

表層形	品詞	品詞細 1	品詞細 2	品詞細 3	活用形	活用例	原形	読み	発音
東京	名詞	固有名詞	地域	一般	*	*	東京	トウキョウ	トーキョー
スカイ	名詞	一般	*	*	*	*	スカイ	スカイ	スカイ
ツリー	名詞	一般	*	*	*	*	ツリー	ツリー	ツリー
の	助詞	連体化	*	*	*	*	の	ノ	ノ
最寄り駅	名詞	一般	*	*	*	*	最寄り駅	モヨリエキ	モヨリエキ
は	助詞	係助詞	*	*	*	*	は	ハ	ワ
とう	副詞	助詞類接続	*	*	*	*	とう	トウ	トウ
きょう	名詞	副詞可能	*	*	*	*	きょう	キョウ	キョー
スカイ	名詞	一般	*	*	*	*	スカイ	スカイ	スカイ
ツリー	名詞	一般	*	*	*	*	ツリー	ツリー	ツリー
駅	名詞	接尾	地域	*	*	*	駅	エキ	エキ
です	助動詞	*	*	*	特殊・デス	基本形	です	デス	デス

Kana Selection

You can choose to output ruby in Hiragana (default) or Katakana:

```
#import "@preview/auto-jruby:0.3.4": *
#set page(width: auto, height: auto, margin: 0.5cm)

#let sample = "東京スカイツリーの高さは634mです"

// Default (Hiragana)
#show-ruby(sample)
// Katakana
#show-ruby(sample, kana: "katakana")
```

とうきょう たか
東京 スカイツリーの高さは 634m です

トウキョウ タカ
東京 スカイツリーの高さは 634m です

API Reference

show-ruby

Renders the input text with automatic furigana.

```
#show-ruby(  
  input-text,  
  size: 0.5em,  
  leading: 1.5em,  
  ruby-func: auto,  
  user-dict: none,  
  dict: "ipadic",  
  kana: "hiragana"  
)
```

Parameters:

- `input-text` (string): The Japanese text to analyze and render.
- `size` (length): The font size of the ruby text. Defaults to 0.5em.
- `leading` (length): The vertical space between lines to accommodate ruby text. Defaults to 1.5em.
- `ruby-func` (function | auto): A custom ruby function from the `rubby` package.
- If `auto`, it uses the default configuration (`get-ruby(size: size)`).
- If provided, it allows advanced customization of ruby positioning (e.g., specific `pos` or `alignment`).
- `user-dict` (string | array | none): Optional user dictionary for custom tokenization.
- If `string`: A CSV-formatted string with custom dictionary entries.
- If `array`: An array of arrays, where each inner array represents a CSV row.
- If `none`: No user dictionary is used.
- `dict` (string): The dictionary to use for tokenization. Must be one of:
 - `"ipadic"` (default): Standard Japanese dictionary
 - `"unidic"`: Alternative dictionary with different grammatical analysis
- `kana` (string): The script to use for the ruby text. Must be one of:
 - `"hiragana"` (default): Converts dictionary readings to Hiragana.
 - `"katakana"`: Uses Katakana readings.

show-analysis-table

Renders a table displaying the morphological breakdown of the text.

```
#show-analysis-table(
  input-text,
  user-dict: none,
  dict: "ipadic"
)
```

Parameters:

- `input-text` (string): The text to analyze.
- `user-dict` (string | array | none): Optional user dictionary for custom tokenization.
- `dict` (string): The dictionary to use. Must be one of: `"ipadic"` (default) or `"unidic"`.

Table Columns:

The columns displayed depend on the selected `dict`.

If `dict: "ipadic"` (10 columns):

1. **Surface Form** (表層形): The word as it appears in the text.
2. **Part of Speech** (品詞): Grammatical category (Noun, Verb, etc.).
3. **POS Subcategory 1** (品詞細分類 1)
4. **POS Subcategory 2** (品詞細分類 2)
5. **POS Subcategory 3** (品詞細分類 3)
6. **Conjugation Form** (活用形)
7. **Conjugation Type** (活用型)
8. **Base Form** (原形): The dictionary form of the word.
9. **Reading** (読み): Katakana reading.
10. **Pronunciation** (発音)

If `dict: "unidic"` (18 columns):

1. **Surface Form** (表層形)
2. **POS Major** (品詞大分類)
3. **POS Medium** (品詞中分類)
4. **POS Minor** (品詞小分類)
5. **POS Fine** (品詞細分類)
6. **Conjugation Type** (活用型)
7. **Conjugation Form** (活用形)
8. **Lexeme Reading** (語彙素読み)
9. **Lexeme** (語彙素)

10. **Orthographic Surface** (書字形出現形)
11. **Phonological Surface** (発音形出現形)
12. **Orthographic Base** (書字形基本形)
13. **Phonological Base** (発音形基本形)
14. **Word Type** (語種)
15. **Initial Mutation Type** (語頭変化型)
16. **Initial Mutation Form** (語頭変化形)
17. **Final Mutation Type** (語末変化型)
18. **Final Mutation Form** (語末変化形)

tokenize

Low-level function that returns the raw JSON data from the WASM plugin. Useful if you want to process the analysis data manually.

```
#tokenize(
  input-text,
  user-dict: none,
  dict: "ipadic"
)
```

Parameters:

- `input-text` (string): The text to tokenize.
- `user-dict` (string | array | none): Optional user dictionary for custom tokenization.
- `dict` (string): The dictionary to use. Must be one of: "ipadic" or "unidic".

Returns: An array of dictionaries containing:

- `surface` (string): The surface form of the token.
- `details` (array of strings): The raw detailed information for the token. The content and length depend on the dictionary used (e.g., POS, conjugation, reading, etc.).
- `ruby_segments` (array of dictionaries): A pre-calculated list of segments for furigana, where each item has `text` and `ruby` fields.

User Dictionary Format

The user dictionary allows you to define custom word segmentation and readings. It uses a simple CSV format with three columns:

```
<surface>,<part_of_speech>,<reading>
```

- `surface`: The word as it appears in text
- `part_of_speech`: Custom part-of-speech label (e.g., “カスタム名詞”)
- `reading`: Katakana reading for the word

Usage Examples:

Method 1: Inline string

```
#let sample = "東京タワーの最寄駅は赤羽橋駅です"
#let user-dict-str = "赤羽橋駅,カスタム名詞,アカバネバシエキ"
```

```
#show-ruby(sample)
#show-ruby(sample, user-dict: user-dict-str)
```

Method 2: Array of arrays

```
#let sample = "東京タワーの最寄駅は赤羽橋駅です"
#let user-dict-array = (
  ("赤羽橋駅", "カスタム名詞", "アカバネバシエキ")
)
```

```
#show-ruby(sample)
#show-ruby(sample, user-dict: user-dict-array)
```

Method 3: Load from CSV file

```
$ cat user_dict.csv
赤羽橋駅,カスタム名詞,アカバネバシエキ
#let sample = "東京タワーの最寄駅は赤羽橋駅です"
#let user-dict-from-file = csv("user_dict.csv")

#show-ruby(sample)
#show-ruby(sample, user-dict: user-dict-from-file)
```

とうきょう もよりえき あかはねきょうえき
東京 タワーの最寄駅は赤羽 橋 駅です

とうきょう もよりえき あかばねばしえき
東京 タワーの最寄駅は赤羽橋 駅です

Under the Hood

This package uses **Lindera** (a Rust port of Kuromoji) with two available dictionary options:

- **IPADIC**: Standard Japanese morphological dictionary
- **UniDic**: Alternative dictionary with different part-of-speech classifications

The processing workflow:

1. The text is passed from Typst to the Rust WASM plugin.
2. Lindera tokenizes the text using the specified dictionary and retrieves readings.
3. A custom algorithm aligns the readings with the surface form to separate okurigana (kana endings of verbs/adjectives) from the kanji stems.
4. The structured data is returned to Typst and rendered using the `rubby` package for furigana display.

Optional: Enabling IPADIC-NEologd

IPADIC-NEologd Support

IPADIC-NEologd (an extended dictionary with contemporary terms and named entities) has been removed from the default distribution due to its large file size. However, you can manually enable it if needed:

1. Navigate to `./wasm-plugins/ipadic-neologd` and build the WASM module:

```
cargo build --target wasm32-unknown-unknown --release
```

1. Copy the built WASM file to the package directory:

```
cp ./target/wasm32-unknown-unknown/release/ipadic_neologd.wasm ../../package/
ipadic_neologd.wasm
```

1. Navigate to `./package` and update `./package/lib.typ` to allow the `ipadic-neologd` option. Change:

```
if dict not in ("ipadic", "unidic") {
  panic("dict must be one of: ipadic, unidic")
}
```

to:

```
if dict not in ("ipadic", "ipadic-neologd", "unidic") {
  panic("dict must be one of: ipadic, ipadic-neologd, unidic")
}
```

1. Install the package locally:

```
just install
```

1. Import and use with @local:

```
#import "@local/auto-jruby:0.3.3": *  
#let sample = "東京スカイツリーの最寄り駅はとうきょうスカイツリー駅です"  
#show-ruby(sample, dict: "ipadic-neologd")
```

License

This project is distributed under the AGPL-3.0-or-later License. See LICENSE for details.