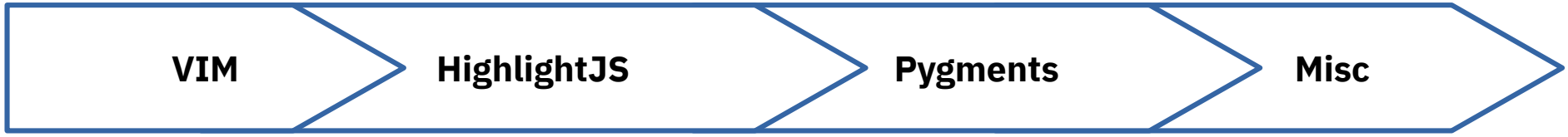
Syntax Highlighting Libraries for Rexx

Syntax Highlighting



Syntax Highlighter

- Pattern-matching engine based on regular expressions
  - **`syntax keyword`** — matches exact words

```
syn keyword rexxBuiltinClass .Alarm .ArgUtil .Array .Bag .CaselessColumnComparator
```

  - **`syntax match`** — matches a regex pattern

```
syn match rexxKeywordStatements "\<\(arg\|catch\|do\|drop\|end\|exit\|expose\|finally\|forward\|if\|interpret\|iterate\|leave\|loop\|nop\)\>"
```

  - **`syntax region`** — matches a start/end pattern pair (comments, …)

```
syn region rexxComment start="/\*" end="\*/"
```

- Each match is assigned to a syntax group, which is then linked to a highlight group that carries the actual color/style attributes

```
hi def link rexxComment Comment
```


<https://github.com/vim/vim/blob/master/runtime/syntax/rexx.vim>

Till Winkler

Syntax File

- **Used in:** Vim, Neovim, Gvim
  - Inspired other syntax highlighting libraries
- Currently lack support for ooRexx 4.0.0+ features
- Well documented
- Minimal changes were need

```
" Vim syntax file
" Language:         Rexx
" Maintainer:       Thomas Geulig <geulig@nentec.de>
" Last Change:  2012 Sep 14, added support for new ooRexx 4.0 features
" URL:              http://www.geulig.de/vim/rexx.vim
" Special Thanks to Dan Sharp <dwsharp@hotmail.com> and Rony G. Flatscher
" <Rony.Flatscher@wu-wien.ac.at> for comments and additions
```

Version	Instructions	Directives	Classes	Env Variables
4.2.0	USE ARG, LOSTDIGITS			
5.0.0	USE LOCAL, SELECT CASE, ADDRESS WITH, GUARD ON WHEN, USE ARG ...rest, [...], ns:Class	::ANNOTATE, ::RESOURCE, ::REQUIRES	.AlarmNotification, .EventSemaphore, .IdentityTable, .MessageNotification, .MutexSemaphore, .RexxInfo, .Singleton, .StringTable, .Ticker, .Validate, .VariableReference	.debuginput, .traceoutput, .syscargs, .resources
5.1.0			.TraceObject	
5.2.0	rexxextensions	::OPTIONS ERROR/FAILURE/... CONDITION		

<https://github.com/vim/vim/blob/master/runtime/syntax/rexx.vim>

HighlightJS

- Syntax highlighting library written in JavaScript
- No Rexx or ooRexx support
- Pattern-matching engine based on **regular expressions**
- 190+ languages supported
  - Each language is defined as a JavaScript object describing modes
- 250+ themes as CSS files
- **Used in:** GitLab, Stack Overflow, Discord, ...

<https://github.com/highlightjs/highlight.js/blob/main/SUPPORTED_LANGUAGES.md>

Mapping between Vim and HighlighJS (Excerpt)

Category	OoRexx Element	Vim Token	HighlightJS Scope
Block comment	`/* … */`	`rexxComment`	`BLOCK_COMMENT`
Line comment	`-- …`	`rexxLineComment`	`LINE_COMMENT`
Comment error	`*/`		
Double-quoted	`"…"`	`rexxString`	`STRING`
Double-quoted	`'…'`	`rexxString`	`STRING`
::RESOURCE body	`Multi-line text until ::`	`rexxResourceBody`	`RESOURCE_REGION`
Core instructions	`say do if select parse raise signal …`	`rexxKeyword rexxKeywordStatements`	`KEYWORD_INSTRUCTIONS`
Directive	`::class ::method ::routine ::requires`	`rexxDirective`	`DIRECTIVE_CLASS`
Built-in classes	`.Array .StringTable`	`rexxBuiltinClass`	`BUILTIN_CLASS_MODE`
Environment symbols	`.environment .syscargs`	`rexxSpecialVariable`	`BUILTIN_CLASS_MODE`
…	`...`	`...`	`...`

<https://gitlab.com/dylwi/oorexx-syntax-highlighting>

Example: Static with Lexer Tokens

static_lexer.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    … cut …
    <link rel="stylesheet" href="common.css" />
</head>
<body>

<pre class="programlisting" data-filename="example.rexx">
/* Simple ooRexx example to test HighlighJS */
TextArr = .array~of("Test1", 'Test2', Test3)
… cut …
</pre>

<div class="section">
    <h2>Lexer tokens</h2>
    <table>
        <thead><tr><th>#</th><th>type</th><th>value</th></tr></thead>
        <tbody id="tokens"></tbody>
    </table>
</div>

<script src="theme-switcher.js"></script>
<script src="hljs/highlight.min.js"></script>
<script src="hljs/languages/rexx.js"></script>
<script src="code-box.js"></script>
<script src="static.js"></script>
<script src="tokeniser.js"></script>
<script src="token-table.js"></script>
</body>
</html>
```

Code

Output:

copy theme-switcher

example.rexx idea

```
/* Simple ooRexx example to test HighlighJS */
TextArr = .array~of("Test1", 'Test2', Test3)

do i = 1 to TextArr~items
  say "Line" i || ":" TextArr[i]
end

TestClass = .TestClass~new(8)
say TestClass~increase
say TestClass~Increase
say TestClass~Decrease

::Class TestClass Public
  ::Method Init
    Expose Number
    Use arg Number

    if Number = "NUMBER" then Number = 1

  ::Method Increase
    Expose Number
    Number = Number + 1
    return Number

  ::Method Decrease
    Expose Number
    Number -=1
    return Number
```

Lexer tokens

#	TYPE	VALUE
1	hljs-comment	/* Simple ooRexx example to test HighlighJS */
2	(text)	␣TextArr =

<https://gitlab.com/dylwi/oorexx-syntax-highlighting>

Till Winkler

Use Case: Fancify Online Manual (1 of 2)

- Add `hljs` folder to `Common_Content`
- Inject dependencies in every *.html which contains:

```
<pre class="programlisting">
```

  - Inject:

```
…cut…
        <link rel="stylesheet" href="Common_Content/hljs/hljs.css" />
    </head>
…cut…
        <script src="Common_Content/hljs/highlight.min.js"></script>
        <script src="Common_Content/hljs/languages/rexx.js"></script>
        <script src="Common_Content/hljs/theme-switcher.js"></script>
        <script src="Common_Content/hljs/code-box.js"></script>
        <script src="Common_Content/hljs/static.js"></script>
    </body>
</html>
```

```
hljs
├── languages
├── styles
├── code-box.js
├── highlight.min.js
├── hljs.css
├── static.js
└── theme-switcher.js
```

Future: Extend existing CSS

Future: More testing

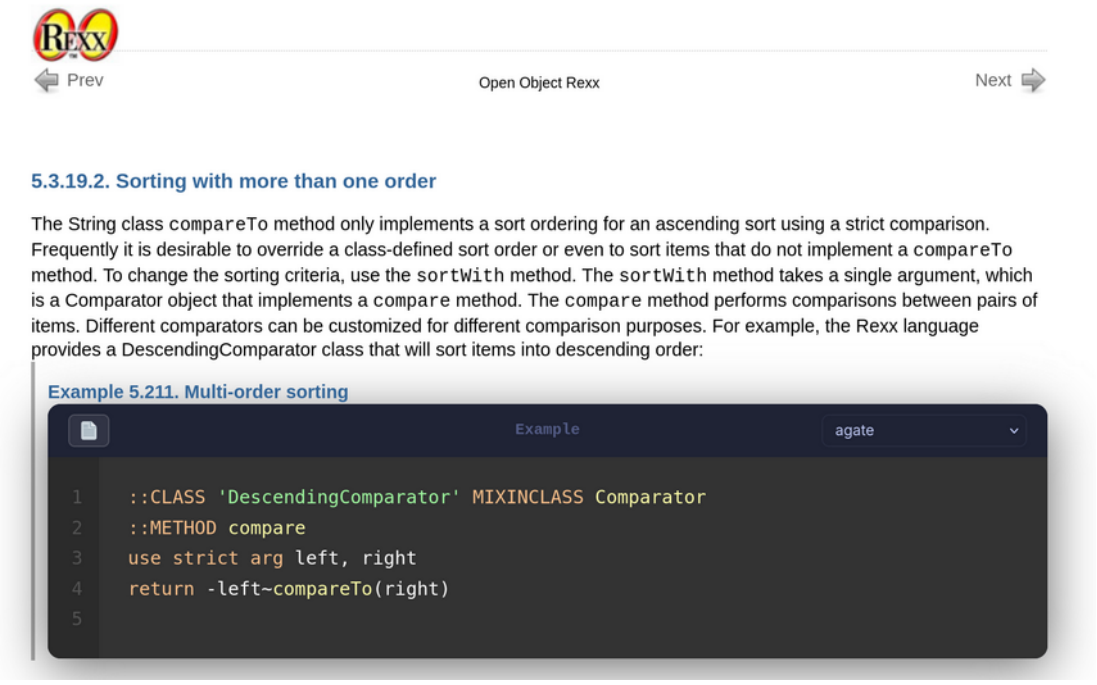
Future: Refactoring

<https://gitlab.com/dylwi/oorexx-syntax-highlighting>

Use Case: Fancify Online Manual (2 of 2)

- Script: ExtendManual.rexx (still some bugs)

Output:



<https://gitlab.com/dylwi/oorexx-syntax-highlighting>

Example: Online Editor

editor.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>ooRexx inline editor</title>
    <link rel="stylesheet" href="common.css" />
</head>
<body>


<div class="window" id="window"></div>

<script src="hljs/highlight.min.js"></script>
<script src="hljs/languages/rexx.js"></script>
<script src="theme-switcher.js"></script>
<script src="code-box.js"></script>
<script src="editor-box.js"></script>
<script src="seed.js"></script>
</body>
</html>
```

HighligthJS

Editor

Output:

<https://gitlab.com/dylwi/oorexx-syntax-highlighting>

Use Case: Online Editor (Future Perspective)

- Enabling the execution of ooRexx code from a web browser within a VM
- Two approaches I would like to explore next year:
  - A) One-shot VMs (Kernel isolation)
    - One Tiny Core Linux snapshot per execution
      - e.g. Firecracker (microVM runtime)
      - Considerations: Latency, Operational complexity
    - One RISC-V ELF binary
      - e.g. Libriscv emulator
      - Considerations: No RISC-V port of ooRexx, Only a process emulator
  - B) Linux process wrapping (Kernel sharing)
    - Would enable to use and share larger Linux distributions for code execution
    - e.g.: nsjail
      - wraps process in a combination of Linux primitives (Namespaces, seccomp-bpf, cgroups v1/v2, rlimits)

<https://gitlab.com/dylwi/oorexx-syntax-highlighting>

Pygments Library

- Python library for syntax highlighting
  - Regex-based language lexers
    - Rexx-support: <https://pygments.org/docs/lexers/#pygments.lexers.scripting.RexxLexer>
    - No ooRexx support yet
      - Seymour J. Metz is working on that:
        - <https://github.com/shmuelmetz/Pygments-Extensions>
- **Used in**: Reddit, Jupyter notebooks, LaTeX via minted package, GitHub, Wikipedia, ...
  - <https://pygments.org/languages/>

Pygments Library
Mapping between Vim, HighlightJS and Pygments (Excerpt)

Category	OoRexx Element	Vim Token	HighlightJS Scope	Pygments Token
Block comment	`/* … */`	`rexxComment`	`BLOCK_COMMENT`	`Comment.Multiline`
Line comment	`-- …`	`rexxLineComment`	`LINE_COMMENT`	`Comment.Single`
Comment error	`*/`			
Double-quoted	`"…"`	`rexxString`	`STRING`	`String.Double`
Double-quoted	`'…'`	`rexxString`	`STRING`	`String.Single`
::RESOURCE body	`Multi-line text until ::`	`rexxResourceBody`	`RESOURCE_REGION`	`String.Other`
Core instructions	`say do if select parse raise signal …`	`rexxKeyword rexxKeywordStatements`	`KEYWORD_INSTRUCTIONS`	`Keyword`
Directive	`::class ::method ::routine ::requires`	`rexxDirective`	`DIRECTIVE_CLASS`	`Keyword.Declaration`
Built-in classes	`.Array .StringTable`	`rexxBuiltinClass`	`BUILTIN_CLASS_MODE`	`Name.Builtin`
Environment symbols	`.environment .syscargs`	`rexxSpecialVariable`	`BUILTIN_CLASS_MODE`	`Name.Variable.Global`
…	`...`	`...`	`...`	

<https://gitlab.com/dylwi/oorexx-syntax-highlighting>

Pygments Lexer

- 1) Get Pygments Library
  - ArchLinux: `yay -S python-pygments`
- 2) Get test script & Lexer
- 3) Run script

```
pygmentize -x -l oorexx.py rexxText.rexx
```

RexxText.rexx

```
/* Simple ooRexx example to test HighlighJS */
TextArr = .array~of("Test1", 'Test2', Test3)

do i = 1 to TextArr~items
  say "Line" 1 || ":" TextArr[i]
end

TestClass = .TestClass~new(8)
say TestClass~increase
say TestClass~Increase
say TestClass~Decrease

… cut …
```

Output (Terminal):

```
TextArr = .array~of("Test1", 'Test2', Test3)

do i = 1 to TextArr~items
  say "Line" 1 || ":" TextArr[i]
end

TestClass = .TestClass~new(8)
say TestClass~increase
say TestClass~Increase
say TestClass~Decrease

::Class TestClass Public
  ::Method Init
    Expose Number
    Use arg Number

    if Number = "NUMBER" then Number = 1

  ::Method Increase
    Expose Number
    Number = Number + 1
    return Number

  ::Method Decrease
    Expose Number
    Number -=1
    return Number
```

Conclusion

- Early stage: Tests & Refactoring required
  - Will be available: <https://gitlab.com/dylwi/oorexx-syntax-highlighting>
- Pushing changes to HighlightJS and Pygments would enable syntax highlighting in many products
  - e.g.: GitLab, Wikipedia, Jupyter notebooks ...
- HighlightJS could:
  - Fancify the HTML manuals
  - Could be a step towards an online REXX editor

Misc

- **Wikipedia:**
  - Considerable amount of content was removed from the **Rexx** Page (**Please check!**):
    - <https://en.wikipedia.org/wiki/Rexx>
  - Content was added to the **ooRexx** Page:
    - <https://en.wikipedia.org/wiki/Object_REXX>
- **Rexx References Project**
  - <https://gitlab.com/dylwi/rexx-references>
  - **2617 Entries** (additional 320 entries since 36th Rexx Symposium)
  - Generates overview statistics

`Statistics.html`