每个 sectioning root/content 中，你不应该使用超过一个的

<header>或<footer>。（一个就好，不能两个相同）

作为最后说明，<header>经常作为其上下文保存标题元素（<h1>-<h6>）。这不是必须的，但可以帮助将其它相关元素与标题分组，比如链接，图片或子标题，并且可以维持一直的结构，即使标题是<header>中的唯一元素。

好东西：<main>

第三个主要区域元素--<main>很特别。规范中说明了关于<main>的两个非常重要的内容：

文档的主要内容区域包括文档的特定内容，且不包括在一组文档中重复的内容，例如站点导航链接，版本信息，站点的徽标，横幅和搜索表单（除非文档或应用的主功能是一种搜索形式）-- www.w3.org/TR/html5/gr…

所以，<main>是你放置好东西的区域，是页面的重要部分，特别是用户访问此页面的原因（或说目的），而不是您的站点。换句话来说，主要内容。😯😳😷

所有其它东西，徽标、搜索表单和导航栏等都可以在<body>中的<header>或<footer>中，但是在<main>之外。

文档中不能有多个可见的 main 元素。如果文档中存在多个 main 元素，则必须使用隐藏属性隐藏所有其它（main）实例。 -- www.w3.org/TR/html5/gr…

这很独特。和<header>和<footer>（以及其它块元素不同），<main>不能在任意切片内容的整个页面中使用；它应该只被使用一次。或者更确切地说，它可以在文档中多次被使用，但是一次只能看到一个<main>元素，所有其它的（）必须被使用隐藏属性隐藏，如 CSS 中的 display:none。如果您思考下，（你会明白）这在应用程序中预加载视图是种很有用的模式：创建一个新的<mian hidden>，获取用户可能接下来查看的一些内容（例如：系列文中的下一篇，下一张幻灯图放映等），然后，当用户点击链接/按钮加载该视图时，通过在两者上切换隐藏属性，将当前的<main>切换到预加载的（那个）。

在继续之前，我们暂停下并查看上面的示例。如果我们使用<header>，<main>和<footer>作为文章的主要结构，它的外观如下：

```
<header>
    <h1>Super duper best blog ever</h1>
    ...
</header>
<main>
    <h2>Why you should buy more cheeses than you currently do</h2>
    ...
</main>
<footer>
    Contact us!
    <div class="contact-info">this.is.us@example.com</div>
</footer>
```

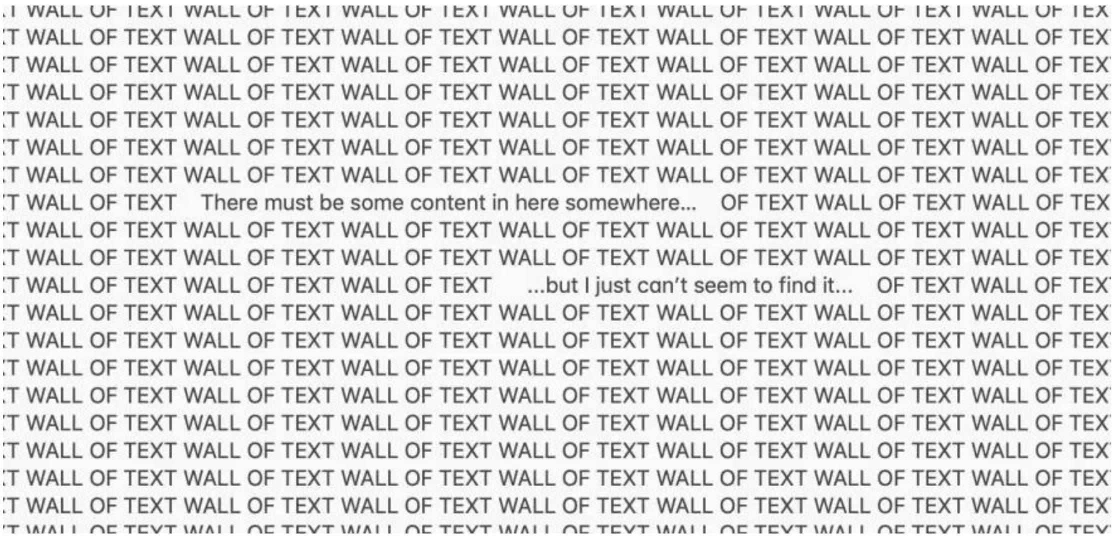
那真的很棒！但是，还有很多工作要做。

分解：<section>

因此，我们为页面提供了一个基本大纲：页眉，页脚和主要内容区域。现在是时候

添加些美妙的内容了。

通常，你会希望将你的内容分解为多个部分，尤其是对像本文这样的大量文本内容，因为没人喜欢阅读这些难以理解的文本墙。



<section>派上用场了。这是在系列规则中最简单的一个：从结构上讲，它基本上只是一个具有特殊含义的<div>。一个<section>开始一个新的"sectioning content"区域，因此它可以有自己的<header>和<footer>。

那么，<section>和普通的旧<div>之间有什么区别，然后，你应该在什么时候使用它们呢？好吧，允许我再次引用规范：

渡一教育 DUYI EDUCATION

你知道，概述来说，HTML5 规范实际上是可读的。它是那个比较可读的规范之一。每当我浏览它以获取快速答复时，我都不可避免地学到一些意想不到的和有用的东西，尤其是当我开始点击链接的时候。有时（你也）试试吧！

简而言之，如果要在目录中列出文档的一部分，请使用<section>。如果没有，请使用<div>或其它元素。

很好，我们已经得到了一个坚固的页面结构。我们已经明确标记了页面的主要内容区域，而不仅仅是单独调整<div>，我们已经调整出了页眉，页脚和章节。但是，肯定还有比我们的文档更多的语义。

让我们来谈谈 HTML5 中添加的一些元素，它们传达的内容语义而不是结构。

整体：<article>

<article>元素用于表示完全独立的内容区域，这些内容可以从页面中提取出来并放入另一个内容中，并且仍然有意义。这可能是文字文章或博客，但也可用于社交媒体帖子，如推特或脸书的墙贴。

HTML5 规范建议文章总有一个标题，标识它是什么，理想的情况下使用标题元素(<h1>-<h6>)。<article>也可以有<header>，<footer>和<section>元素，因此你可以使用它来嵌入一个完整的文档片段，其中包含其它页面中所需的所有结构。

从上面的方式返回到示例，我们使用<article>和我们讨论的其它一些元素来重写带class="article-*"的元素。

```
<article>
    <header>
        <h1>Why you should buy more cheeses than you currently do</h1>
    </header>
```

```
        <section>
            <header>
                <h2>Part 1: Variety is spicy</h2>
            </header>
            <!-- cheesy content -->
        </section>
        <section>
            <header>
                <h2>Part 2: Cows are great</h2>
            </header>
            <!-- more cheesy content -->
        </section>
    </article>
```

这不是比原来更具可读性吗？而且，不仅更容易阅读，它对辅助技术更有用；机器人不能总是弄清楚你的特定类名模式，但是它们可以遵循这种结构。

使用：<nav>

这个元素比其它元素更有名。<nav>旨在清楚地识别页面上的主要导航块，帮助用户围绕站点其余部分找到路径的链接组（例如站点地图或标题中的链接列表）或当前页面（例如目录）。

在我们的示例顶部，让我们将<nav>应用于标题中的那组链接。

```
    <nav>
        <a href="/">Home</a>
        <a href="/about">About</a>
        <a href="/archive">Archive</a>
    </nav>
```

根本不改变结构，但你知道它是什么，一目了然而不需要在 上读物和处理类名来找到它，更重要的是机器人也可以找到它。

接触：<address>

我们要讨论的最后一个元素是<address>。这个元素旨在调出联系信息，它通常在主页<footer>中用于标记企业的邮寄地址，电话号码，客户服务邮箱地址等等。

有趣的是，如何在<address>元素中标记内容的规则是开放的。规范提到有几个其它规范可以解决这个问题，并且提供这种级别的粒度可能超出了 HTML 本身的范围。

常见的解决方案是 RDFa，也是 W3C 规范，它使用标签上的属性来标记数据的不同组件。下面是我们示例中的页脚在标记<address>元素和 RDFa 时可能看起来的样子：

```
    <footer>
        <section class="contact" vocab="http://schema.org/" typeof="LocalBusiness">
            <h2>Contact us!</h2>
            <address property="email">
                <a href="mailto:us@example.com">us@example.com</a>
            </address>
            <address property="address" typeof="PostalAddress">
                <p property="streetAddress">123 Main St., Suite 404</p>
                <p>
                    <span property="addressLocality">Yourtown</span>,
```

渡一教育 DUYI EDUCATION

```
                    <span property="addressRegion">AK</span>,
                    <span property="postalCode">12345</span>
                </p>
                <p property="addressCountry">United States of America</p>
            </address>
        </section>
    </footer>
```

总结

好了，我们已经介绍了很多，我们已经看到很多零零散散的元素应用到我们的例子中。那么，让我们把它们放在一起看看它的样子。

如果你问我（怎么看改造后的内容？），那这比原始例子的可读性高 100 倍，而且对于搜索引擎优化和可访问性目的而言，其效率将提高 100 倍。

这些绝不是 HTML 中唯一的语义元素。有很多其它元素可以帮助你标记和构建你的文本内容，嵌入媒体资源等等。如果你喜欢这个并且希望深入挖掘，这里有一些（标签）可以查看下。你可能认识一些：

```
<aside>
<blockquote>
<cite>
code
<data>
<del>
<figure>
<ins>
<time>
<var>
```

渡一教育 DUYI EDUCATION

这只是一个开始！就像我说的，当你开始阅读 HTML 规范时，很难停下来。它是种非常丰富的语言，我认为人们经常会低估这种语言。