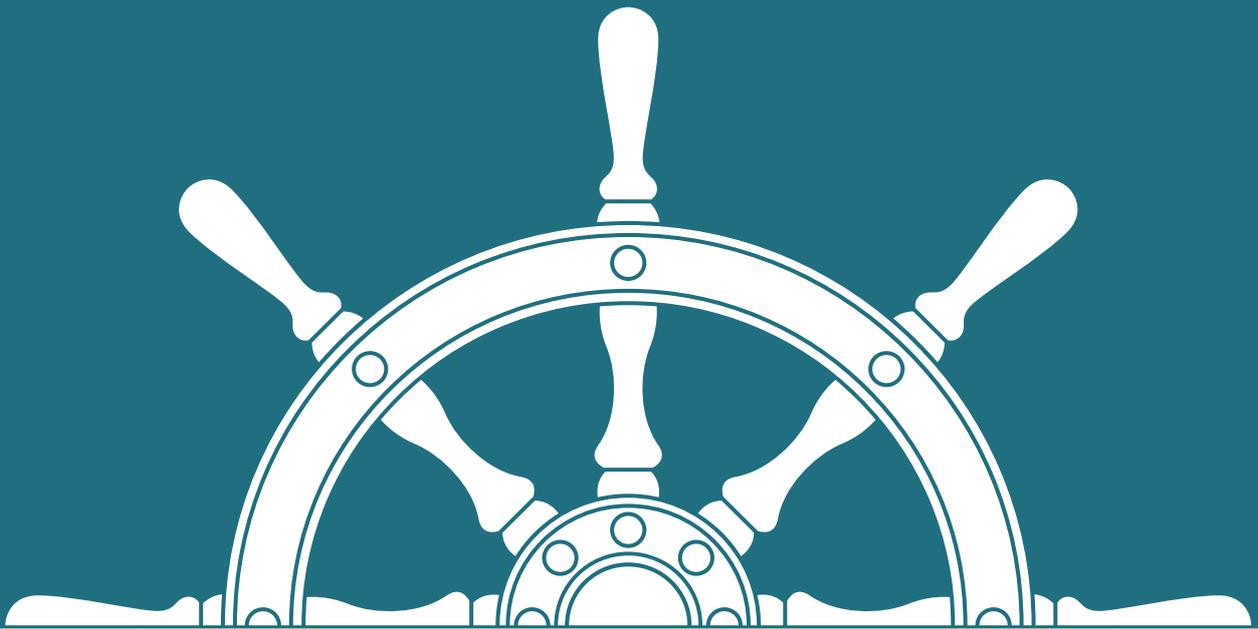


# entwickler

magazin

Ausgabe 7.2023



# KUBERNETES

**KubeOne:** Container einfach und flexibel

**MathML:** Fabelhafte Formeln

**Web:** Sprach- und Bildverarbeitung



© Andrew Angelov/Shutterstock.com

## Grundlagen und Praxis zum Formelsatz mit MathML

# Fabelhafte Formeln

Die Wurzeln der XML-basierten Auszeichnungssprache MathML für Formeln reichen bis zu den Anfängen der Etablierung von HTML zurück. Mangels Unterstützung in Browsern war der produktive Einsatz im Webkontext lange unbefriedigend. Während Firefox und Safari seit einigen Jahren mit brauchbaren Implementierungen aufwarten, zogen Chrome und die darauf aufsetzenden Derivate erst Anfang 2023 nach. Wir schauen uns die nötigen Grundlagen und aktuelle Nutzungsmöglichkeiten an.

von Dr. Thomas Meinike

Die MathML Association twitterte im Januar über den Account [@mathml3](#): „The time has come. Welcome to 2023: The Year of #MathML [...]“ mit dem Verweis auf die gerade erschienene Version 109 des Browsers Google Chrome, siehe [1] und **Abbildung 1**. In dieser Ankündigung steckt bei allem Pathos ein wahrer Kern, nämlich das Ankommen der bereits seit 25 Jahren vom W3C zur Darstellung von mathematischen und wissenschaftlich-technischen Formeln spezifizierten Webtechnologie in den marktüblichen Browsern.

### Weiter Weg in die Browser

Erste Ideen für eine Auszeichnungssprache für Formeln gab es schon 1994. Während eines Working Draft zu

HTML 3.0 tauchten Vorschläge für ein als HTML Math bezeichnetes Konzept auf. 1997 wurde die W3C Math Working Group gegründet. Deren Arbeit mündete 1998 in die erste veröffentlichte MathML-Spezifikation [2]. Die wesentlichen Weiterentwicklungen, die es in der Folge gab, repräsentiert die noch aktuelle zweite Ausgabe der Version 3.0 von 2014 [3]. Bemerkenswert ist auch die im selben Jahr verkündete Adoption von MathML (und SVG) durch den HTML5-Standard. HTML5-fähige Browser sollten also auch Formeln implementieren und zur Nutzung bereitstellen können. Die Entwicklungsabteilungen von Apple und Mozilla haben diesbezüglich ihre Hausaufgaben in ihren Browsern Safari und Firefox einigermaßen schnell gemacht, aber das Fehlen in weiteren populären Produkten hat eine breite Akzeptanz und Nutzbarkeit nicht gerade ge-

fördert. Durch die genannte Initiative von Google sind nun neben Chrome auch Chromium-basierte Derivate wie Brave (Brave Software), Edge (Microsoft), Opera (Opera Software) und Vivaldi (Vivaldi Technologies) direkt zum Rendern von Formelinhalten ausgelegt. **Abbildung 2** beschreibt den Implementierungsstand laut „Can I use“ von Anfang Juli, wobei interessanterweise einst Chrome 24 kurzzeitig ein Auge auf MathML geworfen hatte [4].

Übrigens ist die Technologie seit 2011 auch Teil des Ökosystems für E-Books im EPUB-3.x-Format. An dieser Stelle sei auch auf die bereits länger bestehende Möglichkeit verwiesen, Formeln bei Bedarf über die JavaScript-Bibliothek MathJax zugänglich zu machen [5]. Dafür wird im Wesentlichen ein Einzeiler im *head*-Element des HTML-Dokuments mit dem CDN-Link benötigt.

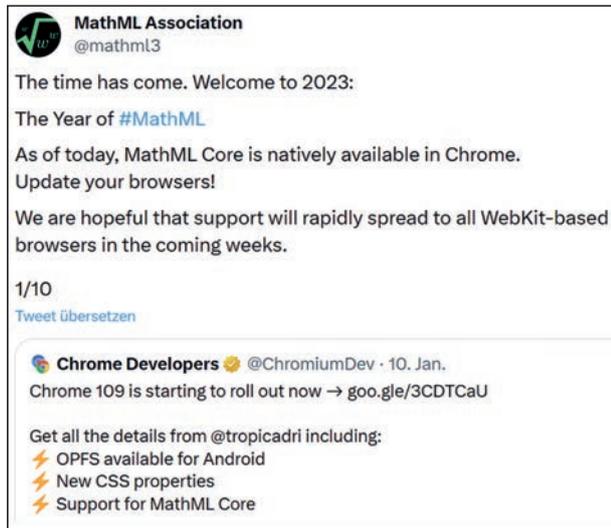


Abb. 1: Tweet der MathML Association vom 10.01.2023



Abb. 2: MathML-Unterstützung in Browsern [4]

<p><b>(F1) mrow, mi, mn, mo</b></p> $x + y = 3$	<p><b>(F2) mfrac</b></p> $\frac{x-1}{x+1}$	<p><b>(F3) msup, msup, msupsub</b></p> $y_i \quad x^2 \quad a_j^k$ $e^{i\pi} + 1 = 0$	<p><b>(F4) msqrt, mroot</b></p> $\sqrt{a+b} \quad \sqrt[3]{c}$ $c = \sqrt{a^2 + b^2}$
<p><b>(F5) munder, mover</b></p> $\overbrace{x+y}^{\vec{v}}$ $\lim_{x \rightarrow 0} \frac{1}{x} = \infty$	<p><b>(F6) munderover (a)</b></p> $\int_a^b \int_a^b \sum_{i=1}^n \sum_{i=1}^n$	<p><b>(F7) munderover (b)</b></p> $\int_a^b$ $\sum_{i=1}^n$	<p><b>(F8) mo für Klammern</b></p> $(a+b)^2$ $\binom{n}{k} = \frac{n!}{(n-k)! \cdot k!}$
<p><b>(F9) mtable, mtr, mtd</b></p> $M = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	<p><b>(F10) mtext, mspace, ms</b></p> <p>if <math>a = 1</math> then <math>b = 2</math></p> <p>"Hallo Welt!"</p>	<p><b>(F11) mmultiscripts, mprescripts, mphantom, mpadded</b></p> $\frac{1}{2} B_i^3$ $\frac{a+c}{a+b+c}$	<p><b>(F12) semantics, annotation</b></p> $\frac{1}{2}$ <p>&lt;!-- Der Bruch einhalb. --&gt;</p>
<p><b>(F13) mstyle, menclose, CSS</b></p> $a + b = c$ $\frac{5!}{3!} = \frac{5 \cdot 4 \cdot \cancel{3!}}{\cancel{3!}} = 20$	<p><b>(F14) display: inline vs. block</b></p> <p>Zur Lösung der quadratischen Gleichung <math>x^2 + px + q = 0</math> dient folgende Formel:</p> $x_{1,2} = -\frac{p}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q}$	<p><b>(F15) Arithmetisches Mittel</b></p> $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{x_1 + x_2 + \dots + x_n}{n}$	<p><b>(F16) Integralrechnung</b></p> $\int_0^\pi \sin x \, dx = [-\cos x]_0^\pi$ $= -(-1) - (-1) = 2$

Abb. 3: MathML-Testformeln

Die Möglichkeiten von MathML 3.0 werden in Firefox und Safari gut ausgeschöpft, während Google und Co. primär die neue, vor der Fertigstellung stehende W3C-Initiative MathML Core adressieren, also den nun allgemein weitgehend problemlos einsetzbaren Kernbereich [6]. Zudem wird an Version 4.0 gearbeitet [7]. Entsprechende Erweiterungen sind vor allem für Dokumentformate der technischen Kommunikation wie DocBook und DITA oder fachspezifische Anwendungen interessant. In diesem Artikel geht es daher um die Möglichkeiten des für Darstellungen eingesetzten „Presentation Markup“ der Core-Variante. Das außerdem existierende Content Markup adressiert die inhaltliche Bedeutung von Formeln und kommt zur Interpretation mit mathematischer Software zum Einsatz.

### MathML Core

Zur Einbindung von einzelnen Formeln oder Teilen davon wird jeweils ein *math*-Element in der HTML-Struktur an gewünschter Stelle eingefügt. Dabei kann die Darstellung einzeilig, z. B. innerhalb eines Textabsatzes, oder in abgesetzter Blockform erfolgen. Dazu dient das Attribut *display* mit den Werten *inline* bzw. *block*, wobei diese Formatierung analog im CSS-Stylesheet über die *display*-Eigenschaft erfolgen kann. Der folgende Code zeigt den Grundaufbau eines *math*-Containers. Dabei ist die Angabe des Namensraumes zumindest in HTML5-Dokumenten optional. Für eine mit anderen Standards kompatible Nutzung ist die grundsätzliche Notation jedoch zu empfehlen.

```
<math xmlns="http://www.w3.org/1998/Math/MathML" display="...">
  <!-- weitere Inhalte ... -->
</math>
```

In den konkreten Einzelbeispielen ist das umschließende *math*-Element nicht vorhanden und wir gehen bis auf eine Ausnahme von der *block*-Variante aus. Unterhalb von *math* existieren ca. 30 Kindelemente, die überwiegend mit dem Buchstaben *m* beginnen und auch in

weiteren Verschachtelungen vorkommen können. Die Bezugnahme auf die in **Abbildung 3** dargestellten Testformeln erfolgt mit der jeweiligen Nummer in runden Klammern als (Fn). Diese sind online unter [8] nachvollziehbar.

### mrow

Das Element *mrow* dient zur Abgrenzung und Gruppierung von Ausdrücken, Termen oder sonstigen zusammenhängenden Teilbereichen einer Formel. Listing 1 beschreibt die einfache lineare Anordnung einer Gleichung, vgl. dazu (F1). Hier wäre *mrow* zur eigentlichen Darstellung noch nicht unbedingt erforderlich, betont jedoch die prinzipielle Funktion des Umschließens von Inhalten.

### mi, mn und mo

Von grundsätzlicher Bedeutung sind die bereits in Listing 1 verwendeten Elemente *mi* für Variablen, Konstanten und Funktionsnamen (Identifer), *mn* für ganzzahlige oder mit Fließkomma versehene Zahlenwerte (Number) sowie *mo* für Grundoperationen wie hier die Addition und auch für das Gleichheitszeichen (Operator).

### mfrac

Brüche werden mit *mfrac* in der typischen Form des Zählers über dem Nenner ausgezeichnet. Die Teile sind gemäß Listing 2 übereinander angeordnet, wobei hier *mrow* zu ihrer Abgrenzung erforderlich ist, vgl. (F2).

### msub, msup und msubsup

Das Tief- bzw. Hochstellen von Indizes, Variablen oder Termen wird häufig benötigt und mit den Elementen *msub* und *msup* realisiert. In Abhängigkeit vom behandelten Inhalt kann wiederum *mrow* zur Aufteilung notwendig sein. Hinzu gesellt sich das Element *msubsup*, das gemeinsam tief- und hochgestellte Angaben erlaubt, wie in Listing 3 im oberen Teil von (F3) gezeigt.

Im unteren Teil von (F3) ist die als Eulersche Identität bekannte Formel ersichtlich. Dort ist das in *mrow*

#### Listing 1

```
<mrow>
  <mi>x</mi>
  <mo>+</mo>
  <mi>y</mi>
  <mo>=</mo>
  <mn>3</mn>
</mrow>
```

#### Listing 2

```
<mfrac>
  <mrow>
    <mi>x</mi>
  </mrow>
  <mo>+</mo>
  <mn>1</mn>
</mfrac>
```

#### Listing 3

```
<msub>
  <mi>y</mi>
</msub>
<msup>
  <mi>x</mi>
  <mn>2</mn>
</msup>
<msubsup>
  <mi>a</mi>
  <mi>j</mi>
  <mi>k</mi>
</msubsup>
```

#### Listing 4

```
<msup>
  <mi>e</mi>
</msup>
<mrow>
  <mi>i</mi>
  <mi>π</mi>
</mrow>
<mo>+</mo>
<mn>1</mn>
<mo>=</mo>
<mn>0</mn>
```

#### Listing 5

```
<msqrt>
  <mi>a</mi>
  <mo>+</mo>
  <mi>b</mi>
</msqrt>
<mroot>
  <mi>c</mi>
  <mn>3</mn>
</mroot>
```

#### Listing 6

```
<mi>c</mi>
<mo>=</mo>
<msqrt>
  <mi>a</mi>
  <mn>2</mn>
</msup>
<mo>+</mo>
<msup>
  <mi>b</mi>
  <mn>2</mn>
</msup>
</msqrt>
```

eingebundene Produkt  $i\pi$  hochgestellt. Da die imaginäre Einheit  $i$  und die Kreiszahl  $\pi$  Konstanten repräsentieren, sind beide in  $mi$  abgelegt (Listing 4).

### msqrt und mroot

Quadratwurzeln werden mit  $msqrt$  beschrieben und  $n$ -te Wurzeln mit  $mroot$ . Der Inhalt unter der Wurzel und bei  $n$ -ten Wurzeln der zusätzliche Wurzelexponent müssen klar abgegrenzt werden. In Listing 5 ist das Vorgehen für beide Wurzelarten ersichtlich und Listing 6 beschreibt die allseits bekannte Pythagoras-Formel in der Umformung als Wurzel Ausdruck, vgl. (F4).

### munder und mover

Das Setzen von Formelteilen oder nur einzelnen Zeichen unterhalb oder oberhalb von Termen ist ebenfalls eine häufige Anforderung. Dafür sind  $munder$  bzw.  $mover$  zuständig, vgl. Listing 7 und (F5). Dort wird im  $munder$ -Teil eine horizontal ausgerichtete geschweifte Klammer verwendet. Meistens lassen sich Sonderzeichen in Codeeditoren über die verfügbaren Schriftarten direkt darstellen. In diesem Fall wurde die alternative Entity-Referenz zum hexadezimalen Unicode-Point  $x23DF$  verwendet. Der im  $mover$ -Block gewählte Pfeil ist dagegen direkt eingefügt.

Listing 8 und (F5) zeigen zusätzlich die Notation eines Grenzwertes mit der durch  $munder$  deklarierten Aussage „Limes  $x$  gegen  $0$ “ mit Verweis auf das Ergebnis „unendlich“, wiederum als direktes Zeichen ersichtlich.

### munderover

Die parallele Setzung von Inhalten unterhalb und oberhalb eines Konstruktes ist eine typische Aufgabe beim Erstellen von Integralen und Summen. Dazu dient das Element  $munderover$ . Den Grundaufbau der in (F6) und (F7) gezeigten Varianten demonstriert Listing 9. Hinzu kommen noch drei Attribute für das  $mo$ -Element zur Gestaltung der Integral- bzw. Summenzeichen:  $stretchy$ ,  $largeop$  und  $moveablelimits$ . Diesen können jeweils die booleschen Werte  $true$  oder  $false$  zugewiesen werden. Mit den beiden erstgenannten Elementen lassen sich Ausdehnung und Größe ändern, während das letztere eine kompaktere seitliche Anordnung der Grenzen für Integral oder Summe ermöglicht.

### Klammern mit mo

Während MathML 3.0 ein separates Element für Klammern namens  $mfenced$  mit den Attributen  $open$  und  $close$  zur Zuweisung der Klammerart bietet, ist MathML Core auf die Nutzung von  $mo$  angewiesen. Die jeweili-

Anzeige

#### Listing 7

```
<munder>
<mrow>
<mi>x</mi>
<mo>+</mo>
<mi>y</mi>
</mrow>
<mo>&#x23DF;
</mo>
</munder>
<mover>
<mi>v</mi>
<mo>→</mo>
</mover>
```

#### Listing 8

```
<munder>
<mo>lim</mo>
<mrow>
<mi>x</mi>
<mo>→</mo>
<mn>0</mn>
</mrow>
</munder>
<mfrac>
<mn>1</mn>
<mi>x</mi>
</mfrac>
<mo>=</mo>
<mi>∞</mi>
```

#### Listing 9

```
<munderover>
<mo>]</mo>
<mi>a</mi>
<mi>b</mi>
</munderover>
<munderover>
<mo>Σ</mo>
<mrow>
<mi>i</mi>
<mo>=</mo>
<mn>1</mn>
</mrow>
<mi>n</mi>
</munderover>
```

#### Listing 10

```
<msup>
<mrow>
<mo></mo>
<mi>a</mi>
<mo>+</mo>
<mi>b</mi>
<mo></mo>
</mrow>
<mn>2</mn>
</msup>
```

ge Klammer wird darin notiert. Für Anfangs- und Endklammer sind somit zwei *mo*-Elemente nötig. Listing 10 beschreibt den oberen Teil von (F8). Dessen unterer Teil ist eine Formel aus dem Bereich Kombinatorik. Da der Code etwas umfangreicher ausfällt, sei auf die bereitgestellte Testseite verwiesen. Interessant ist hier die Nutzung des bereits genannten Attributs *stretchy="true"* für die großen Klammern auf der linken Seite der Gleichung, die eigentlich wie die rechte Seite als Bruch mit *mfrac* erzeugt wurde. Der Bruchstrich ist jedoch infolge des Attributes *linethickness="0"* ausgeblendet.

Da beispielsweise Firefox einen Großteil von MathML 3.0 kennt, wird dieser auch *mfenced* interpretieren. Von daher ist es im Zweifelsfall beim Arbeiten mit Bestandscode ratsam, einen der Chromium-Browser für Tests auf Core-Kompatibilität zu verwenden, um eine möglichst breite Zielgruppe zu erreichen.

### mtable, mtr und mtd

Diese Elementnamen haben eine offensichtliche Konnotation zu den in HTML verfügbaren Elementen für Tabellen und darunter befindlichen Reihen sowie Datenzellen und fungieren als Konstruktoren für Matrizen. Den Code zur Umsetzung der in (F9) aufgebauten Matrix enthält Listing 11. Das Aufspannen der großen, hier rechteckigen Klammern regelt ebenfalls *stretchy="true"*. Alternative Klammern wären je nach Bedarf runde oder auch geschweifte sowie gerade Striche.

### mtext, mspace und ms

Das Element *mtext* erlaubt das Einfügen von zusätzlichen Textfragmenten. In Listing 12 ist eine Art Bedin-

gung aufgeschrieben, die etwa innerhalb eines Beweises oder einer Programmlogik Verwendung finden könnte, vgl. (F10). Zur Vergrößerung von Abständen wird *mspace* verwendet, wovon auch in den meisten der hier verwendeten Formeln Gebrauch gemacht wurde, ohne damit die Listings zu überfrachten.

Erwähnenswert ist an dieser Stelle noch das Element *ms*, das lediglich Stringlitterale kapseln kann. Die Ausgabe des unteren Teils von (F10) kommt durch `<ms>"Hallo Welt!"</ms>` zustande. MathML Core fordert die Anführungszeichen, während diese unter 3.0 ohne Vorgabe erzeugt werden.

### mmultiscripts und mprescripts

Mit den Elementen *mmultiscripts* und darunter *mprescripts* lassen sich gemäß Listing 13 und (F11) oben Objekte erzielen, die ausgehend von einer Basis (hier B genannt) an den beiden Seiten oberhalb und unterhalb Indizes haben können. Zu beachten ist die Reihenfolge im Code.

### mphantom und mpadded

Das Element *mphantom* dient zum Ausblenden von Fragmenten einer Formel, um beispielsweise den Eindruck des unteren Teils von (F11) zu erreichen. Im Zähler des Bruches sind *b* und *+* zwar vorhanden, werden jedoch nicht angezeigt. Das verbleibende *c* steht weiterhin über dem im Nenner ersichtlichen. In diesem Fall ist zusätzlich *mpadded* vorhanden. Damit lassen sich Formelteile blockweise farblich hervorheben und durch die Attribute *lspace*, *voffset*, *width* und *height* positionieren (Listing 14).

### semantics und annotation

Formeln lassen sich generell in *semantics*-Elemente einschließen, was auch zum Teil von Formeleditoren praktiziert wird. Unbedingt notwendig ist das nicht, außer man möchte noch beschreibende Informationen einbinden. Listing 15 untersetzt das mit dem dafür gedachten *annotation*-Element. Dieser Textteil wird im Browser nicht dargestellt, sodass in (F12) nur der Bruch selbst sowie ein ergänzter Kommentar zu sehen sind.

#### Listing 11

```
<mi>M</mi>
<mo>=</mo>
<mrow>
  <mo stretchy="true">[</mo>
  <mtable>
    <mtr>
      <mtd><mn>1</mn></mtd>
      <mtd><mn>2</mn></mtd>
    </mtr>
    <mtr>
      <mtd><mn>3</mn></mtd>
      <mtd><mn>4</mn></mtd>
    </mtr>
  </mtable>
  <mo stretchy="true">]</mo>
</mrow>
```

#### Listing 12

```
<mtext>if</mtext>
<mspace width="0.3em"/>
<mi>a</mi>
<mo>+</mo>
<mn>1</mn>
<mspace width="0.3em"/>
<mtext>then</mtext>
<mspace width="0.3em"/>
<mi>b</mi>
<mo>+</mo>
<mn>2</mn>
```

#### Listing 13

```
<mmultiscripts>
  <mi>B</mi>
  <mn>4</mn>
  <mn>3</mn>
  <mprescripts/>
  <mn>2</mn>
  <mn>1</mn>
</mmultiscripts>
```

#### Listing 14

```
<mpadded
  mathbackground="#FFC" lspace=
    "0.5em" voffset="0.75em"
  width="5em" height="3em">
  <mfrac>
    <mrow>
      <mi>a</mi>
      <mo>+</mo>
    </mrow>
    <mi>b</mi>
    <mo>+</mo>
  </mfrac>
  </mpadded>
  <mi>c</mi>
</mrow>
<mrow>
  <mi>a</mi>
  <mo>+</mo>
  <mi>b</mi>
  <mo>+</mo>
  <mi>c</mi>
</mrow>
</mfrac>
</mpadded>
```

**Anzeige**



zu finden. Es erlaubt über sein *notation*-Attribut bestimmte Umrandungen und Durchstreichungen, interessant z. B. für das Kürzen von Brüchen oder ähnliche Anforderungen. In Listing 18 steht ein Ausschnitt des Kürzens von  $3!$  (Fakultät von 3, also  $3 * 2 * 1$ ) in der unteren Formeldarstellung von (F13). Wird *notation* erkannt, etwa im Firefox, dann erscheint eine diagonale Linie von links unten nach rechts oben. Ein Chromium-Browser würde gar keinen Strich erzeugen. Durch die zusätzliche *style*-Angabe wird dann immerhin horizontal durchgestrichen.

Die restlichen Demonstrationen lassen sich im umfangreicheren Code auf der Testseite im Browser nachvollziehen. Das Zusammenspiel von Inline- und Blockformatierung veranschaulicht (F14). Die Inline-Formel befindet sich im einführenden Textabsatz, während die eigentliche Hauptformel darunter geblockt ist. Das Arithmetische Mittel (F15) und die Integralrechnung (F16) sind weitere typische Formeln, die sich unter Nutzung der hier beschriebenen Basistechniken erzeugen lassen.

### Formelwerkzeuge

Formelcode lässt sich in dafür ausgestatteten Editoren wie u. a. dem kommerziellen `<oxygen>` XML Editor [9] und dem frei verfügbaren Visual Studio Code [10] mit MathML-Erweiterung [11] komfortabel schreiben. MathType ist ein spezialisierter kommerzieller Formeleditor, der in einer für den MathML-Export nutzbaren Demofassung verfügbar ist [12]. Im ingenieurwissenschaftlichen Bereich ist die mächtige Software MATLAB bekannt [13]. Damit lassen sich ebenfalls Formelausdrücke nach MathML konvertieren. Die bekannte Wolfram-Plattform bündelt zum Thema einige nützliche Onlinetools unter [14].

Formeleditoren populärer Textverarbeitungen wie Microsoft Word sowie OpenOffice/LibreOffice ermöglichen ebenfalls den Codeexport von grafisch erzeugten Formeln. **Abbildung 4** veranschaulicht die zum Erstellen und Exportieren einer Formel mit Word relevanten Stellen.

Listing 19 enthält den aus Word über die Zwischenablage exportierten Code des Bruches  $x/y$ . Auffällig sind die eigentlich unerwünschten *mml*:-Namespace-Präfixe bei allen Elementen. Diese lassen sich durch einfaches Suchen und Ersetzen mit einem Leerstring eliminieren. Anschließend wird noch *xmlns:mml* zu *xmlns* verkürzt und der zum Office-Universum gehörende Namensraum *xmlns:m* entfernt. Im Nachgang könnte hier, wie eingangs diskutiert, auch *mrow* entfallen.

### Fazit und Ausblick

MathML gewinnt durch die verbesserte Unterstützung in den gebräuchlichen Browsern an Attraktivität für den Einsatz im Kontext von HTML5. Eine direkte Einbindung in Websites verbessert gegenüber Rastergrafiken mit Formelinhalt Usability und Accessibility. Dank brauchbarer Unterstützung in Codeeditoren und grafi-

scher Formelwerkzeuge wird die praktische Nutzung erleichtert. Der Spezifikationsprozess wurde nach Jahren der Stagnation neu belebt und so sind künftig zusätzliche Erweiterungen und Verbesserungen zu erwarten.

Von den zahlreichen Referenzen sei abschließend das im Bereich Webentwicklung populäre MDN mit Fokus auf MathML Core zum vertieften Lernen und Nachschlagen empfohlen [15].



**Dr. Thomas Meinike** hat ein naturwissenschaftliches Studium absolviert und ist seit 1997 an der Hochschule Merseburg als Lehrkraft tätig. Arbeitsschwerpunkte sind XML-Anwendungen in der Technischen Kommunikation, Onlinehilfen und Webentwicklung. Er publiziert Fachartikel und hält Vorträge zu Themen im Umfeld von XML und Web.

✉ [thomas.meinike@hs-merseburg.de](mailto:thomas.meinike@hs-merseburg.de)

### Links & Literatur

- [1] <https://twitter.com/mathml3/status/1612881623510388738>
- [2] W3C: Mathematical Markup Language 1.0 Specification: <https://www.w3.org/TR/1998/REC-MathML-19980407/>
- [3] W3C: Mathematical Markup Language Version 3.0 2nd Edition: <https://www.w3.org/TR/MathML/>
- [4] Can I use: MathML: <https://caniuse.com/?search=MathML>
- [5] MathJax: Beautiful and accessible math in all browsers: <https://www.mathjax.org/>
- [6] W3C: MathML Core Candidate Recommendation: <https://w3c.github.io/mathml-core/>
- [7] W3C: Mathematical Markup Language Version 4.0: <https://w3c.github.io/mathml/>
- [8] Meinike, T.: MathML-Grundlagen: [https://datenverdrahten.de/mathml/MathML\\_GL.html](https://datenverdrahten.de/mathml/MathML_GL.html)
- [9] `<oxygen>` XML Editor: <https://www.oxygenxml.com>
- [10] Visual Studio Code: <https://code.visualstudio.com>
- [11] MathML extension for VS Code: <https://github.com/michijs/vscode-mathml/>
- [12] MathType: Wiris: <https://demo.wiris.com/mathtype/en/developers.php#mathml-latex>
- [13] MathWorks: MATLAB: <https://de.mathworks.com/products/matlab.html>
- [14] Wolfram Research, Inc.: MathML Central: <https://www.mathmlcentral.com/>
- [15] Mozilla: MDN Web Docs: <https://developer.mozilla.org/en-US/docs/Web/MathML>