# Open-Source Rich Text Editor Libraries for Web and Cross-Platform Desktop

## JavaScript/TypeScript Solutions

**JavaScript offers a rich ecosystem for WYSIWYG editors**, with many open-source libraries ranging from full-featured editors to low-level frameworks. These can run in the browser (and by extension in desktop apps via Electron or similar wrappers). The table below compares prominent JS/TS options:

| **Library** | **Description & Features** | **DOCX Support** | **Platforms** | **License / Notes** |
| --- | --- | --- | --- | --- |
| **CKEditor 5** | Full-featured web WYSIWYG editor (contenteditable) with extensive plugins (images, tables, collaboration, etc.). Resembles MS Word interface. | *Import/Export*: Official plugins for DOCX (cloud-based) available with paid plans[[1]](https://ckeditor.com/docs/ckeditor5/latest/features/converters/export-word.html#:~:text=The%20export%20to%20Word%20feature,file%20directly%20from%20the%20editor)[[2]](https://ckeditor.com/docs/ckeditor5/latest/features/converters/export-word.html#:~:text=Copy%20link). No free built-in DOCX support. | Web (browser); desktop via Electron wrapper. | GPL 2+ (open-source for OSS use)[[3]](https://stackoverflow.com/questions/75397333/do-i-need-a-license-to-use-ckeditor5-to-build-internal-admin-applications#:~:text=,same%20Open%20Source%20license). *Note:* Premium features (e.g. Word export) require commercial license[[1]](https://ckeditor.com/docs/ckeditor5/latest/features/converters/export-word.html#:~:text=The%20export%20to%20Word%20feature,file%20directly%20from%20the%20editor). |
| **TinyMCE** | Mature, widely-used web editor with a UI toolbar, formatting buttons, etc. Highly customizable; supports images, media, tables, etc. | *Import/Export:* No native DOCX; content is HTML-based (users can paste from Word, or convert HTML to DOCX via separate tools). | Web (all modern browsers)[[4]](https://quilljs.com/#:~:text=Cross%20Platform); can integrate in Electron for desktop. | GPL 2+ (core editor is open-source)[[5]](https://en.wikipedia.org/wiki/TinyMCE#:~:text=TinyMCE%20is%20an%20online%20rich,HTML%20elements%2C%20into%20editor%20instances). Free core with paid add-ons[[5]](https://en.wikipedia.org/wiki/TinyMCE#:~:text=TinyMCE%20is%20an%20online%20rich,HTML%20elements%2C%20into%20editor%20instances). |
| **Quill** | Lightweight, API-driven rich text editor producing JSON (Delta) output. Supports basic formatting (bold, italic, lists, headings), images, and embeds via extensions[[6]](https://quilljs.com/#:~:text=An%20API%20Driven%20Rich%20Text,Editor). Designed for consistent cross-platform output[[7]](https://quilljs.com/#:~:text=output). | *Import/Export:* No built-in DOCX. Typically integrates by converting Quill’s HTML output to DOCX via external libraries (e.g. *Mammoth* for import). | Web (desktop & mobile browsers)[[4]](https://quilljs.com/#:~:text=Cross%20Platform); often embedded in Electron for desktop apps. | MIT (completely free/open-source)[[8]](https://github.com/odygrd/quill#:~:text=GitHub%20github,Your%20use%20of%20the). Widely used in web apps[[9]](https://quilljs.com/#:~:text=Fits%20Like%20a%20Glove). |
| **Tiptap** (built on ProseMirror) | Headless editor framework for rich text, used to build Notion-like editors. Supports bold/italic, lists, tables, images, etc., with optional UI components. Highly extensible (custom nodes/marks)[[10]](https://tiptap.dev/product/editor#:~:text=Powered%20by%20ProseMirror). | *Import/Export:* Offers DOCX import/export **via extensions** that use a cloud/REST service[[11]](https://tiptap.dev/product/conversion#:~:text=Tiptap%20makes%20it%20simple%20to,formatting%20intact%20in%20your%20editor)[[12]](https://tiptap.dev/product/conversion#:~:text=Ensure%20your%20users%20can%20seamlessly,formatting%20for%20a%20smooth%20experience) (open-source core, but DOCX conversion is a paid/hosted feature). | Web (framework-agnostic: works with React, Vue, etc.)[[13]](https://tiptap.dev/product/editor#:~:text=Integrate%20with%20any%20framework); can run in Electron for desktop. | MIT (core editor is open-source)[[14]](https://tiptap.dev/product/editor#:~:text=,free). *Note:* Advanced features (e.g. DOCX/PDF conversion) require a Tiptap Pro account or self-hosted service[[14]](https://tiptap.dev/product/editor#:~:text=,free). |
| **Slate** | Developer-oriented framework for building rich text editors in React. Provides complete customization of schema, rendering, and commands. Supports rich text formatting (Bold, links, etc.) by design of your implementation. | *Import/Export:* No native DOCX handling. Community tools exist (e.g. slate-docx-deserializer) to import Word content[[15]](https://www.npmjs.com/package/slate-docx-deserializer#:~:text=slate,npm%20i). Otherwise requires converting DOCX to HTML before loading. | Web (primarily React apps); can be used in Electron for desktop. | MIT (open-source). High flexibility but requires more effort to build UI. |
| **ProseMirror** | Low-level rich text editing toolkit. Provides an editor state, schema, and plugin system to create custom WYSIWYG editors. Powers Tiptap and others. Supports collaborative editing, custom schema for bold, italic, etc. | *Import/Export:* No built-in DOCX. Typically would convert DOCX to HTML and then into ProseMirror schema. (ProseMirror focuses on content JSON/HTML structure, not file formats.) | Web (DOM-based); usable in any JS environment (can integrate in browser or Electron). | MIT/Apache 2.0 (open-source)[[16]](https://discuss.prosemirror.net/t/meet-prosemirror-utils-package/1289#:~:text=One%20question%20I%20have%20about,The%20license%20itself%20is%20fine). Highly extensible but steep learning curve; used as a foundation for other editors. |
| **Lexical** (Facebook/Meta) | Modern extensible text editor framework emphasizing performance and accessibility[[17]](https://www.reddit.com/r/reactjs/comments/u4zv7p/lexical_an_extensible_text_editor_framework/#:~:text=Lexical%2C%20an%20extensible%20text%20editor,on%20reliability%2C%20accessibility%2C%20and%20performance)[[18]](https://github.com/facebook/lexical#:~:text=Lexical%20is%20an%20extensible%20JavaScript,scale%20in%20size%20and%20functionality). Provides plugins for rich text, code, lists, etc., with a React integration. | *Import/Export:* No native DOCX support. Meant to handle editor state and HTML; external conversion needed for .docx. | Web (browser; framework-agnostic with bindings for React, etc.)[[19]](https://github.com/facebook/lexical#:~:text=Getting%20started%20with%20React). Can be run in Electron. | MIT (Meta’s open-source project). Fast and accessibility-focused; relatively new community. |

**Notes:** All the above support **basic rich text formatting** (bold, italic, underline, headings, lists, links, etc.) as well as **undo/redo** and other editing basics by default or via plugins. Many allow **pasting from Word**, which converts some Word formatting to HTML. However, **true DOCX file integration is not typically built-in** to these editors. For DOCX import/export, developers often use a converter library or service in addition to the editor.

### Document Format Libraries (JS)

* **Mammoth.js** – A JavaScript library to **convert Word .docx files to HTML or Markdown** for display[[20]](https://github.com/mwilliamson/mammoth.js#:~:text=mwilliamson%2Fmammoth,and%20convert%20them%20to%20HTML). It preserves formatting like headings, bold/italic, lists, etc., aiming for clean HTML[[21]](https://mike.zwobble.org/projects/mammoth/#:~:text=Mammoth%20,HTML%20by%20using%20semantic%20information). *Use case:* import a DOCX into an editor by converting it to HTML first. (MIT License)
* **docx.js** (npm package “docx”) – Library for **generating and modifying .docx files** in JS/TS (works in Node or browser)[[22]](https://github.com/dolanmiu/docx#:~:text=Easily%20generate%20and%20modify%20,Node%20and%20on%20the%20Browser). Provides a declarative API to create Word documents with text, tables, etc., programmatically. *Use case:* export editor content to a Word file. (MIT License)
* **Docxtemplater** – Templating library to **fill Word document templates** (e.g. replace placeholders with data). Core is open-source (MIT/GPLv3) with basic features[[23]](https://docxtemplater.com/faq/#:~:text=FAQ%20,Advanced%20features%20require%20paid%20modules), but advanced modules (for images, HTML, etc.) are paid[[24]](https://docxtemplater.com/#:~:text=,the%20HTML%20module%20allows)[[25]](https://docxtemplater.com/faq/#:~:text=Can%20I%20use%20Docxtemplater%20for,Advanced%20features%20require%20paid%20modules).
* **Other tools:** *OnlyOffice Docs* and *Collabora Online* deserve mention as full **online office suites** (AGPL license). They provide complete Word-like editors in the browser (with high fidelity DOCX support) but are heavyweight solutions that run in iframes or separate services[[26]](https://github.com/ONLYOFFICE#:~:text=ONLYOFFICE%20Docs%20is%20a%20free,%E2%80%A6)[[27]](https://github.com/ONLYOFFICE#:~:text=License), not simple drop-in libraries.

## Flutter/Dart Solutions

Flutter has several open-source packages for rich text editing, suitable for mobile and desktop apps (and even web with Flutter Web). These range from high-level ready-to-use editors to lower-level frameworks. Below is a comparison of notable Flutter packages:

| **Package** | **Description & Features** | **DOCX Support** | **Platforms** | **License / Notes** |
| --- | --- | --- | --- | --- |
| **Flutter Quill** (flutter\_quill) | Rich text editor widget based on Quill JS’s data format (Delta). Provides a configurable toolbar and editing area. Supports **basic formatting** (bold, italic, underline, lists, headings) and embedding images/videos. Highly customizable and extensible (you can add custom formats or themes)[[28]](https://www.walturn.com/insights/a-comparison-between-various-rich-text-editors-for-flutter#:~:text=Quill%20is%20an%20open%20source,key%20features%20that%20Quill%20offers)[[29]](https://www.walturn.com/insights/a-comparison-between-various-rich-text-editors-for-flutter#:~:text=Extensibility%3A%20The%20modular%20architecture%20allows,custom%20formats%2C%20modules%2C%20and%20themes). | *Import/Export:* No direct DOCX handling. Content is kept in Delta/HTML; for DOCX, one would convert the data (e.g. via an API or package like docx\_template for export). | Flutter apps: Android, iOS, Web, Windows, macOS, Linux (Flutter Web support requires using the web renderer). | MIT License. Active community (forked from the older Zefyr editor). |
| **Fleather** | A modern rich text editor for Flutter with broad platform support. Supports **multi-platform** editing (Android/iOS, Windows, macOS, Linux, web)[[30]](https://www.walturn.com/insights/a-comparison-between-various-rich-text-editors-for-flutter#:~:text=Fleather%20is%20a%20modern%20rich,key%20features%20that%20Fleather%20offers). Provides core formatting like bold, italics, strikethrough, lists, and can embed images/links. Also capable of **encoding/decoding to Markdown or HTML** for compatibility with web formats[[31]](https://www.walturn.com/insights/a-comparison-between-various-rich-text-editors-for-flutter#:~:text=elements%20or%20inline%20elements%20in,their%20applications). | *Import/Export:* No native DOCX; uses its own document model (similar to Quill’s). Developers can convert Fleather’s content to HTML/Markdown and then use a DOCX converter if needed. | Flutter: Android, iOS, web, desktop (as noted, supports all Flutter targets)[[30]](https://www.walturn.com/insights/a-comparison-between-various-rich-text-editors-for-flutter#:~:text=Fleather%20is%20a%20modern%20rich,key%20features%20that%20Fleather%20offers). | BSD-3-Clause (open-source). Evolved from the Zefyr project; emphasis on clean design and web format interoperability[[31]](https://www.walturn.com/insights/a-comparison-between-various-rich-text-editors-for-flutter#:~:text=elements%20or%20inline%20elements%20in,their%20applications). |
| **Super Editor** (super\_editor) | An open-source toolkit by Superlist for building rich text editors in Flutter. Provides a ready-to-use **SuperEditor widget** with default behaviors, and lower-level tools for customization[[32]](https://www.walturn.com/insights/a-comparison-between-various-rich-text-editors-for-flutter#:~:text=Text%20Editing%20Capabilities%3A%20The%20edit,ordered%20and%20unordered%20lists). Supports typical rich text features (bold, italics, headings, lists) and embedding images or links[[32]](https://www.walturn.com/insights/a-comparison-between-various-rich-text-editors-for-flutter#:~:text=Text%20Editing%20Capabilities%3A%20The%20edit,ordered%20and%20unordered%20lists). Highly configurable; comes with example implementations to jump-start development[[33]](https://www.walturn.com/insights/a-comparison-between-various-rich-text-editors-for-flutter#:~:text=that%20you%20can%20fine,match%20your%20application%E2%80%99s%20design%20requirements). | *Import/Export:* No built-in DOCX support. Uses its own document model. Content could be converted to JSON or Markdown/HTML, then to DOCX via an external library if required. | Flutter: Android, iOS, Web (in early development), desktop platforms. (It’s Flutter, so cross-platform by nature; early stages may require fixes for web)[[34]](https://www.walturn.com/insights/a-comparison-between-various-rich-text-editors-for-flutter#:~:text=dependencies%3A%20flutter%3A%20sdk%3A%20flutter). | MIT License. Still in early development (v0.x)[[35]](https://www.walturn.com/insights/a-comparison-between-various-rich-text-editors-for-flutter#:~:text=super_editor%3A%20), but offers deep customization for bespoke editors. |
| **AppFlowy Editor** (appflowy\_editor) | A new rich-text editor extracted from the AppFlowy project (Notion-like app). Supports rich text blocks, lists, checkboxes, embeds, etc., with a focus on **customizable block components and themes**[[36]](https://github.com/AppFlowy-IO/appflowy-editor#:~:text=Key%20Features)[[37]](https://github.com/AppFlowy-IO/appflowy-editor#:~:text=,by%20AppFlowy%27s%20core%20team%20and). Designed for extensibility (developers can add custom block types or shortcut behaviors). | *Import/Export:* No direct DOCX support noted. Focuses on JSON/Markdown/Delta as input-export formats (it can initialize from Markdown or Quill Delta)[[38]](https://github.com/AppFlowy-IO/appflowy-editor#:~:text=final%20editorState%20%3D%20EditorState,editorState%3A%20editorState%2C). Would require external conversion to handle DOCX files. | Flutter: Android, iOS, desktop, Web. (AppFlowy targets mobile/desktop; Flutter Web should work for viewing, though editing on web might be experimental.) | Dual-licensed AGPL 3.0 + MPL 2.0[[39]](https://github.com/AppFlowy-IO/appflowy-editor#:~:text=License). Open-source with a growing community. Suitable if building Notion-like rich text experiences. |
| **HTML Editor Enhanced** (html\_editor\_enhanced) | A Flutter plugin that embeds a mobile-friendly **HTML/JavaScript editor** (Summernote) in a WebView. Supports all standard rich text formatting via the HTML editor and returns HTML content. | *Import/Export:* No native DOCX; content is HTML. You could convert the HTML to DOCX using JS libraries or server-side services. | Flutter: Android, iOS, Web. (Uses WebView on mobile, which is not available on Flutter Web builds.) | MIT License (Summernote is MIT). This is essentially a wrapper around a web editor within Flutter. |

**Notes:** Flutter’s rich text packages all support **basic text styling** like bold, italics, underlines, headings, lists, text alignment, etc. Many also support **inserted media** (images, videos) and hyperlinks. However, **none have native DOCX import/export** capabilities – typically you would need to convert a DOCX file to an HTML or Delta format to load into the editor, or use a package like those below to generate/save DOCX:

* **docx\_template (Dart)** – Library to **generate .docx documents** using templates in Flutter/Dart apps. You design a DOCX with placeholders, and this library fills them (e.g. for reports). Useful for exporting content to DOCX. (MIT License)[[40]](https://www.reddit.com/r/dotnet/comments/9wq105/is_there_a_docx_template_library_for_net/#:~:text=Reddit%20www,NET).
* **docx\_viewer (Flutter)** – Widget for **viewing DOCX files** by rendering text and basic formatting in-app[[41]](https://www.reddit.com/r/FlutterDev/comments/1gnyrhw/published_the_new_flutter_pacakge_to_render_the/#:~:text=Reddit%20www,who%20need%20DOCX%20viewing). It’s read-only (not an editor), but can display the content of Word files on Android/iOS. This helps if you only need to show a DOCX’s text.
* **PDF/Word via platform** – Since direct DOCX editing in Flutter is limited, a common approach is to convert DOCX to PDF (using packages or cloud APIs) for viewing, or rely on a platform channel to call native code (e.g. MS Office on device, or LibreOffice on desktop) to handle complex documents. These are outside the Flutter widget ecosystem.

## C# / .NET Solutions

In the .NET ecosystem, rich text editing solutions vary by UI framework. Traditional .NET (Windows) had built-in rich text controls, but **cross-platform rich text editors are not yet first-class** in .NET MAUI or .NET Core. Developers often embed a web-based editor or use third-party controls. Below we outline both UI components and libraries for .NET:

### Rich Text Editor UI Components in .NET

* **WinForms RichTextBox / WPF RichTextBox** – Built-in controls on Windows for editable rich text. They support basic formatting (font styles, colors, alignment, lists) and can load/save RTF content. However, they do **not support DOCX natively** – only RTF or plain text. (On WPF, the RichTextBox uses a FlowDocument internally, which can be saved to XAML or RTF; importing a DOCX would require converting it to FlowDocument via Word automation or OpenXML, which isn’t straightforward). *Platforms:* Windows only. *License:* Closed-source (part of .NET Framework) but free to use on Windows.
* **Blazor-based Editors** – With Blazor (WebAssembly or Blazor Hybrid), one can use JavaScript editors in .NET apps. For example, **Blazored TextEditor** is an open-source Blazor component that wraps the Quill JS editor (MIT licensed)[[42]](https://github.com/Blazored/TextEditor#:~:text=Rich%20text%20editor%20for%20Blazor,Uses%20Quill%20JS). This allows rich text editing in web or in a Blazor Hybrid desktop app using web tech. *Platforms:* Web (Blazor WASM) or cross-platform desktop via .NET MAUI Blazor Hybrid. *Note:* The heavy lifting is done by the JS library (Quill, TinyMCE, etc.), so DOCX support would still require a converter as in JavaScript solutions.
* **No native MAUI editor (yet)** – .NET MAUI (for cross-platform UI) does *not* include a RichTextEditor control as of .NET 7/8. Developers resort to embedding a **WebView with an open-source web editor** to achieve rich text functionality[[43]](https://stackoverflow.com/questions/75565199/net-maui-rich-text#:~:text=1). This essentially loads an HTML/JS editor within the app. Microsoft’s MAUI team has an open issue for a RichTextEditor control[[44]](https://stackoverflow.com/questions/75565199/net-maui-rich-text#:~:text=For%20now%2C%20the%20MAUI%20does,opensource%20web%20RichTextEditor%20in%20blazor), but currently one must use third-party solutions (Syncfusion and Telerik offer paid controls for MAUI, or use the Blazor approach above).
* **Avalonia/AvalonEdit** – *AvaloniaEdit* is an open-source text editor control for Avalonia (cross-platform WPF-like UI). It’s a port of **AvalonEdit** (from the SharpDevelop project)[[45]](https://github.com/AvaloniaUI/AvaloniaEdit#:~:text=AvaloniaUI%2FAvaloniaEdit%3A%20Avalonia,highlighting%20using%20TextMate%20grammars). These are primarily designed for code editing (syntax highlighting, etc.) rather than WYSIWYG word processing. They support styled text display and editing, but not high-level constructs like embedded images or DOCX import. *Platforms:* AvaloniaEdit runs on Windows/Linux/macOS; AvalonEdit on Windows (WPF). *License:* MIT. They are useful for text and code editing with coloring, but would require substantial work to turn into a Word-like editor.

**Note on DOCX in UI:** Unlike web editors, .NET UI controls don’t handle DOCX files directly (except by leveraging Office APIs). If you need to **open a Word document in a .NET app’s editor**, one approach is converting the DOCX to HTML or RTF, then loading it. For example, a DOCX could be converted to HTML (using a library or service) and displayed in a WebView or converted to RTF via Open XML and loaded into RichTextBox. There is no out-of-the-box solution in .NET for full-fidelity DOCX editing without using Microsoft Word itself or heavy libraries.

### Document Format Libraries (C#/.NET)

* **Open XML SDK** – Official Microsoft-provided library for **manipulating Office OpenXML files (DOCX, XLSX, PPTX)** at a low level. Allows creating, reading, and editing DOCX files via code (e.g. inserting paragraphs, replacing text, reading content)[[46]](https://github.com/dotnet/Open-XML-SDK#:~:text=The%20Open%20XML%20SDK%20is,level%20abstractions%20or%20productivity%20tools). It works with the document’s XML structure (requires knowledge of Word’s OpenXML schema). *Use case:* Import/export: e.g. read a DOCX, extract text or convert to HTML, or generate a DOCX from scratch. *License:* MIT (open-source on GitHub by Microsoft)[[47]](https://github.com/dotnet/Open-XML-SDK#:~:text=,Security). No UI – this is back-end only.
* **NPOI** – .NET port of Apache POI, used for reading/writing Office files (Word, Excel, etc.). **Supports DOC and DOCX** for text extraction and creation[[48]](https://github.com/nissl-lab/npoi#:~:text=GitHub%20github,Office%202003%2F2007%20files%20very%20easily)[[49]](https://www.reddit.com/r/dotnet/comments/12lvfzy/is_it_possible_to_use_microsoft_word_features/#:~:text=NPOI%20). Easier for basic tasks than OpenXML SDK in some cases (handles older binary .doc as well). *License:* Apache 2.0 (open-source)[[49]](https://www.reddit.com/r/dotnet/comments/12lvfzy/is_it_possible_to_use_microsoft_word_features/#:~:text=NPOI%20). Often used server-side to generate .docx or parse them without needing Office installed.
* **DocX (Xceed)** – A high-level .NET library for Word docs with a friendly API (e.g. document.InsertParagraph("Hello")). Supports formatting, images, tables in DOCX. *License:* **Open-source for non-commercial use**[[50]](https://www.reddit.com/r/dotnet/comments/12lvfzy/is_it_possible_to_use_microsoft_word_features/#:~:text=DocX%20,non%20commercial%20use) (Xceed Community License) – meaning it’s free for OSS or internal use, but requires a paid license for commercial redistribution. It’s popular for quick DOCX generation/manipulation in C# if the licensing fits.
* **Others:** *Gemini* and *MigraDoc* are open libraries to create documents (often targeting PDF but can sometimes output RTF/DocX via Word’s DLLs). Also note **Aspose.Words**, **Syncfusion DocIO**, **Spire.Doc** are powerful .NET Word-processing libraries that support nearly the entire DOCX spec, but they are commercial (not open-source). For open-source needs, Open XML SDK or NPOI are the primary choices for handling Word files programmatically.

**Summary of .NET Approach:** For a **cross-platform text editor in .NET**, a common strategy is to embed a web-based editor (like those from the JS section) inside a Blazor or WebView control – this leverages the rich capabilities of those editors and their cross-platform nature[[43]](https://stackoverflow.com/questions/75565199/net-maui-rich-text#:~:text=1). Meanwhile, use the **Open XML SDK or NPOI** on the back-end to import/export DOCX data (e.g., converting the editor’s HTML to DOCX and vice versa). This combination covers rich text UI and Microsoft Word format support using entirely open-source components.

**Sources:**

* Quill JS – “Quill is a free, open source WYSIWYG editor built for the modern web.”[[8]](https://github.com/odygrd/quill#:~:text=GitHub%20github,Your%20use%20of%20the)
* Tiptap (ProseMirror-based) – MIT licensed open-source core[[14]](https://tiptap.dev/product/editor#:~:text=,free); DOCX import/export via cloud service[[11]](https://tiptap.dev/product/conversion#:~:text=Tiptap%20makes%20it%20simple%20to,formatting%20intact%20in%20your%20editor)[[12]](https://tiptap.dev/product/conversion#:~:text=Ensure%20your%20users%20can%20seamlessly,formatting%20for%20a%20smooth%20experience)
* CKEditor 5 – Dual-licensed (GPL2+ for OSS); offers export to Word as a premium feature[[1]](https://ckeditor.com/docs/ckeditor5/latest/features/converters/export-word.html#:~:text=The%20export%20to%20Word%20feature,file%20directly%20from%20the%20editor)[[2]](https://ckeditor.com/docs/ckeditor5/latest/features/converters/export-word.html#:~:text=Copy%20link)
* TinyMCE – Open-source under GPL 2.0+ (free core editor)[[5]](https://en.wikipedia.org/wiki/TinyMCE#:~:text=TinyMCE%20is%20an%20online%20rich,HTML%20elements%2C%20into%20editor%20instances)
* Mammoth.js – Converts .docx files to simple HTML/Markdown[[20]](https://github.com/mwilliamson/mammoth.js#:~:text=mwilliamson%2Fmammoth,and%20convert%20them%20to%20HTML)
* Open XML SDK (Microsoft) – .NET library for Word/Excel/PowerPoint formats (OpenXML)[[46]](https://github.com/dotnet/Open-XML-SDK#:~:text=The%20Open%20XML%20SDK%20is,level%20abstractions%20or%20productivity%20tools)
* Reddit (r/dotnet) – Notes on .NET Word libs: *“DocX (by Xceed, open source, non commercial use)… NPOI (Open source, MIT)”*[[50]](https://www.reddit.com/r/dotnet/comments/12lvfzy/is_it_possible_to_use_microsoft_word_features/#:~:text=DocX%20,non%20commercial%20use)[[49]](https://www.reddit.com/r/dotnet/comments/12lvfzy/is_it_possible_to_use_microsoft_word_features/#:~:text=NPOI%20)
* Walturn.com – *Comparison of Flutter editors* (Fleather, SuperEditor, Quill features)[[51]](https://www.walturn.com/insights/a-comparison-between-various-rich-text-editors-for-flutter#:~:text=Fleather%2C%20Super%20Editor%2C%20and%20Quill,align%20with%20specific%20application%20needs)[[30]](https://www.walturn.com/insights/a-comparison-between-various-rich-text-editors-for-flutter#:~:text=Fleather%20is%20a%20modern%20rich,key%20features%20that%20Fleather%20offers)[[32]](https://www.walturn.com/insights/a-comparison-between-various-rich-text-editors-for-flutter#:~:text=Text%20Editing%20Capabilities%3A%20The%20edit,ordered%20and%20unordered%20lists)
* StackOverflow – On .NET MAUI: *“MAUI does not have a native rich text editor… use WebView with an opensource web editor”*[[43]](https://stackoverflow.com/questions/75565199/net-maui-rich-text#:~:text=1)

[[1]](https://ckeditor.com/docs/ckeditor5/latest/features/converters/export-word.html#:~:text=The%20export%20to%20Word%20feature,file%20directly%20from%20the%20editor) [[2]](https://ckeditor.com/docs/ckeditor5/latest/features/converters/export-word.html#:~:text=Copy%20link) Export to Word | CKEditor 5 Documentation

<https://ckeditor.com/docs/ckeditor5/latest/features/converters/export-word.html>

[[3]](https://stackoverflow.com/questions/75397333/do-i-need-a-license-to-use-ckeditor5-to-build-internal-admin-applications#:~:text=,same%20Open%20Source%20license) Do I need a license to use CKEditor5 to build internal admin ...

<https://stackoverflow.com/questions/75397333/do-i-need-a-license-to-use-ckeditor5-to-build-internal-admin-applications>

[[4]](https://quilljs.com/#:~:text=Cross%20Platform) [[6]](https://quilljs.com/#:~:text=An%20API%20Driven%20Rich%20Text,Editor) [[7]](https://quilljs.com/#:~:text=output) [[9]](https://quilljs.com/#:~:text=Fits%20Like%20a%20Glove) Quill - Your powerful rich text editor

<https://quilljs.com/>

[[5]](https://en.wikipedia.org/wiki/TinyMCE#:~:text=TinyMCE%20is%20an%20online%20rich,HTML%20elements%2C%20into%20editor%20instances) TinyMCE - Wikipedia

<https://en.wikipedia.org/wiki/TinyMCE>

[[8]](https://github.com/odygrd/quill#:~:text=GitHub%20github,Your%20use%20of%20the) odygrd/quill: Asynchronous Low Latency C++ Logging Library - GitHub

<https://github.com/odygrd/quill>

[[10]](https://tiptap.dev/product/editor#:~:text=Powered%20by%20ProseMirror) [[13]](https://tiptap.dev/product/editor#:~:text=Integrate%20with%20any%20framework) [[14]](https://tiptap.dev/product/editor#:~:text=,free) Tiptap Rich Text Editor - the Headless WYSIWYG Editor

<https://tiptap.dev/product/editor>

[[11]](https://tiptap.dev/product/conversion#:~:text=Tiptap%20makes%20it%20simple%20to,formatting%20intact%20in%20your%20editor) [[12]](https://tiptap.dev/product/conversion#:~:text=Ensure%20your%20users%20can%20seamlessly,formatting%20for%20a%20smooth%20experience) Import & Export DOCX with Tiptap

<https://tiptap.dev/product/conversion>

[[15]](https://www.npmjs.com/package/slate-docx-deserializer#:~:text=slate,npm%20i) slate-docx-deserializer - NPM

<https://www.npmjs.com/package/slate-docx-deserializer>

[[16]](https://discuss.prosemirror.net/t/meet-prosemirror-utils-package/1289#:~:text=One%20question%20I%20have%20about,The%20license%20itself%20is%20fine) Meet prosemirror-utils package

<https://discuss.prosemirror.net/t/meet-prosemirror-utils-package/1289>

[[17]](https://www.reddit.com/r/reactjs/comments/u4zv7p/lexical_an_extensible_text_editor_framework/#:~:text=Lexical%2C%20an%20extensible%20text%20editor,on%20reliability%2C%20accessibility%2C%20and%20performance) Lexical, an extensible text editor framework : r/reactjs - Reddit

<https://www.reddit.com/r/reactjs/comments/u4zv7p/lexical_an_extensible_text_editor_framework/>

[[18]](https://github.com/facebook/lexical#:~:text=Lexical%20is%20an%20extensible%20JavaScript,scale%20in%20size%20and%20functionality) [[19]](https://github.com/facebook/lexical#:~:text=Getting%20started%20with%20React) GitHub - facebook/lexical: Lexical is an extensible text editor framework that provides excellent reliability, accessibility and performance.

<https://github.com/facebook/lexical>

[[20]](https://github.com/mwilliamson/mammoth.js#:~:text=mwilliamson%2Fmammoth,and%20convert%20them%20to%20HTML) mwilliamson/mammoth.js: Convert Word documents (.docx files) to ...

<https://github.com/mwilliamson/mammoth.js>

[[21]](https://mike.zwobble.org/projects/mammoth/#:~:text=Mammoth%20,HTML%20by%20using%20semantic%20information) Mammoth - Mike's corner of the web

<https://mike.zwobble.org/projects/mammoth/>

[[22]](https://github.com/dolanmiu/docx#:~:text=Easily%20generate%20and%20modify%20,Node%20and%20on%20the%20Browser) Easily generate and modify .docx files with JS/TS with a ... - GitHub

<https://github.com/dolanmiu/docx>

[[23]](https://docxtemplater.com/faq/#:~:text=FAQ%20,Advanced%20features%20require%20paid%20modules) [[25]](https://docxtemplater.com/faq/#:~:text=Can%20I%20use%20Docxtemplater%20for,Advanced%20features%20require%20paid%20modules) FAQ | docxtemplater

<https://docxtemplater.com/faq/>

[[24]](https://docxtemplater.com/#:~:text=,the%20HTML%20module%20allows) Docxtemplater | Word, Powerpoint, Excel generation using ...

<https://docxtemplater.com/>

[[26]](https://github.com/ONLYOFFICE#:~:text=ONLYOFFICE%20Docs%20is%20a%20free,%E2%80%A6) [[27]](https://github.com/ONLYOFFICE#:~:text=License) ONLYOFFICE · GitHub

<https://github.com/ONLYOFFICE>

[[28]](https://www.walturn.com/insights/a-comparison-between-various-rich-text-editors-for-flutter#:~:text=Quill%20is%20an%20open%20source,key%20features%20that%20Quill%20offers) [[29]](https://www.walturn.com/insights/a-comparison-between-various-rich-text-editors-for-flutter#:~:text=Extensibility%3A%20The%20modular%20architecture%20allows,custom%20formats%2C%20modules%2C%20and%20themes) [[30]](https://www.walturn.com/insights/a-comparison-between-various-rich-text-editors-for-flutter#:~:text=Fleather%20is%20a%20modern%20rich,key%20features%20that%20Fleather%20offers) [[31]](https://www.walturn.com/insights/a-comparison-between-various-rich-text-editors-for-flutter#:~:text=elements%20or%20inline%20elements%20in,their%20applications) [[32]](https://www.walturn.com/insights/a-comparison-between-various-rich-text-editors-for-flutter#:~:text=Text%20Editing%20Capabilities%3A%20The%20edit,ordered%20and%20unordered%20lists) [[33]](https://www.walturn.com/insights/a-comparison-between-various-rich-text-editors-for-flutter#:~:text=that%20you%20can%20fine,match%20your%20application%E2%80%99s%20design%20requirements) [[34]](https://www.walturn.com/insights/a-comparison-between-various-rich-text-editors-for-flutter#:~:text=dependencies%3A%20flutter%3A%20sdk%3A%20flutter) [[35]](https://www.walturn.com/insights/a-comparison-between-various-rich-text-editors-for-flutter#:~:text=super_editor%3A%20) [[51]](https://www.walturn.com/insights/a-comparison-between-various-rich-text-editors-for-flutter#:~:text=Fleather%2C%20Super%20Editor%2C%20and%20Quill,align%20with%20specific%20application%20needs) A Comparison Between Various Rich Text Editors for Flutter

<https://www.walturn.com/insights/a-comparison-between-various-rich-text-editors-for-flutter>

[[36]](https://github.com/AppFlowy-IO/appflowy-editor#:~:text=Key%20Features) [[37]](https://github.com/AppFlowy-IO/appflowy-editor#:~:text=,by%20AppFlowy%27s%20core%20team%20and) [[38]](https://github.com/AppFlowy-IO/appflowy-editor#:~:text=final%20editorState%20%3D%20EditorState,editorState%3A%20editorState%2C) [[39]](https://github.com/AppFlowy-IO/appflowy-editor#:~:text=License) GitHub - AppFlowy-IO/appflowy-editor: A highly customizable rich-text editor for Flutter. The AppFlowy Editor project for AppFlowy and beyond.

<https://github.com/AppFlowy-IO/appflowy-editor>

[[40]](https://www.reddit.com/r/dotnet/comments/9wq105/is_there_a_docx_template_library_for_net/#:~:text=Reddit%20www,NET) Is there a docx template library for .NET? : r/dotnet - Reddit

<https://www.reddit.com/r/dotnet/comments/9wq105/is_there_a_docx_template_library_for_net/>

[[41]](https://www.reddit.com/r/FlutterDev/comments/1gnyrhw/published_the_new_flutter_pacakge_to_render_the/#:~:text=Reddit%20www,who%20need%20DOCX%20viewing) Published the New Flutter Pacakge to Render the Docx File - Reddit

<https://www.reddit.com/r/FlutterDev/comments/1gnyrhw/published_the_new_flutter_pacakge_to_render_the/>

[[42]](https://github.com/Blazored/TextEditor#:~:text=Rich%20text%20editor%20for%20Blazor,Uses%20Quill%20JS) GitHub - Blazored/TextEditor: Rich text editor for Blazor applications - Uses Quill JS

<https://github.com/Blazored/TextEditor>

[[43]](https://stackoverflow.com/questions/75565199/net-maui-rich-text#:~:text=1) [[44]](https://stackoverflow.com/questions/75565199/net-maui-rich-text#:~:text=For%20now%2C%20the%20MAUI%20does,opensource%20web%20RichTextEditor%20in%20blazor) c# - .NET MAUI - Rich Text - Stack Overflow

<https://stackoverflow.com/questions/75565199/net-maui-rich-text>

[[45]](https://github.com/AvaloniaUI/AvaloniaEdit#:~:text=AvaloniaUI%2FAvaloniaEdit%3A%20Avalonia,highlighting%20using%20TextMate%20grammars) AvaloniaUI/AvaloniaEdit: Avalonia-based text editor (port of ... - GitHub

<https://github.com/AvaloniaUI/AvaloniaEdit>

[[46]](https://github.com/dotnet/Open-XML-SDK#:~:text=The%20Open%20XML%20SDK%20is,level%20abstractions%20or%20productivity%20tools) [[47]](https://github.com/dotnet/Open-XML-SDK#:~:text=,Security) GitHub - dotnet/Open-XML-SDK: Open XML SDK by Microsoft

<https://github.com/dotnet/Open-XML-SDK>

[[48]](https://github.com/nissl-lab/npoi#:~:text=GitHub%20github,Office%202003%2F2007%20files%20very%20easily) nissl-lab/npoi: a .NET library that can read/write Office ... - GitHub

<https://github.com/nissl-lab/npoi>

[[49]](https://www.reddit.com/r/dotnet/comments/12lvfzy/is_it_possible_to_use_microsoft_word_features/#:~:text=NPOI%20) [[50]](https://www.reddit.com/r/dotnet/comments/12lvfzy/is_it_possible_to_use_microsoft_word_features/#:~:text=DocX%20,non%20commercial%20use) Is it possible to use Microsoft Word features programatically? : r/dotnet

<https://www.reddit.com/r/dotnet/comments/12lvfzy/is_it_possible_to_use_microsoft_word_features/>