

# DRAFT INTERNATIONAL STANDARD

## ISO/IEC DIS 40314

ISO/IEC JTC 1

Secretariat: ANSI

Voting begins on:  
2014-10-01

Voting terminates on:  
2015-01-01

---

---

## Information technology — Mathematical Markup Language (MathML) Version 3.0 2nd Edition

*Titre manque*

ICS: 35.240.30

**iTeh STANDARD PREVIEW**  
(standards.iteh.ai)  
Full standard:  
<https://standards.iteh.ai/catalog/standards/sist/c37ba4c6-1244-4718-adbf-84edf5b10a7d/iso-iec-40314-2016>

THIS DOCUMENT IS A DRAFT CIRCULATED FOR COMMENT AND APPROVAL. IT IS THEREFORE SUBJECT TO CHANGE AND MAY NOT BE REFERRED TO AS AN INTERNATIONAL STANDARD UNTIL PUBLISHED AS SUCH.

IN ADDITION TO THEIR EVALUATION AS BEING ACCEPTABLE FOR INDUSTRIAL, TECHNOLOGICAL, COMMERCIAL AND USER PURPOSES, DRAFT INTERNATIONAL STANDARDS MAY ON OCCASION HAVE TO BE CONSIDERED IN THE LIGHT OF THEIR POTENTIAL TO BECOME STANDARDS TO WHICH REFERENCE MAY BE MADE IN NATIONAL REGULATIONS.

RECIPIENTS OF THIS DRAFT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.



Reference number  
ISO/IEC DIS 40314:2014(E)

© ISO/IEC 2014

**iTeh STANDARD PREVIEW**  
(standards.itih.ai)  
Full standard:  
<https://standards.itih.ai/catalog/standards/sist/e37ba4c6-1244-4718-adbf-84edf5b10a7d/iso-iec-40314-2016>

### Copyright notice

This ISO document is a Draft International Standard and is copyright-protected by ISO. Except as permitted under the applicable laws of the user's country, neither this ISO draft nor any extract from it may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, photocopying, recording or otherwise, without prior written permission being secured.

Requests for permission to reproduce should be addressed to either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
Case postale 56 • CH-1211 Geneva 20  
Tel. + 41 22 749 01 11  
Fax + 41 22 749 09 47  
E-mail [copyright@iso.org](mailto:copyright@iso.org)  
Web [www.iso.org](http://www.iso.org)

Reproduction may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

# Contents

<b>1</b>	<b>Introduction</b>	<b>9</b>
1.1	Mathematics and its Notation . . . . .	9
1.2	Origins and Goals . . . . .	10
1.2.1	Design Goals of MathML . . . . .	10
1.3	Overview . . . . .	11
1.4	A First Example . . . . .	11
<b>2</b>	<b>MathML Fundamentals</b>	<b>14</b>
2.1	MathML Syntax and Grammar . . . . .	14
2.1.1	General Considerations . . . . .	14
2.1.2	MathML and Namespaces . . . . .	14
2.1.3	Children versus Arguments . . . . .	15
2.1.4	MathML and Rendering . . . . .	15
2.1.5	MathML Attribute Values . . . . .	15
2.1.6	Attributes Shared by all MathML Elements . . . . .	20
2.1.7	Collapsing Whitespace in Input . . . . .	21
2.2	The Top-Level <math> Element . . . . .	22
2.2.1	Attributes . . . . .	22
2.2.2	Deprecated Attributes . . . . .	24
2.3	Conformance . . . . .	24
2.3.1	MathML Conformance . . . . .	24
2.3.2	Handling of Errors . . . . .	27
2.3.3	Attributes for unspecified data . . . . .	27
<b>3</b>	<b>Presentation Markup</b>	<b>28</b>
3.1	Introduction . . . . .	28
3.1.1	What Presentation Elements Represent . . . . .	28
3.1.2	Terminology Used In This Chapter . . . . .	29
3.1.3	Required Arguments . . . . .	30
3.1.4	Elements with Special Behaviors . . . . .	31
3.1.5	Directionality . . . . .	32
3.1.6	Displaystyle and Scriptlevel . . . . .	33
3.1.7	Linebreaking of Expressions . . . . .	34
3.1.8	Warning about fine-tuning of presentation . . . . .	35
3.1.9	Summary of Presentation Elements . . . . .	37
3.1.10	Mathematics style attributes common to presentation elements . . . . .	38
3.2	Token Elements . . . . .	38
3.2.1	Token Element Content Characters, <mglyph/> . . . . .	39
3.2.2	Mathematics style attributes common to token elements . . . . .	41
3.2.3	Identifier <mi> . . . . .	45

3.2.4	Number <code>&lt;mn&gt;</code> . . . . .	46
3.2.5	Operator, Fence, Separator or Accent <code>&lt;mo&gt;</code> . . . . .	47
3.2.6	Text <code>&lt;mtext&gt;</code> . . . . .	60
3.2.7	Space <code>&lt;mspace/&gt;</code> . . . . .	62
3.2.8	String Literal <code>&lt;ms&gt;</code> . . . . .	64
3.3	General Layout Schemata . . . . .	64
3.3.1	Horizontally Group Sub-Expressions <code>&lt;mrow&gt;</code> . . . . .	64
3.3.2	Fractions <code>&lt;mfrac&gt;</code> . . . . .	67
3.3.3	Radicals <code>&lt;msqrt&gt;</code> , <code>&lt;mroot&gt;</code> . . . . .	69
3.3.4	Style Change <code>&lt;mstyle&gt;</code> . . . . .	69
3.3.5	Error Message <code>&lt;merror&gt;</code> . . . . .	72
3.3.6	Adjust Space Around Content <code>&lt;mpadded&gt;</code> . . . . .	73
3.3.7	Making Sub-Expressions Invisible <code>&lt;mphantom&gt;</code> . . . . .	78
3.3.8	Expression Inside Pair of Fences <code>&lt;mfenced&gt;</code> . . . . .	80
3.3.9	Enclose Expression Inside Notation <code>&lt;mathclose&gt;</code> . . . . .	83
3.4	Script and Limit Schemata . . . . .	85
3.4.1	Subscript <code>&lt;msub&gt;</code> . . . . .	86
3.4.2	Superscript <code>&lt;msup&gt;</code> . . . . .	87
3.4.3	Subscript-superscript Pair <code>&lt;msubsup&gt;</code> . . . . .	87
3.4.4	Underscript <code>&lt;munder&gt;</code> . . . . .	88
3.4.5	Overscript <code>&lt;mover&gt;</code> . . . . .	89
3.4.6	Underscript-overscript Pair <code>&lt;munderover&gt;</code> . . . . .	91
3.4.7	Prescripts and Tensor Indices <code>&lt;multiscripts&gt;</code> , <code>&lt;mprescripts/&gt;</code> , <code>&lt;none/&gt;</code> . . . . .	93
3.5	Tabular Math . . . . .	95
3.5.1	Table or Matrix <code>&lt;mtable&gt;</code> . . . . .	95
3.5.2	Row in Table or Matrix <code>&lt;mtr&gt;</code> . . . . .	99
3.5.3	Labeled Row in Table or Matrix <code>&lt;mlabeledtr&gt;</code> . . . . .	99
3.5.4	Entry in Table or Matrix <code>&lt;mtd&gt;</code> . . . . .	101
3.5.5	Alignment Markers <code>&lt;maligngroup/&gt;</code> , <code>&lt;malignmark/&gt;</code> . . . . .	101
3.6	Elementary Math . . . . .	110
3.6.1	Stacks of Characters <code>&lt;mstack&gt;</code> . . . . .	111
3.6.2	Long Division <code>&lt;mlongdiv&gt;</code> . . . . .	113
3.6.3	Group Rows with Similar Positions <code>&lt;msgroup&gt;</code> . . . . .	114
3.6.4	Rows in Elementary Math <code>&lt;msrow&gt;</code> . . . . .	115
3.6.5	Carries, Borrows, and Crossouts <code>&lt;mscarries&gt;</code> . . . . .	115
3.6.6	A Single Carry <code>&lt;mscarry&gt;</code> . . . . .	116
3.6.7	Horizontal Line <code>&lt;msline/&gt;</code> . . . . .	117
3.6.8	Elementary Math Examples . . . . .	118
3.7	Enlivening Expressions . . . . .	124
3.7.1	Bind Action to Sub-Expression <code>&lt;maction&gt;</code> . . . . .	124
3.8	Semantics and Presentation . . . . .	126
<b>4</b>	<b>Content Markup</b> . . . . .	<b>127</b>
4.1	Introduction . . . . .	127
4.1.1	The Intent of Content Markup . . . . .	127
4.1.2	The Structure and Scope of Content MathML Expressions . . . . .	128
4.1.3	Strict Content MathML . . . . .	128
4.1.4	Content Dictionaries . . . . .	129
4.1.5	Content MathML Concepts . . . . .	130
4.2	Content MathML Elements Encoding Expression Structure . . . . .	131

4.2.1	Numbers <cn>	132
4.2.2	Content Identifiers <ci>	138
4.2.3	Content Symbols <csymbol>	140
4.2.4	String Literals <cs>	142
4.2.5	Function Application <apply>	143
4.2.6	Bindings and Bound Variables <bind> and <bvar>	146
4.2.7	Structure Sharing <share>	148
4.2.8	Attribution via semantics	150
4.2.9	Error Markup <cerror>	151
4.2.10	Encoded Bytes <cbytes>	152
4.3	Content MathML for Specific Structures	152
4.3.1	Container Markup	153
4.3.2	Bindings with <apply>	154
4.3.3	Qualifiers	156
4.3.4	Operator Classes	162
4.3.5	Non-strict Attributes	169
4.4	Content MathML for Specific Operators and Constants	170
4.4.1	Functions and Inverses	170
4.4.2	Arithmetic, Algebra and Logic	180
4.4.3	Relations	200
4.4.4	Calculus and Vector Calculus	205
4.4.5	Theory of Sets	224
4.4.6	Sequences and Series	233
4.4.7	Elementary classical functions	243
4.4.8	Statistics	247
4.4.9	Linear Algebra	253
4.4.10	Constant and Symbol Elements	260
4.5	Deprecated Content Elements	268
4.5.1	Declare <declare>	268
4.5.2	Relation <reln>	268
4.5.3	Relation <fn>	268
4.6	The Strict Content MathML Transformation	268
<b>5</b>	<b>Mixing Markup Languages for Mathematical Expressions</b>	<b>272</b>
5.1	Annotation Framework	272
5.1.1	Annotation elements	272
5.1.2	Annotation keys	273
5.1.3	Alternate representations	274
5.1.4	Content equivalents	275
5.1.5	Annotation references	276
5.2	Elements for Semantic Annotations	276
5.2.1	The <semantics> element	276
5.2.2	The <annotation> element	277
5.2.3	The <annotation-xml> element	278
5.3	Combining Presentation and Content Markup	281
5.3.1	Presentation Markup in Content Markup	281
5.3.2	Content Markup in Presentation Markup	282
5.4	Parallel Markup	282
5.4.1	Top-level Parallel Markup	282
5.4.2	Parallel Markup via Cross-References	283

<b>6</b>	<b>Interactions with the Host Environment</b>	<b>286</b>
6.1	Introduction . . . . .	286
6.2	Invoking MathML Processors . . . . .	286
6.2.1	Recognizing MathML in XML . . . . .	286
6.2.2	Recognizing MathML in HTML . . . . .	287
6.2.3	Resource Types for MathML Documents . . . . .	287
6.2.4	Names of MathML Encodings . . . . .	287
6.3	Transferring MathML . . . . .	288
6.3.1	Basic Transfer Flavor Names and Contents . . . . .	288
6.3.2	Recommended Behaviors when Transferring . . . . .	289
6.3.3	Discussion . . . . .	289
6.3.4	Examples . . . . .	290
6.4	Combining MathML and Other Formats . . . . .	292
6.4.1	Mixing MathML and XHTML . . . . .	294
6.4.2	Mixing MathML and non-XML contexts . . . . .	294
6.4.3	Mixing MathML and HTML . . . . .	294
6.4.4	Linking . . . . .	295
6.4.5	MathML and Graphical Markup . . . . .	296
6.5	Using CSS with MathML . . . . .	297
6.5.1	Order of processing attributes versus style sheets . . . . .	298
<b>7</b>	<b>Characters, Entities and Fonts</b>	<b>299</b>
7.1	Introduction . . . . .	299
7.2	Unicode Character Data . . . . .	299
7.3	Entity Declarations . . . . .	300
7.4	Special Characters Not in Unicode . . . . .	300
7.5	Mathematical Alphanumeric Symbols . . . . .	300
7.6	Non-Marking Characters . . . . .	303
7.7	Anomalous Mathematical Characters . . . . .	303
7.7.1	Keyboard Characters . . . . .	303
7.7.2	Pseudo-scripts . . . . .	304
7.7.3	Combining Characters . . . . .	306
<b>A</b>	<b>Parsing MathML</b>	<b>308</b>
A.1	Use of MathML as Well-Formed XML . . . . .	308
A.2	Using the RelaxNG Schema for MathML3 . . . . .	308
A.2.1	Full MathML . . . . .	309
A.2.2	Elements Common to Presentation and Content MathML . . . . .	309
A.2.3	The Grammar for Presentation MathML . . . . .	311
A.2.4	The Grammar for Strict Content MathML3 . . . . .	323
A.2.5	The Grammar for Content MathML . . . . .	325
A.2.6	MathML as a module in a RelaxNG Schema . . . . .	332
A.3	Using the MathML DTD . . . . .	333
A.3.1	Document Validation Issues . . . . .	333
A.3.2	Attribute values in the MathML DTD . . . . .	333
A.3.3	DOCTYPE declaration for MathML . . . . .	334
A.4	Using the MathML XML Schema . . . . .	334
A.4.1	Associating the MathML schema with MathML fragments . . . . .	334
A.5	Parsing MathML in XHTML . . . . .	334
A.6	Parsing MathML in HTML . . . . .	334

<b>B</b>	<b>Media Types Registrations</b>	<b>335</b>
B.1	Selection of Media Types for MathML Instances . . . . .	335
B.2	Media type for Generic MathML . . . . .	336
B.3	Media type for Presentation MathML . . . . .	337
B.4	Media type for Content MathML . . . . .	338
<b>C</b>	<b>Operator Dictionary (Non-Normative)</b>	<b>340</b>
C.1	Indexing of the operator dictionary . . . . .	340
C.2	Format of operator dictionary entries . . . . .	340
C.3	Notes on lspace and rspace attributes . . . . .	341
C.4	Operator dictionary entries . . . . .	341
<b>D</b>	<b>Glossary (Non-Normative)</b>	<b>379</b>
<b>E</b>	<b>Working Group Membership and Acknowledgments (Non-Normative)</b>	<b>383</b>
E.1	The Math Working Group Membership . . . . .	383
E.2	Acknowledgments . . . . .	386
<b>F</b>	<b>Changes (Non-Normative)</b>	<b>387</b>
F.1	Changes between MathML 3.0 First Edition and Second Edition . . . . .	387
F.2	Changes between MathML 2.0 Second Edition and MathML 3.0 . . . . .	390
<b>G</b>	<b>Normative References</b>	<b>391</b>
<b>H</b>	<b>References (Non-Normative)</b>	<b>393</b>
<b>I</b>	<b>Index (Non-Normative)</b>	<b>395</b>
I.1	MathML Elements . . . . .	395
I.2	MathML Attributes . . . . .	400

iTeH STANDARD PREVIEW  
 (standards.iteh.ai)  
 Full standard:  
<https://standards.iteh.ai/catalog/standards/sist/3770a4c6-1244-4718-adbf-84edf5b10a7d/iso-iec-40314-2016>

# Chapter 1

## Introduction

### 1.1 Mathematics and its Notation

A distinguishing feature of mathematics is the use of a complex and highly evolved system of two-dimensional symbolic notation. As J. R. Pierce writes in his book on communication theory, mathematics and its notation should not be viewed as one and the same thing [Pierce1961]. Mathematical ideas can exist independently of the notation that represents them. However, the relation between meaning and notation is subtle, and part of the power of mathematics to describe and analyze derives from its ability to represent and manipulate ideas in symbolic form. The challenge before a Mathematical Markup Language (MathML) in enabling mathematics on the World Wide Web is to capture both notation and content (that is, its meaning) in such a way that documents can utilize the highly evolved notation of written and printed mathematics as well as the new potential for interconnectivity in electronic media.

Mathematical notation evolves constantly as people continue to innovate in ways of approaching and expressing ideas. Even the common notation of arithmetic has gone through an amazing variety of styles, including many defunct ones advocated by leading mathematical figures of their day [Cajori1928]. Modern mathematical notation is the product of centuries of refinement, and the notational conventions for high-quality typesetting are quite complicated and subtle. For example, variables and letters which stand for numbers are usually typeset today in a special mathematical italic font subtly distinct from the usual text italic; this seems to have been introduced in Europe in the late sixteenth century. Spacing around symbols for operations such as  $+$ ,  $-$ ,  $\times$  and  $/$  is slightly different from that of text, to reflect conventions about operator precedence that have evolved over centuries. Entire books have been devoted to the conventions of mathematical typesetting, from the alignment of superscripts and subscripts, to rules for choosing parenthesis sizes, and on to specialized notational practices for subfields of mathematics. The manuals describing the nuances of present-day computer typesetting and composition systems can run to hundreds of pages.

Notational conventions in mathematics, and in printed text in general, guide the eye and make printed expressions much easier to read and understand. Though we usually take them for granted, we, as modern readers, rely on numerous conventions such as paragraphs, capital letters, font families and cases, and even the device of decimal-like numbering of sections such as is used in this document. Such notational conventions are perhaps even more important for electronic media, where one must contend with the difficulties of on-screen reading. Appropriate standards coupled with computers enable a broadening of access to mathematics beyond the world of print. The markup methods for mathematics in use just before the Web rose to prominence importantly included  $\text{T}_\text{E}\text{X}$  (also written  $\text{T}\text{e}\text{X}$ ) [Knuth1986] and approaches based on SGML ([AAP-math], [Poppelier1992] and [ISO-12083]).

It is remarkable how widespread the current conventions of mathematical notation have become. The general two-dimensional layout, and most of the same symbols, are used in all modern mathematical communications, whether the participants are, say, European, writing left-to-right, or Middle-Eastern,



writing right-to-left. Of course, conventions for the symbols used, particularly those naming functions and variables, may tend to favor a local language and script. The largest variation from the most common is a form used in some Arabic-speaking communities which lays out the entire mathematical notation from right-to-left, roughly in mirror image of the European tradition.

However, there is more to putting mathematics on the Web than merely finding ways of displaying traditional mathematical notation in a Web browser. The Web represents a fundamental change in the underlying metaphor for knowledge storage, a change in which *interconnection* plays a central role. It has become important to find ways of communicating mathematics which facilitate automatic processing, searching and indexing, and reuse in other mathematical applications and contexts. With this advance in communication technology, there is an opportunity to expand our ability to represent, encode, and ultimately to communicate our mathematical insights and understanding with each other. We believe that MathML as specified below is an important step in developing mathematics on the Web.

## 1.2 Origins and Goals

### 1.2.1 Design Goals of MathML

MathML has been designed from the beginning with the following ultimate goals in mind.

MathML should ideally:

- Encode mathematical material suitable for all educational and scientific communication.
- Encode both mathematical notation and mathematical meaning.
- Facilitate conversion to and from other mathematical formats, both presentational and semantic. Output formats should include:
  - graphical displays
  - speech synthesizers
  - input for computer algebra systems
  - other mathematics typesetting languages, such as  $\text{\TeX}$
  - plain text displays, e.g. VT100 emulators
  - international print media, including braille

It is recognized that conversion to and from other notational systems or media may entail loss of information in the process.

- Allow the passing of information intended for specific renderers and applications.
- Support efficient browsing of lengthy expressions.
- Provide for extensibility.
- Be well suited to templates and other common techniques for editing formulas.
- Be legible to humans, and simple for software to generate and process.

No matter how successfully MathML achieves its goals as a markup language, it is clear that MathML is useful only if it is implemented well. The W3C Math Working Group has identified a short list of additional implementation goals. These goals attempt to describe concisely the minimal functionality MathML rendering and processing software should try to provide.

- MathML expressions in HTML (and XHTML) pages should render properly in popular Web browsers, in accordance with reader and author viewing preferences, and at the highest quality possible given the capabilities of the platform.
- HTML (and XHTML) documents containing MathML expressions should print properly and at high-quality printer resolutions.
- MathML expressions in Web pages should be able to react to user gestures, such those as with a mouse, and to coordinate communication with other applications through the browser.

- Mathematical expression editors and converters should be developed to facilitate the creation of Web pages containing MathML expressions.

The extent to which these goals are ultimately met depends on the cooperation and support of browser vendors and other developers. The W3C Math Working Group has continued to work with other working groups of the W3C, and outside the W3C, to ensure that the needs of the scientific community will be met. MathML 2 and its implementations showed considerable progress in this area over the situation that obtained at the time of the MathML 1.0 Recommendation (April 1998) [MathML1]. MathML3 and the developing Web are expected to allow much more.

### 1.3 Overview

MathML is a markup language for describing mathematics. It is usually expressed in XML syntax, although HTML and other syntaxes are possible. A special aspect of MathML is that there are two main strains of markup: Presentation markup, discussed in Chapter 3, is used to display mathematical expressions; and Content markup, discussed in Chapter 4, is used to convey mathematical meaning. Content markup is specified in particular detail. This specification makes use of an XML format called Content Dictionaries. This format has been developed by the OpenMath Society, [OpenMath2004] with the dictionaries being used by this specification involving joint development by the OpenMath Society and the W3C Math Working Group.

Fundamentals common to both strains of markup are covered in Chapter 2, while the means for combining these strains, as well as external markup, into single MathML objects are discussed in Chapter 5. How MathML interacts with applications is covered in Chapter 6. Finally, a discussion of special symbols, and issues regarding characters, entities and fonts, is given in Chapter 7.

### 1.4 A First Example

The quadratic formula provides a simple but instructive illustration of MathML markup.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

MathML offers two flavors of markup of this formula. The first is the style which emphasizes the actual presentation of a formula, the two-dimensional layout in which the symbols are arranged. An example of this type is given just below. The second flavor emphasizes the mathematical content and an example of it follows the first one.

```
<mrow>
  <mi>x</mi>
  <mo>=</mo>
  <mfrac>
    <mrow>
      <mrow>
        <mo>-</mo>
        <mi>b</mi>
      </mrow>
      <mo>&PlusMinus;</mo>
      <msqrt>
```

```

<mrow>
  <msup>
    <mi>b</mi>
    <mn>2</mn>
  </msup>
  <mo>-</mo>
  <mrow>
    <mn>4</mn>
    <mo>&InvisibleTimes;</mo>
    <mi>a</mi>
    <mo>&InvisibleTimes;</mo>
    <mi>c</mi>
  </mrow>
</mrow>
</msqrt>
</mrow>
<mrow>
  <mn>2</mn>
  <mo>&InvisibleTimes;</mo>
  <mi>a</mi>
</mrow>
</mfrac>
</mrow>

```

Consider the superscript 2 in this formula. It represents the squaring operation here, but the meaning of a superscript in other situations depends on the context. A letter with a superscript can be used to signify a particular component of a vector, or maybe the superscript just labels a different type of some structure. Similarly two letters written one just after the other could signify two variables multiplied together, as they do in the quadratic formula, or they could be two letters making up the name of a single variable. What is called Content Markup in MathML allows closer specification of the mathematical meaning of many common formulas. The quadratic formula given in this style of markup is as follows.

```

<apply>
  <eq/>
  <ci>x</ci>
  <apply>
    <divide/>
    <apply>
      <plus/>
      <apply>
        <minus/>
        <ci>b</ci>
      </apply>
    </apply>
    <apply>
      <root/>
      <apply>
        <minus/>
        <apply>
          <power/>
          <ci>b</ci>
          <cn>2</cn>

```

```
</apply>
<apply>
  <times/>
  <cn>4</cn>
  <ci>a</ci>
  <ci>c</ci>
</apply>
</apply>
</apply>
<apply>
  <times/>
  <cn>2</cn>
  <ci>a</ci>
</apply>
</apply>
</apply>
```

**iTeh STANDARD PREVIEW**  
(standards.iteh.ai)

Full standard:  
<https://standards.iteh.ai/catalog/standards/sist/c37ba4c6-244-4718-adbf-84edf5b10a7d/iso-iec-40314-2016>

## Chapter 2

### MathML Fundamentals

#### 2.1 MathML Syntax and Grammar

##### 2.1.1 General Considerations

The basic ‘syntax’ of MathML is defined using XML syntax, but other syntaxes that can encode labeled trees are possible. Notably the HTML parser may also be used with MathML. Upon this, we layer a ‘grammar’, being the rules for allowed elements, the order in which they can appear, and how they may be contained within each other, as well as additional syntactic rules for the values of attributes. These rules are defined by this specification, and formalized by a RelaxNG schema [RELAX-NG]. The RelaxNG Schema is normative, but a DTD (Document Type Definition) and an XML Schema [XMLSchemas] are provided for continuity (they were normative for MathML2). See Appendix A.

MathML’s character set consists of legal characters as specified by Unicode [Unicode], further restricted by the characters not allowed in XML. The use of Unicode characters for mathematics is discussed in Chapter 7.

The following sections discuss the general aspects of the MathML grammar as well as describe the syntaxes used for attribute values.

##### 2.1.2 MathML and Namespaces

An XML namespace [Namespaces] is a collection of names identified by a URI. The URI for the MathML namespace is:

```
http://www.w3.org/1998/Math/MathML
```

To declare a namespace when using the XML serialisation of MathML, one uses an `xmlns` attribute, or an attribute with an `xmlns` prefix. When the `xmlns` attribute is used alone, it sets the default namespace for the element on which it appears, and for any child elements. For example:

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
<mrow>...</mrow>
</math>
```

When the `xmlns` attribute is used as a prefix, it declares a prefix which can then be used to explicitly associate other elements and attributes with a particular namespace. When embedding MathML within XHTML, one might use:

```
<body xmlns:m="http://www.w3.org/1998/Math/MathML">
...
<m:math><m:mrow>...</m:mrow></m:math>
...
</body>
```