

数学标记语言MathML简介、工具及兼容

这篇文章发布于 2018年10月24日, 星期三, 23:42, 归类于 [HTML相关](#)。阅读 3753 次, 今日 10 次 9 条评论

by zhangxinxu from <https://www.zhangxinxu.com/wordpress/?p=8108>

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。

//xxx: 本文在Firefox或者Safari浏览器下浏览会有更好的体验。

一、MathML简介

MathML指“数学标记语言”, 是XML语言的一个子集, 用来在web网页, 甚至部分软件中显示数学公式。

简言之, 就是使用特殊的类似HTML的标记在网页中显示数学公式。

因为有些数学公式很复杂, 普通HTML根本没法驾驭, 例如下面这个公式:

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{2\pi i(n/T)x} = \sum_{n=-\infty}^{\infty} \hat{f}(\xi_n) e^{2\pi i \xi_n x} \Delta \xi$$

都有哪些MathML标记呢? 可以看下面的定义列表说明 (非[原文访问](#)会数学公式排版异常):

1. 按类别分类的MathML表现元素

顶级元素

<math>

用在最外部包裹, 表示里面的都是数学公式。例如, 就一个变量 x , 则有:

```
<math><mi>x</mi></math>
```

效果是: x

记号元素

<mglyph>

有些现存的Unicode字符是不可用的, 此时就可以使用 `<mglyph>` 替换显示, 可以理解为MathML世界中的图片元素, 支持 `width`, `height` 以及 `alt` 等属性。例如:

```
<math>
  <mi><mglyph src="my-glyph.png" alt="my glyph"/></mi>
</math>
```

<mi>

`mi` 是'math identifier'的缩写, 字面意思数学标识符, 多指函数名, 变量或者符号常量。示意 (下面标识符之间无任何关联, 仅仅示意语义):

```
<math>
  <mi> y </mi>
  <mi> sin </mi>
  <mi mathvariant="monospace"> x </mi>
  <mi mathvariant="bold"> &pi; </mi>
</math>
```

```
</math>
```

效果是： $y \sin x \pi$

<mn>

`mn` 是'math number'的缩写，表示数值，支持各种数值。示意（仅示意，无关联）：

```
<math>
  <mn> 0 </mn>
  <mn> 1.337 </mn>
  <mn> twelve </mn>
  <mn> XVI </mn>
  <mn> 2e10 </mn>
</math>
```

效果是： 0 1.337 twelve XVI 2e10

<mo>

`mn` 是'math operators'的缩写，表示数学操作符，例如加减乘除，各种括号，分号等。示意：

```
<math>
<mrow>
  <mn>5</mn>
  <mo>+</mo>
  <mn>5</mn>
</mrow>

<mrow>
  <mo> [ </mo>
  <mrow>
    <mn> 0 </mn>
    <mo> ; </mo>
    <mn> 1 </mn>
  </mrow>
  <mo> ) </mo>
</mrow>
</math>
```

效果是： $5 + 5 [0 ; 1)$

<ms>

`ms` 是'math string literal'的缩写，表示一个字符串文字，这个字符串需要由编程语言和计算机代数系统来解释。默认情况下，字符串文字显示为用双引号括起来（`"`）；通过使用 `lquote` 和 `rquote` 属性，您可以设置要显示的自定义字符。示意：

```
<math>
  <ms lquote="," rquote="'"> abc </ms>
</math>
```

效果是： abc

<mspace>

`mspace` 是'math space'的缩写，表示空白间距，其尺寸可以通过 `width`，`height` 以及 `depth` 等尺寸控制。示意：

```
<math>
  <mi>x</mi>
  <mspace width="40px"></mspace>
  <mi>y</mi>
</math>
```

```
</math>
```

效果是：xy

<mtext>

MathML `<mtext>` 元素用于呈现没有符号含义的任意文本，例如注释或注解。示意：

```
<math>
  <mtext> 毕达哥拉斯定理 </mtext>
  <mtext> /* 在这里注释 */ </mtext>
</math>
```

总体布局

<menclose>

MathML `<menclose>` 元素用来把内容封闭在指定的记号中（通过设置 `notation` 属性）。示意一个根号效果：

```
<math>
  <menclose notation="radical">
    <msup><mi>a</mi><mn>2</mn></msup>
    <mo>+</mo>
    <msup><mi>b</mi><mn>2</mn></msup>
  </menclose>
</math>
```

在支持的浏览器，例如Firefox浏览器中，表现会如后面这样： $\sqrt{a^2 + b^2}$

`<menclose>` 元素支持的封闭符合非常多，款式各种各样，让人大开眼界，有兴趣可以看[MDN这个文档](#)。

<merror>

MathML `<merror>` 元素用来标记这个计算公式或者表达式是错误的，浏览器会通过边框和背景色样式加以明显区分。示意：

```
<merror>
  <mrow>
    <mtext>除以0: </mtext>
    <mfrac>
      <mn> 1 </mn>
      <mn> 0 </mn>
    </mfrac>
  </mrow>
</merror>
</math>
```

当前浏览器实时效果是：除以0: 1 0

Firefox浏览器截图效果是：

<mfenced>

MathML `<maction>` 元素可以添加自定义的对称闭合符号（如括号），以及自定义分隔符（如逗号）。示意：

```
<math>
  <mfenced open="{" close="}" separators=";;;">
    <mi>a</mi>
  </mfenced>
</math>
```

```
<mi>b</mi>
<mi>c</mi>
<mi>d</mi>
<mi>e</mi>
</mfenced>
</math>
```

当前浏览器实时效果是： abcde

在支持的浏览器下面会这样呈现：

<mfrac>

MathML `<mfrac>` 指 'math fraction'，表示分数，也即是除法。示意：

```
<math>
  <mfrac bevelled="true">
    <mfrac>
      <mi> a </mi>
      <mi> b </mi>
    </mfrac>
    <mfrac>
      <mi> c </mi>
      <mi> d </mi>
    </mfrac>
  </mfrac>
</math>
```

当前浏览器实时效果是： a b c d

在支持的浏览器下面会这样呈现：

<mpadded>

MathML `<mpadded>` 元素用来给封闭内容增加格外的padding一遍进行位置和尺寸的调整。

<mpantom>

Phantom这个词是幻影的意思。因此MathML `<mpantom>` 表示当前元素是个“幻影”，也就是不可见，但是位置尺寸都保留，类似CSS世界中的 `visibility:hidden`。示意：

```
<math>
<mrow>
  <mi> x </mi>
  <mo> + </mo>
  <mpantom>
    <mi> y </mi>
    <mo> + </mo>
  </mpantom>
  <mi> z </mi>
</mrow>
</math>
```

当前浏览器实时效果是： x + y + z

在支持的浏览器下面会这样呈现：

<mroot>

MathML `<mroot>` 元素表示根号。示意：

```
<math>
```

```
<mroot>
  <mi>x</mi>
  <mn>3</mn>
</mroot>
</math>
```

当前浏览器实时效果是： x^3

在支持的浏览器下面会这样呈现：

<mrow>

MathML `<mrow>` 元素用于对子表达式进行分组，子表达式通常包含一个或多个运算符及其各自的操作数（例如和）。此元素水平呈现。上面很多案例已经有示意，这里不重复演示。

<msqrt>

MathML `<msqrt>` 元素表示开平方根。示意：

```
<math>
  <msqrt>
    <mi>x</mi>
  </msqrt>
</math>
```

当前浏览器实时效果是： \sqrt{x}

<mstyle>

MathML `<mstyle>` 元素用于更改其子项的样式。例如设置颜色为茶色：

```
<math>
  <mstyle displaystyle="true" mathcolor="teal">
    <mrow>
      <mi>i</mi>
      <mo form="infix">=</mo>
      <mn>1</mn>
    </mrow>
  </mstyle>
</math>
```

在支持的浏览器下面会这样呈现：

脚本和限制元素

<mmultiscripts>

MathML `<mmultiscripts>` 元素允许你创建一个类似张量对象。“张量”（tensor）理论是数学的一个分支学科，在力学中有重要应用。张量是一个定义在的一些向量空间和一些对偶空间的笛卡儿积上的多重线性映射，其坐标是 n 维空间内，有 n 个分量的一种量，其中每个分量都是坐标的函数，而在坐标变换时，这些分量也依照某些规则作线性变换。

这个元素的理解难度已经超出我的能力范畴，溜了溜了……

<mover>

MathML `<mover>` 元素用来在表达式的上方添加重音或限制。示意：

```
<math>
  <mover accent="true">
    <mrow>
      <mi>x</mi>
      <mo>+</mo>
    </mrow>
  </mover>
</math>
```

```

    <mi> y </mi>
    <mo> + </mo>
    <mi> z </mi>
  </mrow>
  <mo> &#x23DE; <!--上花括号--> </mo>
</mover>
</math>

```

当前浏览器实时效果是： $x + y + z^{\frown}$

在支持的浏览器下面会这样呈现：

<msub>

MathML `<msub>` 好理解，表示下标。示意：

```

<math>
  <msub>
    <mi>X</mi>
    <mn>1</mn>
  </msub>
</math>

```

当前浏览器效果： X_1

<msubsup>

MathML `<msubsup>` 元素表示同时上标和下标。示意：

```

<math displaystyle="true">
  <msubsup>
    <mo> &#x222B; <!-- 积分 --> </mo>
    <mn> 0 </mn>
    <mn> 1 </mn>
  </msubsup>
</math>

```

当前浏览器实时效果是： \int_0^1

在支持的浏览器下面会这样呈现：

<msup>

MathML `<msup>` 表示上标，参见下标 `<msub>`

<munder>

MathML `<munder>` 元素和 `<mover>` 对立，表示在表达式下方强调或者限制。示意：

```

<math>
  <munder accent="true">
    <mrow>
      <mi> x </mi>
      <mo> + </mo>
      <mi> y </mi>
      <mo> + </mo>
      <mi> z </mi>
    </mrow>
    <mo> &#x23DF; <!--下花括号--> </mo>
  </munder>
</math>

```

当前浏览器实时效果是： $x + y + z_{\smile}$

在支持的浏览器下面会这样呈现：

<munderover>

MathML `<munderover>` 元素表示同时出现在上面和下面。示意：

```
<math displaystyle="true">
  <munderover >
    <mo> ∫ </mo>
    <mn> 0 </mn>
    <mi> ∞ </mi>
  </munderover>
</math>
```

当前浏览器实时效果是： \int_0^∞

在支持的浏览器下面会这样呈现：

表格数学

<mlabeledtr>

MathML `<mlabeledtr>` 元素用于表示行内label标签，要么在左侧要么在右侧（由 `<mtable>` 元素上的 `side` 属性决定）。`<mlabeledtr>` 的子元素必须是 `<mttd>` 元素。第一个子节点是label标签，而所有其他子节点表示行内容，并且与 `<mtr>` 元素的子节点相同。根据我在Firefox浏览器下的测试，应当作为label标签的第一个子节点并未显示。

<mtable>

MathML `<mtable>` 元素可以理解为HTML届的 `<table>` 元素，虽然属性出入较大，但排版布局表现很类似。

<mttd>

MathML `<maction>` 元素可以理解为HTML届的 `<td>` 元素，表示单元格，两者排版效果很类似。

<mtr>

MathML `<maction>` 元素可以理解为HTML届的 `<tr>` 元素，表示表格行，两者排版布局效果很类似。

未分类元素

<maction>

MathML `<maction>` 元素可以添加交互行为，例如展现的是表达式，点击一下出现的是表达式计算结果。示意一个1+1=2的效果：

```
<math>
  <maction actiontype="toggle">
    <mrow>
      <mn>1</mn>
      <mo>+</mo>
      <mn>1</mn>
    </mrow>
    <mrow>
      <mo>=</mo>
      <mn>2</mn>
    </mrow>
  </maction>
</math>
```

效果如下GIF:



2. 语义注释

在MathML中，有两种标记数学的方法：Presentation MathML用于控制数学方程或表达式的布局，也就是用户看到的外观（上面介绍的部分），而Content MathML用于使数学方程或表达式更语义化，便于计算机理解。MathML元素 `<semantics>`，`<annotation>` 和 `<annotation-xml>` 用于联合展示和内容标记，以及提供数学表达式的布局信息和语义。

`<annotation>`

MathML `<annotation>` 表示注解与说明。

`<annotation-xml>`

MathML `<annotation-xml>` 指明函数或表达式结构。

`<semantics>`

MathML `<semantics>` 表示里面含有语义化信息。

上面3元素看下面一个案例就能知道大概作用了：

```
<math>
  <semantics>
    <!-- Presentation MathML -->
    <mrow>
      <msup>
        <mi>x</mi>
        <mn>2</mn>
      </msup>
      <mo>+</mo>
      <mi>y</mi>
    </mrow>
    <!-- Content MathML -->
    <annotation-xml encoding="MathML-Content">
      <apply>
        <plus/>
        <apply>
          <power/>
          <ci>x</ci>
          <cn type="integer">2</cn>
        </apply>
        <ci>y</ci>
      </apply>
    </annotation-xml>
    <!-- 图像注解 -->
    <annotation encoding="image/png" src="some/path/formula.png"/>
    <!-- Tex注解 -->
    <annotation encoding="application/x-tex">
      x^{2} + y
    </annotation>
  </semantics>
</math>
```

其他

- 可以看到都以MathML标记都是字母 `m` 开头。
- 上面所有展示的MathML代码效果都可以在这个页面看到，您可以狠狠的点击这里：[MathML示意图mo](#)

二、浏览器的兼容处理

首先看下兼容性，Chrome浏览器在版本24的时候曾经昙花一现支持了下，不过很快就取消了支持，据说是出于安全考虑。



据说IE浏览器可以安装MathPlayer插件支持下，不过我个人觉得最终效果尔尔。

至于Chrome浏览器，我查找了下，有个名叫mathml.css的项目：<https://github.com/fred-wang/mathml.css>

针对Chrome这类不支持的浏览器使用CSS进行了公式布局的模拟。使用方法可以是直接引入下面JS代码：

```
<script src="//fred-wang.github.io/mathml.css/mspace.js"></script>
```

这个mspace.js会检测当前浏览器是否支持MathML，如果不支持，就会加载mathml.css文件做兼容，但也只是一定程度上的兼容，根据我的实际使用测评，很多标签和属性功能并不支持，然后，最终呈现的排版效果和原本支持的浏览器（如Firefox）相差甚远，只能说凑合使用。

三、在线生成MathML的工具介绍

从第一段的介绍就可以看出MathML非常复杂，是个完整的XML语言体系，其中还有茫茫多的XML属性还没介绍，由于对于复杂公式，我们想要完全手写出来，呵呵，估计要废掉大半的血条。

好在有现成的工具，可以直接生成MathML代码。经过自己一番实践，发现下面这个工具是极好的！

[手写公式自动变MathML](#)

例如，我在画布中写个 $x^2+y^2=z^2$ ，结果右上角MathML代码就出来了，实在太牛了！



四、小众语言，总会有用的

MathML这么语言，虽然小众，但总归有用的，举例来说，我之前写了很多动画相关的文章，例如抛物线运动，矩阵变换介绍等，都需要用到数学公式，如果得到精美的自己想要的公式代码呢？经过这几天学习，我就知道该怎么玩了。

先打开上面介绍的在线生成MathML的工具，手写数学公式，然后把生成的MathML代码在Firefox浏览器中显示，然后截图，一个精致的数学公式效果图就出来了！

要是没有对MathML的学习和研究，肯定不会知道原来还可以这么玩，对吧？

学习呢，不要功利心太强，就算当下看不到效益，但在将来某一天，总会给你带来雪中送炭般的帮助的。

感谢阅读，欢迎交流！

《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](#)。



« 有人考了我一道CSS题目

CSS margin-inline和margin-block区别是什么? »

猜你喜欢

- 了解: 使用CSS namespace进行分隔
- HTML CSS列表元素ul,ol,dl的研究与应用
- HTML5新增的form属性简介
- 不使用JavaScript让IE浏览器支持HTML5元素
- 基于原生HTML的UI组件开发
- 了解HTML/HTML5中的download属性
- IE6下png背景不透明问题的综合拓展
- 复选框单选框与文字对齐问题的研究与解决
- 对html与body的一些研究与理解
- 关于Google圆角高光高宽自适应按钮及其拓展
- ajax与HTML5 history pushState/replaceState实例

分享到: 1

标签: html, HTML5, MathML

发表评论 (目前9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. **jg**说道:
2018年11月25日 16:41
这ML太复杂了, 还是LaTeX表示方便理解
[回复](#)



2. **7P_LonG**说道:
2018年11月12日 18:25
想做个 可插入数学符号的 文本编辑器。。
感觉这个还是不够用
[回复](#)
3. **老猪**说道:
2018年11月12日 15:29
前年用过，还挺好的。
[回复](#)
4. **网站建设**说道:
2018年11月5日 11:25
有时候用工具不错
[回复](#)
5. **啦啦啦**说道:
2018年11月3日 13:10
mn是'math operators'的缩写

这个有笔误哦
[回复](#)
6. **WangNianyi2001**说道:
2018年10月28日 18:36
张量.....就是向量和矩阵的推广
比如把正常的标量，数，叫做零阶张量；把向量叫一阶张量；矩阵叫二阶张量.....以此类推。
几阶张量就是把数往几个维度上排，至于为什么叫『阶』而不叫『维』，是因为已经被代表方阵尺寸的量占用了。
[回复](#)
7. **mfk**说道:
2018年10月25日 13:06
那个在线工具太牛了，纯js实现手写识别，正确率还非常高
[回复](#)
8. **Draco**说道:
2018年10月25日 11:36
渲染公式还是Mathjax和Katex强，对于目前主流的浏览器也能兼容，MathML感觉路还很长.....不过基于web的图形化公式编辑器倒真是少见，即使有也很难用，碰到过最好用的还是基于swf的。
[回复](#)
- harttle**说道:
2018年10月25日 15:54
> 渲染公式还是Mathjax和Katex强，对于目前主流的浏览器也能兼容，MathML感觉路还很长...
这不能比吧。MathML 是语言，MathJax 和 Katex 是编译器。况且 MathJax 生成的就是 MathML 或 HTML。
[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](#) (179)
- » [粉丝群第1期CSS小测点评与答疑](#) (113)
- » [未来必热: SVG Sprite技术介绍](#) (111)
- » [HTML5终极备忘大全 \(图片版+文字版\)](#) (85)
- » [让所有浏览器支持HTML5 video视频标签](#) (83)
- » [Selectivizr-让IE6~8支持CSS3伪类和属性选择器](#) (80)
- » [CSS3下的147个颜色名称及对应颜色值](#) (78)
- » [小tips: 纯CSS实现打字动画效果](#) (73)
- » [写给自己看的display: flex布局教程](#) (70)
- » [分享三个纯CSS实现26个英文字母的案例](#) (70)

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

- [了解: 使用CSS namespace进行分隔](#)
- [HTML CSS列表元素ul,ol,dl的研究与应用](#)
- [HTML5新增的form属性简介](#)
- [不使用JavaScript让IE浏览器支持HTML5元素](#)
- [基于原生HTML的UI组件开发](#)
- [了解HTML/HTML5中的download属性](#)
- [IE6下png背景不透明问题的综合拓展](#)
- [复选框单选框与文字对齐问题的研究与解决](#)
- [对html与body的一些研究与理解](#)
- [关于Google圆角高光高宽自适应按钮及其拓展](#)

