

# Authoring Web-accessible STEM documents

Volker Sorge

School of Computer Science  
University of Birmingham  
[cs.bham.ac.uk](http://cs.bham.ac.uk)



Progressive Accessibility  
Solutions  
Birmingham, UK  
[progressiveaccess.com](http://progressiveaccess.com)



MathJax  
Consortium  
[mathjax.org](http://mathjax.org)



Empower 2018, Delhi, 26 October 2018

Read along at:

Play along at:

- ① Generating Math documents for the Web
- ② Using accessible Maths on the Web
- ③ Using accessible diagrams on the Web
- ④ Generating accessible diagrams for the Web

## Part 1:

## Generating Math documents for the Web (with MathJax)

# What is MathJax?

- MathJax is a JavaScript library for rendering Mathematics in all browsers
- Can take  $\text{\LaTeX}$ , AsciiMath, and MathML as input
- Generates browser output, e.g. HTML/CSS, SVG
- Standard Maths rendering solution for: stackexchange, wordpress blogs, mediawiki, etc.

MathJax is the de facto rendering solution of (nearly) all Mathematics on the web (in 2016: 35 million unique daily rendering requests via CDN)

<http://www.mathjax.org>

# Getting your Document to the Web

- Common sources for Maths document:  
 $\text{\LaTeX}$ , Word, Markdown
- Translate into HTML format for the web
- There are a number of tools, many source dependent
- One flexible multi-format transformer is pandoc at <http://pandoc.org>

# Getting started with Pandoc

- Web demo at <http://pandoc.org/try/>
- Simple example usage:

```
pandoc --self-contained -s example.docx -t html -o example.html
```

- Complex example:

```
pandoc -s -S --toc -c pandoc.css -A footer.html example.docx
```

- See for more details and examples also  
<https://pkra.github.io/slides-ahg>

Web documents can be rendered by including MathJax directly:

```
pandoc --mathjax=https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.5/latest.js?config=TeX-AMS-MML_HTMLorMML example.docx
```

- Use it directly from CDNjs
- Configure according to the need of your web document
- Local installations possible
- Detailed documentation available at:  
`http://docs.mathjax.org`
- Large user community and support

# Configuring MathJax: CDN

- Load directly from its Content Distribution Network
  - Include single line script tag into web document
  - Example with broad, standard configuration

```
<script src='https://cdnjs.cloudflare.com/ajax/libs/  
mathjax/2.7.5/MathJax.js?  
config=TeX-MML-AM_CHTML'  
async</script>
```



# Configuring MathJax: Locally

- Local configurations to customise for your web content
  - Allows for fine-grained control of MathJax's behaviour
  - Needs to be added BEFORE the CDNjs call
  - Example for including inline  $\text{\LaTeX}$  formulas:

```
<script type="text/x-mathjax-config">
MathJax.Hub.Config({
  tex2jax: {
    inlineMath: [['$', '$'], ['\\(', '\\)']]
  },
});
</script>
```

## Part 2:

## Using accessible Maths on the Web (with MathJax)

# MathJax's Assistive Technology Extension

- Inbuilt and optional accessibility features
- Selectable in context menu since MathJax v2.7

More details at:

<https://www.mathjax.org/mathjax-accessibility-extensions-v1-now-available/>

# MathJax's AT Features

- Magnification
- Responsive Equations and Abstraction
- Highlighting
- Interactive Exploration
- Speech Generation

- Zoom feature for single math expression
- Magnification up to 500%
- Option selectable in context menu
- Customisable trigger via mouse and keyboard actions

# Responsive Equations

- Responsive design enhances reflow and readability of math documents
- Automatic reflow for simplifying layout, adapting to form factor of display and magnification
- Intelligent linebreaking by exploiting semantic enrichment
  - Don't break in the middle of an expression
- Chunking: Abstracting over large elements
  - collapsing mathematically meaningful sub-expressions

- “Walkers” allow to interactively dive into mathematical expression
- Synchronised highlighting together with aural rendering
- Use `<shift><space>` to explore expressions
- Simple navigation model using arrow keys
- Different types of walkers: syntactic, semantic
- Interactive collapse and expansion of sub-expressions

# Aural Rendering and Highlighting

- Speech strings are computed with Speech Rule Engine (SRE)
- Currently uses the MathSpeak rules: verbose, brief, superbrief
  - special summarisations for collapsed parts
  - Other rule sets and localisations in the future
- WAI-ARIA and CSS to implement interactive exploration
  - Speech output by updating ARIA live regions
  - Colour/contrast changes by rewriting CSS properties
- Speech strings can be precomputed or generated on the fly
- Works for all renderers MathJax provides



- 1 Build a web document for the quadratic formula:

$$\backslash[\quad x = \backslashfrac{-b \quad \backslash\mathbf{pm} \quad \backslash\mathbf{sqrt}\{b^2-4ac\}}{2a} \quad \backslash]$$

- 2 Include basic MathJax configuration, e.g. take it from <https://www.mathjax.org/#docs>
- 3 Experiment with the MathJax accessibility extension
  - 1 Load it from sub menu
  - 2 Use <shift><space> to explore expressions
  - 3 Experiment with a screen reader

# Hint: Basic web document

```
<html>  
  <head>  
    ... MathJax script tag goes here ...  
  </head>  
  
  <body>  
    ... Quadratic equation goes here ...  
  </body>  
</html>
```

After loading the explorer extension, you will see that subtitles are still greyed out in the sub-menu.

④ Switch sub-titles on programmatically in your page

- ① Add a configuration option for MathJax
- ② Make sure to add it BEFORE the call to CDN

```
menuSettings: {  
  'Assistive-subtitle': true  
}
```

⑤ If you want the explorer to always be loaded, add this to the menuSettings:

```
explorer: true,
```

## Hint: Full Configuration Tag

```
<script type="text/x-mathjax-config">
  MathJax.Hub.Config({
    menuSettings: {
      explorer: true,
      'Assistive-subtitle': true
    }
  });
</script>
```

For other demonstrations and experimental tools follow the links at

<https://github.com/mathjax/MathJax-a11y>

# Solution 1

```
<html><head>
<script src='https://cdnjs.cloudflare.com/ajax/
libs/mathjax/2.7.5/MathJax.js?config=TeX-MML-AM_CHTML'
      async></script>
</script>
</head>

<body>
\[
  x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}
\]
</body>
</html>
```

# Solution 1 advanced

```
<html><head>
<script type="text/x-mathjax-config">
  MathJax.Hub.Config({
    menuSettings: {
      explorer: true,
      'Assistive-subtitle': true });
</script>
<script src='https://cdnjs.cloudflare.com/ajax/\
libs/mathjax/2.7.5/MathJax.js?config=TeX-MML-AM_CHTML'
  async></script>
</head>

<body>
\[
  x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}
\]
</body>
</html>
```

## Solution 2

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD_HTML//EN">
<html><head>
<title>Empower-18 Exercise</title>
<script type="text/x-mathjax-config">
  MathJax.Hub.Config({
    menuSettings: {explorer: true,
'Assistive-subtitle': true}});
</script>
<script src='https://cdnjs.cloudflare.com/ajax/\
libs/mathjax/2.7.5/MathJax.js?config=TeX-MML-AM_CHTML'
  async></script>
</head>

<body>
<h1>Quadratic Equation</h1>
\[ x = \frac{-b \pm \sqrt{b^2-4ac}}{2a} \]
<hr>
Last modified: Sun Nov 13 20:11:55 MST 2016
</body> </html>
```



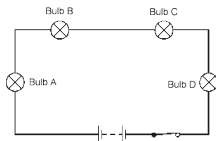
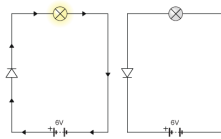
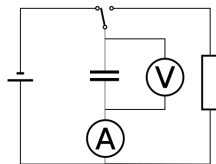
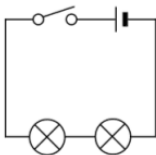
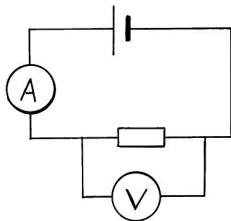
## Part 3:

## Using accessible diagrams on the Web

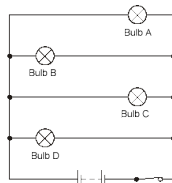
# Goal of our Work

- Diagrams are an important teaching means in STEM
- Accessibility is key to inclusive education
- Avoid additional hurdles for both authors and readers
- **Fully automatic** to reduce need for human intervention
- Source independence
  - Do not rely on the benevolent, educated author
- Tool Independence
  - Do not force readers to use a specialist software tool
- Platform independence
  - Accessible with all browsers, screen readers, on all platforms

# Physics: Electric Circuits Examples

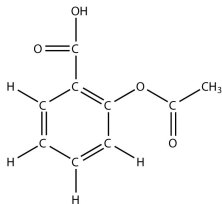


Circuit 1

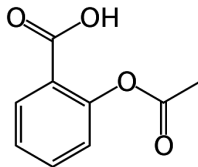


Circuit 2

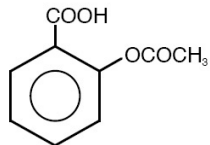
# Chemistry Examples



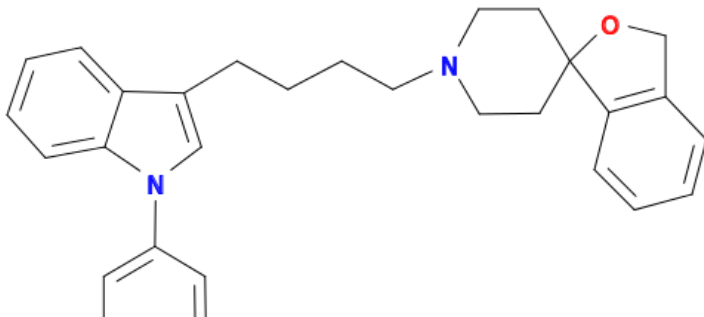
Displayed formula.



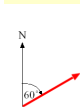
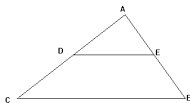
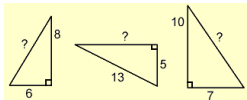
Skeletal formula.



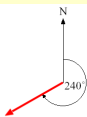
Structural formula.



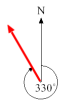
# Future: Other Subject Areas



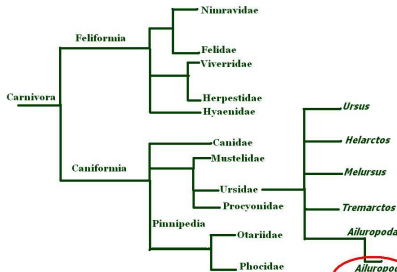
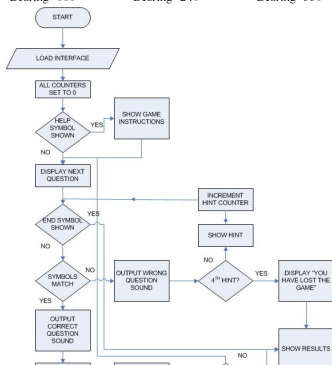
Bearing 060°



Bearing 240°



Bearing 330°



- Simple navigation model with arrow keys
- Screen Reader Support:
  - Generate speech output on different levels
  - Display of speech output using subtitling
- Low Vision/Learning Disability Support:
  - Highlighting of inspected components
  - Optional zooming and magnification of components
  - Changing contrast, colour configurations
- l18n support

# Interactive Session 1

<https://www.surveymonkey.com/r/RNWMVQL>

## Part 4:

## Generating accessible diagrams for the Web



# Automatic Generation of Accessible SVG

- Start with (bitmap) images of diagrams
- Understand their content
- Enrich information specifically for teaching
- Generate “content-heavy” SVG

Do all this fully automatically without user interaction

Input: A bitmap image or a specification code (in the case of chemistry diagrams)

- ① Upload the image or code
- ② Have it recognised
- ③ Navigate the recognised image
- ④ Alternatively:
  - Download the recognised image
  - Print it on an embosser.

<https://www.surveymonkey.com/r/RCQTG3B>