MathML isn’t that scary!

# What is MathML?

MathML is **Mathematical Markup Language**. It is a language of the [XML](https://www.w3schools.com/xml/default.asp) (https://bit.ly/xmlbasics) family of markup languages designed for describe mathematical expressions and scientific notations (See [MDN reference pages for MathML](https://developer.mozilla.org/en-US/docs/Web/MathML) (https://bit.ly/mathmlintro)).

MathML – like it’s cousin HTML – must be **rendered** to be properly viewed. Many browsers use a library called [MathJax](https://www.mathjax.org/) (https://bit.ly/mathjaxintro) to render MathML. MathJax is written in JavaScript and runs in all major browsers.

Many people who typeset scientific work are familiar with LATEX but fewer have much – if any – experience of MathML. I will use a simple expression to show what MathML looks like and how it compares with LATEX notation.

Afterwards I will discuss creating MathML from LATEX or [markdown](https://en.wikipedia.org/wiki/Markdown) (https://bit.ly/markdowndef) without coding MathML yourself.

# An Example

## Microsoft Word equation

Here is a formula constructed in the Word equation editor:

## LATEX notation

Here is the same formula in LATEX notation:

$$\sum{\frac{(x - \bar{x})^2} {n}}$$

Many of us are quite comfortable with this, and think it is a relatively simple system for typesetting mathematics.

It isn’t! We all took a long time to learn this stuff and from time to time we have all been held up debugging our maths code.

## MathML

And here is the same formula in MathML (also presented in Appendix 1 Annotated MathML with a few annotations to explain things):

<math display="block">

<mo>∑</mo>

<mfrac>

<mrow>

<mo stretchy="false">(</mo>

<msub>

<mi>x</mi>

<mi>i</mi>

</msub>

<mo>−</mo>

<mover accent="true">

<mi>x</mi>

<mo>ˉ</mo>

</mover>

<msup>

<mo stretchy="false">)</mo>

<mn>2</mn>

</msup>

</mrow>

<mi>n</mi>

</mfrac>

</math>

We may not be so familiar with this and may not read it fluently at all.

# Is MathML difficult?

MathML is just another markup language. If you are a computer scientist you may care to spend a drunken evening debating whether or not if LATEX and MathML can both be used to describe mathematical expressions, they must in someway be of similar complexity: go on, I won’t tell.

I am still much less used to writing MathML than LATEX, but the more I read MathML expressions, the more I understand and the easier it becomes.

## Why does it look difficult?

Compared to LATEX, MathML does look rather long winded. This is a feature of the use of **tags** for scoping commands in the XML family of languages (which don’t allow the abbreviation – such as omitting </p> - that SGML allows). Every tag has an end tag. On the other hand, the **scope** of tags in XML is much easier to grasp than counting {‘s in LATEX. If LATEX is easier to debug than a Microsoft Word equation, then honestly MathML is easier to debug than LATEX (if it is written reasonably well).

# Is there any advantage to using MathML?

Most modern screen reader software can read MathML in a coherent, reasonably natural way.

# How can I produce MathML?

The best output for MathML is an html document. If you are looking for detail on using RMarkdown or LATEX you can find links at [Making equations accessible at the University of York (v2.7)](https://bit.ly/Equations4All) (https://bit.ly/Equations4All).

You can produce the document in several ways.

### From RStudio by creating an rmarkdown file. The yaml header should look like: ---

title: "Testing MathML"  
author: "Jim Tyson"  
date: "`r Sys.Date()`"  
output:  
 html\_document:  
 pandoc\_args: "--mathml"  
 mathjax: NULL

### --- From LATEX

You can convert a LATEX file with mathematics to html using [Pandoc](https://pandoc.org/demos.html#examples):

pandoc -s [math.tex](https://pandoc.org/demo/math.tex) -o output.html

### From standard markdown

You can convert a .md file with mathematics to html using Pandoc:

pandoc -s --mathml math.md -o math.html

# Appendix 1 Annotated MathML

|  |  |  |
| --- | --- | --- |
| Comment | MathML Code | Comment |
| <math></math> maths begins and ends.  <mfrac></mfrac> surrounds a fraction element.  <mo stretchy="false|true"> determines whether an operator resizes to match adjacent elements.  <mover accent=”false|true”> specifies that the decoration over the following element should be closely bound – like an accent in writing. | <math display="block">  <mo>∑</mo>  <mfrac>  <mrow>  <mo stretchy="false">(</mo>  <msub>  <mi>x</mi>  <mi>i</mi>  </msub>  <mo>−</mo>  <mover accent="true">  <mi>x</mi>  <mo>ˉ</mo>  </mover>  <msup>  <mo stretchy="false">)</mo>  <mn>2</mn>  </msup>  </mrow>  <mi>n</mi>  </mfrac>  </math> | <mo></mo> introduces an operator element, such as + or here a summation sign.  <mrow></mrow> groups together several elements into one.\*  <msub></msub> defines a subscript expression. It contains a single base element and a single index or subscript in that sequence.  <msup></sup> defines a superscript environment.  \*This is important since some elements can have exactly two children and so some must be grouped. |