



The poster features a central circular graphic with '2024' in the center, surrounded by icons for 'Literature', 'Education', 'Software', 'Worldwide via the Internet', and 'Research'. To the right, a large '50' is displayed with 'Years' and 'OF HISTORY' on banners. Below this, the text reads: 'A celebration of 50 years writing sign languages! Comemoramos 50 anos escrevendo as Línguas de Sinais!'. The main title is 'SignWriting Symposium 2024' and 'Simpósio SignWriting 2024'. The event details are '4TH DAY: SOFTWARE | 4º DIA: PROGRAMAS' on Thursday July 18, 2024. A vertical purple bar on the right contains various sign-writing symbols.

SignWriting Symposium  
2024  
Literature  
Education  
Software  
Worldwide via the Internet  
Research

**50** Years OF HISTORY

A celebration of 50 years writing sign languages!  
Comemoramos 50 anos escrevendo as Línguas de Sinais!

**SignWriting Symposium 2024**  
**Simpósio SignWriting 2024**

**4TH DAY: SOFTWARE** | **4º DIA: PROGRAMAS**  
Thursday July 18, 2024 | Quinta-feira, 18 de julho de 2024

# Formal SignWriting

---

Basics, Software, Visuals, and Searching



## Steve Slevinski

Sign language user, friend of the deaf, and SignWriting Evangelist

Creator of SignPuddle

Sutton SignWriting Encoder of 2010, 2012, and 2017

System Administrator of SignWriting.org

Author of the Sutton SignWriting Packages

<https://steveslevinski.me/>

# Sutton SignWriting

- Sutton SignWriting is the universal and complete solution for written sign language. It has been applied by a wide and deep international community of sign languages.
- Sutton SignWriting is an international standard for writing sign languages by hand or with computers. From education to research, from entertainment to religion, SignWriting has proven useful because people are using it to write signed languages.

# Formal SignWriting

- Formal SignWriting is one particular computerized encoding for Sutton SignWriting. The design is based on character processing with regular expressions. With Formal SignWriting, each sign is written as a two-part word of time and space.
- Where as American Sign Language is a natural language, Formal SignWriting is a formal language. A formal language uses words and punctuation to form text. Each word is expressed as a string of characters. Well-formed words are governed by the structural rules of the grammar. A formal language is useful in mathematics, computer science, and linguistics.

# Formal SignWriting Design Principles

- **Complete:** Sutton SignWriting is a complex script with unique requirements and processing. Formal SignWriting supports all of the structures inherent to the script.
- **Universal:** Sutton SignWriting can be used to write any sign language, natural or constructed. Formal SignWriting supports all sign languages without requiring the addition of new characters or updated fonts.
- **Empowering:** Sutton SignWriting is flexible enough to let each writer decide how they want to write their signs. Formal SignWriting enable the writers to decide for themselves the spelling of their respective signs.
- **Possible:** Sutton SignWriting is a practical script that makes it possible to write sign language. Formal SignWriting is a practical encoding because it works with existing font technologies across operating systems.

# Table of Contents

- **Basics:** characters sets, two-part word, sorting, and punctuation
- **Software:** core, font-db, font-ttf, sgnw-components, and signmaker
- **Visuals:** SVG with fonts, self-contained SVG, and the styling string
- **Searching:** query language, regular expressions, and transformations

# Basics: Character Sets

- **Formal SignWriting in ASCII (FSW)**
  - Formal SignWriting in ASCII (FSW) was released in January 2012 and has been stable since. FSW only uses characters from the ASCII subset of "ABLMRS0123456789xabcdef"
- **SignWriting in Unicode (SWU)**
  - SignWriting in Unicode (SWU) was first published in October 2016 and officially submitted to the Unicode Technical Committee in July 2017. SWU is not part of the Unicode standard.
  - SignWriting in Unicode (SWU) is an experimental Unicode design that is supported by the Sutton SignWriting community. This alternate encoding overwrites the Sutton SignWriting block in Unicode and uses plane 4 for the SignWriting symbols.

# Basics: Character Sets

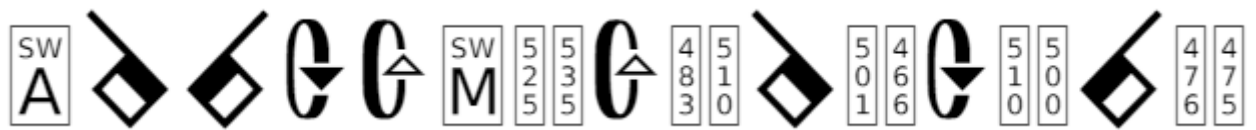


- Formal SignWriting in ASCII (FSW)

- AS10011S10019S2e704S2e748M525x535S2e748483x510S10011501x466S2e704510x500S10019476x475

- SignWriting in Unicode (SWU)

- U+1D800 U+40012 U+4001A U+4B6A5 U+4B6E9 U+1D803 U+1D91F U+1D929 U+4B6E9 U+1D8F5  
U+1D910 U+40012 U+1D907 U+1D8E4 U+4B6A5 U+1D910 U+1D906 U+4001A U+1D8EE U+1D8ED





# Basics: Character Sets

| Description     | FSW Characters   | SWU Characters     |
|-----------------|------------------|--------------------|
| Sequence Marker | A                | U+1D800            |
| Signbox Markers | B, L, M, R       | U+1D801 to U+1D804 |
| Numbers         | 250 to 749       | U+1D80C to U+1D9FF |
| Symbols         | S10000 to S38b07 | U+40001 to U+4F428 |

# Basics: Character Sets

| FSW String | SWU Character | SWU Glyph |
|------------|---------------|-----------|
| A          | U+1D800       |           |
| S10011     | U+40012       |           |
| S10019     | U+4001A       |           |
| S2e704     | U+4B6A5       |           |
| S2e748     | U+4B6E9       |           |
| M          | U+1D803       |           |
| 525        | U+1D91F       |           |


# Two-Part Word of Time and Space

- **Temporal Prefix** - The temporal prefix is a one-dimensional list of symbols that is written by an author. The arrangement of the symbols is based on a particular theory of sorting. The order of the symbols in the temporal prefix is significant because sorting is possible with a binary string comparison. The temporal prefix is neither formatting nor style and represents meaning not found in the spatial signbox.
- **Spatial Signbox** - The spatial signbox is a two-dimensional cluster of symbols. The position of each symbol is determined by the writer and defined using Cartesian Coordinates that represent the top-left of the symbol image. 2-dimensional space does not have a normative 1-dimensional order. When symbols overlap, the relative order of the overlapping symbols is important. Symbols written first appear underneath symbols that are written later. Otherwise, the exact string order of the spatial symbols is unpredictable. The spatial signbox is neither formatting nor style and represents meaning that is beyond the temporal prefix.


# FSW: Two-Part Word

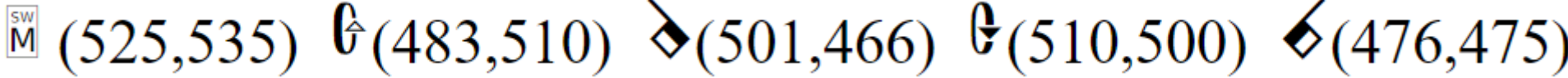
- 1) AS10011S10019S2e704S2e748M525x535S2e748483x510S10011501x466S2e704510x500S10019476x475  
---
- 2) Time: AS10011S10019S2e704S2e748  
Space: M525x535S2e748483x510S10011501x466S2e704510x500S10019476x475  
---
- 3) Time: A S10011 S10019 S2e704 S2e748  
Space: M 525x535 S2e748 483x510 S10011 501x466 S2e704 510x500 S10019 476x475  
---
- 4) Time: Sequence Marker and List of Symbols  
Space: M (525,535) S2e748 (483,510) S10011 (501,466) S2e704 (510,500) S10019 (476,475)  
---
- 5) Space: Middle Lane Signbox Marker, Max Coordinate, List of Symbols with Top-Left Coordinates

# SWU: Two-Part Word

1)  ---

2) Time:   
Space:  ---

3) Time: Sequence Marker and List of Symbols  
Space:  ---

4) Space:  ---

5) Space: Middle Lane Signbox Marker, Max Coordinate, List of Symbols with Top-Left Coordinates

# FSW: Punctuation

- 1)** S38800464x496  
---
- 2)** S38800 464x496  
---
- 3)** S38800 (464,496)  
---
- 4)** Punctuation Symbol, Top-Left Coordinate  
---
- 5)** Max Coordinate = (1000-464, 1000-496)  
---
- 6)** Max Coordinate = (536, 504)

# SWU: Punctuation

1) 

|   |   |
|---|---|
| 4 | 4 |
| 6 | 9 |
| 4 | 6 |

---

2)  (464, 496)

---

3) Punctuation Symbol, Top-Left Coordinate

---

4) Max Coordinate = (1000-464, 1000-496)

---

5) Max Coordinate = (536, 504)

# Software: Development Packages

**@sutton-signwriting/core** - a javascript package for node and browsers that supports general processing of SignWriting text.

**@sutton-signwriting/font-db** - a javascript package for node that generates SVG and PNG images for individual symbols, complete signs, and structured text using pure SVG.

**@sutton-signwriting/font-ttf** - a javascript package for the browser that generates SVG and PNG images for individual symbols, complete signs, and structured text using TrueType Fonts.

**@sutton-signwriting/sgnw-components** - a javascript package for the browser that includes a collection of web components built with <https://stenciljs.com/>

<https://github.com/sutton-signwriting/>

<https://www.npmjs.com/search?q=sutton-signwriting>

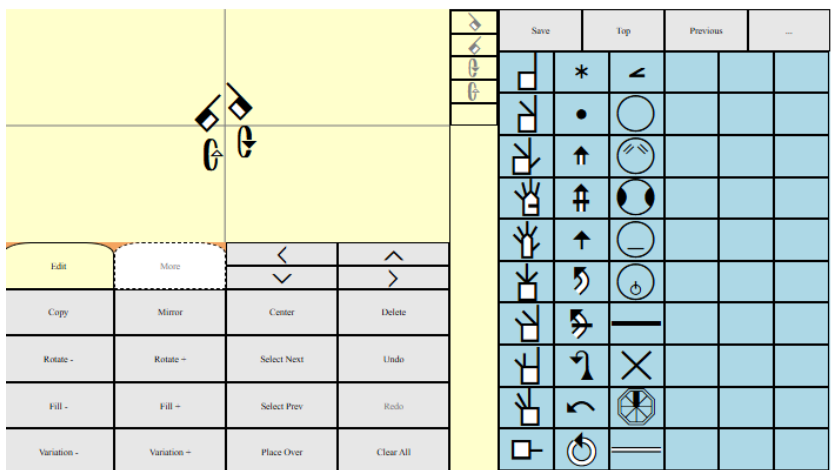


# SignMaker: iframe Editor

## using postMessage and addEventListener

**@sutton-signwriting/signmaker** - an online editor that can be accessed directly, embedded in an iFrame, and downloaded. It uses both Formal SignWriting in ASCII (FSW) and SignWriting in Unicode (SWU) character sets, along with the associated style string.

```
<iframe style="width:640px;height:360px;" src="https://www.sutton-signwriting.io/signmaker/index.html"></iframe>
```



### Messages Received

- {"signmaker":"save","swu":"A d 6 6 6 M 6 6 6 6 6 6 6 6 6","fsw":"AS100"}



### Messages Sent

- {"swu":"A d 6 6 6 M 6 6 6 6 6 6 6 6 6"}

<https://www.sutton-signwriting.io/signmaker/>

<https://www.sutton-signwriting.io/signmaker/demo.html>

# Visuals: Common SVG Wrapper



1)

2)

3) Max Coordinate – (525,535)

Min X =  $\min(483, 501, 510, 476) = 476$

Min Y =  $\min(510, 466, 500, 475) = 466$

Width = Max X – Min X =  $525 - 476 = 49$

Height = Max Y – Min Y =  $535 - 466 = 69$

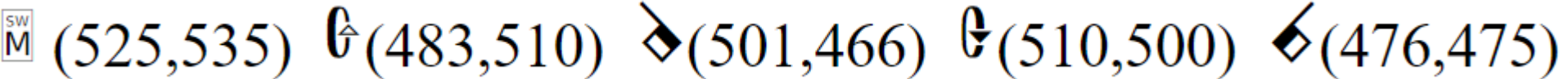
4) 

```
<svg version="1.1" xmlns="http://www.w3.org/2000/svg"
  width="49" height="69" viewBox="476 466 49 69">
  <text font-size="0"> </text>
  ...
</svg>
```

# Visuals: Self-Contained SVG



1) 

2) 

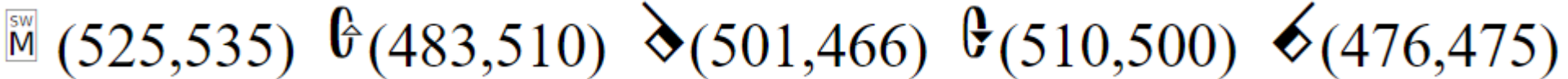
3) Each symbol gets an <svg> group

```
<svg x="483" y="510">  
  <g transform="translate(...)">  
    <path class="sym-fill" d="M895 ..."/>  
    <path class="sym-line" d="M345 ..."/>  
  </g>  
</svg>
```

# Visuals: SVG with Fonts



1) 

2) 

3) Each symbol gets a `<g>` group

```
<g transform="translate(483,510)">
  <text class="sym-fill" style="font-family:'SuttonSignWritingFill';">␣</text>
  <text class="sym-line" style="font-family:'SuttonSignWritingLine';">␣</text>
</g>
```

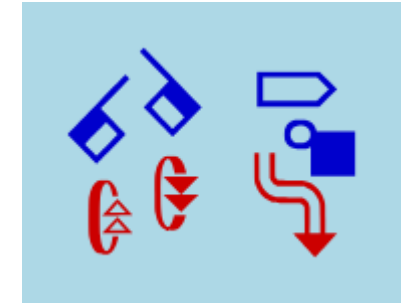
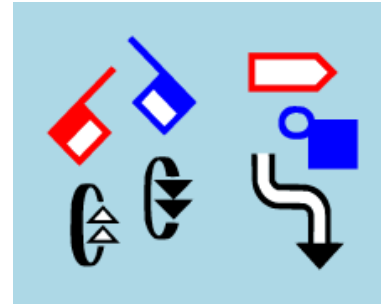
# Visuals: CSS for SVG with Fonts



```
@font-face {
  font-family: "SuttonSignWritingLine";
  src:
    local('SuttonSignWritingLine'),
    url('./SuttonSignWritingLine.ttf') format('truetype'),
    url('https://.../SuttonSignWritingLine.ttf') format('truetype');
}

@font-face {
  font-family: "SuttonSignWritingFill";
  src:
    local('SuttonSignWritingFill'),
    url('./SuttonSignWritingFill.ttf') format('truetype'),
    url('https://.../SuttonSignWritingFill.ttf') format('truetype');
}
```

# Visuals: Styling String



- Colorize
- Padding
- Background color
- Detail color for line and fill

- Zoom level
- Colors for individual symbols
- Class names for SVG
- ID name for SVG

<https://www.sutton-signwriting.io/core/#style>

# Visuals: Styling String parse and compose

```
style.parse(' -CP10G_blue_D_red,Cyan_')  
  
return {  
  'colorize': true,  
  'padding': 10,  
  'background': 'blue',  
  'detail': ['red', 'Cyan']  
}
```

```
style.compose({  
  'colorize': true,  
  'padding': 10,  
  'background': 'blue',  
  'detail': ['red', 'Cyan']  
})  
  
return ' -CP10G_blue_D_red,Cyan_'
```

<https://www.sutton-signwriting.io/core/#styleparse>

<https://www.sutton-signwriting.io/core/#stylecompose>

# Searching: Query Language

The query language of Formal SignWriting allows for precise searching of signs written in either FSW or SWU. A query string is a concise representation for a much larger and detailed set of regular expressions. The regular expressions can be used to quickly and accurately search large files and databases containing Formal SignWriting.

<https://www.ietf.org/archive/id/draft-slevinski-formal-signwriting-09.html#name-query-language>



# Searching: Query Language Possibilities

Search the temporal prefix or the spatial signbox

Search for exact symbols

Search for symbols with any fill or any rotation

Search for symbols near a coordinate within a specified range

Search for a range of symbols

Search for any symbol within a list of individual symbols

<https://www.sutton-signwriting.io/core/#fswquery>

<https://www.sutton-signwriting.io/core/#swuquery>

# Searching: Regular Expression Basics

| Character | Description                          | Example                                 |
|-----------|--------------------------------------|---|
| *         | Match a literal 0 or more times      | ABC* matches AB, ABC, ABCC, ...         |
| +         | Match a literal 1 or more times      | ABC+ matches ABC, ABCC, ABCCC, ...      |
| ?         | Match a literal 0 or 1 times         | ABC? matches AB or ABC                  |
| {#}       | Match a literal "#" times            | AB{2} matches ABB                       |
| [ ]       | Match any single literal from a list | [ABC] matches A, B, or C                |
| [ - ]     | Match any single literal in a range  | [A-C] matches A, B, or C                |
| ( )       | Creates a group for matching         | A(BC)+ matches ABC, ABCBC, ABCBCBC, ... |
| (   )     | Matches one of several alternatives  | (AB BC CD) will match AB, BC, or CD     |
| (?:)      | Creates a non-capturing group        | A(?:BC) will match ABC as one group     |

# Searching: Regex Number Searching

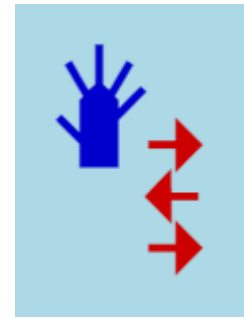
**Find a number from 423 to 612**

| Incremental Step              | Regex       |
|-------------------------------|-------------|
| Find a number from 423 to 429 | 42[3-9]     |
| Find a number from 430 to 499 | 4[3-9][0-9] |
| Find a number from 500 to 599 | 5[0-9][0-9] |
| Find a number from 600 to 609 | 60[0-9]     |
| Find a number from 610 to 612 | 61[0-2]     |

**Final Regular expression**

`(42[3-9]|4[3-9][0-9]|5[0-9][0-9]|60[0-9]|61[0-2])`

# Transformation: FSW to Query String



FSW of **AS14c20S27106M519x529S14c20482x471S27106504x489**

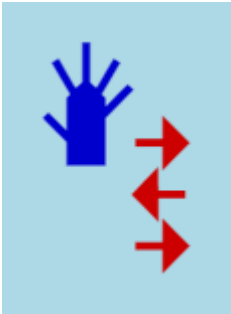
Exact Symbols in temporal prefix: **QAS14c20S27106T**

General Symbols in temporal prefix: **QAS14cuuS271uuT**

Exact Symbols in spatial signbox: **QS14c20S27106**

Exact Symbols near coordinates: **QS14c20482x471S27106504x489**

<https://www.sutton-signwriting.io/core/#fswqueryfsw2query>



# Transformation: SWU to Query String

SWU of 

|    |
|----|
| SW |
| A  |

|    |
|----|
| SW |
| M  |

|   |   |   |
|---|---|---|
| 5 | 1 | 9 |
| 5 | 2 | 9 |

|   |   |   |
|---|---|---|
| 4 | 8 | 2 |
| 4 | 7 | 1 |

|   |   |   |
|---|---|---|
| 5 | 0 | 4 |
| 4 | 8 | 9 |

Exact Symbols in temporal prefix: QA T

General Symbols in temporal prefix: QA fr frT

Exact Symbols in spatial signbox: Q

Exact Symbols near coordinates: Q 

|   |   |   |
|---|---|---|
| 4 | 8 | 2 |
| 4 | 7 | 1 |

|   |   |   |
|---|---|---|
| 5 | 0 | 4 |
| 4 | 8 | 9 |

<https://www.sutton-signwriting.io/core/#swuqueryswu2query>

# Transformation: FSW Query String to Regex

```
fswquery.regex('QS100uuS20500480x520')  
  
return [  
  '(?:A(?:S[123][0-9a-f]{2}[0-5][0-9a-f])+)?[BLMR]([0-9]{3}x[0-9]{3})(S[123]  
  '(?:A(?:S[123][0-9a-f]{2}[0-5][0-9a-f])+)?[BLMR]([0-9]{3}x[0-9]{3})(S[123]  
  ]
```

<https://www.sutton-signwriting.io/core/#fswqueryregex>

# Transformation: SWU Query String to Regex

```
swuquery.regex('QA↵T')  
  
return [  
  '(\uD836\uDC00\uD8C0\uDC12(?:(:\uD8C0[\uDC01-\uDFFF])|(:[\uD8C1-\uD8FC]  
])
```

<https://www.sutton-signwriting.io/core/#swuqueryregex>

# Transformation: Regex to FSW or SWU

Wherever regular expressions are supported, the regex created can be used to search text for matching signs written in either FSW or SWU.



# Would you like to know more?

The SignWriting Stream:

[https://www.youtube.com/channel/UCXu4AXlG0rXFtk\\_5SzumDow](https://www.youtube.com/channel/UCXu4AXlG0rXFtk_5SzumDow)

# Would you like to know more?

About SignWriting Software:

<https://www.sutton-signwriting.io/>

Technical Specification:

<https://www.ietf.org/archive/id/draft-slevinski-formal-signwriting-09.html>

# Would you like to know more?

About Me:

<https://SteveSlevinski.me/>

Email:

[slevinski@signwriting.org](mailto:slevinski@signwriting.org)

Online Chat:

<https://gitter.im/sutton-signwriting/community>

# Would you like to support our work?

Use SignWriting and enjoy!

Discuss and build on our work.

Consider financially supporting our work.

<https://www.patreon.com/signwriting>