

Imaginary Teleprompter

Documentation for version 2.3.3

Licensed under the GNU Free Documentation License version 1.3

Last updated: Feb 3, 2018 at 12:43:38 AM

Author: Javier Oscar Cordero Pérez

Table of Contents

About Imaginary Teleprompter.....	6
About Free Software.....	6
Versions of Imaginary Teleprompter.....	6
Online Version.....	6
Standalone Version.....	6
Mobile Versions.....	6
Specifications.....	7
System Requirements.....	7
Officially Supported Web Browsers.....	7
Officially Supported Platforms.....	8
Unofficially Supported Platforms.....	8
Linux Dependencies.....	8
Supported File Formats.....	8
Teleprompter HTML.....	8
Contribution Guidelines.....	9
Ways to Contribute.....	9
Promote the Software.....	9
Request a Feature.....	9
Report a Bug or Difficulty Using the Software.....	9
Maintain Imaginary Teleprompter on a Linux repository.....	9
Donate Hardware to Us.....	10
Become a Translator.....	10
Add a Feature or Support a Platform.....	10
How to Join our Development Team.....	11
How We Manage Copyright Under the GPL.....	11
Developer Guidelines.....	12
Building Process.....	12
For Every Platform You Must First.....	12
For Linux.....	12
For Mac OS.....	13
For Windows.....	13
Development Cycle.....	14
Release Dates.....	14
Sprints.....	14
Planning Phase.....	14
Development Phase.....	14
Testing Phase.....	15
Coding Practices.....	16
Syntax.....	16
User Manual.....	17
Installation.....	17
Windows.....	17
Versions 2.3.x and Prior.....	17
Mac OS.....	18
Linux.....	19

AppImage.....	19
Ubuntu / Debian.....	20
Fedora.....	20
Arch.....	20
Manjaro (KDE, Octopi).....	20
Editor Window.....	21
Options Header Bar.....	21
Setup Panel.....	21
Screen & Mirror Controls.....	22
Mirroring Modes.....	22
In-frame Prompter.....	24
External Prompter.....	25
Live Update Mode.....	26
Prompter Style.....	27
Color Combinations for Version 2.3.1.....	27
Custom Style.....	28
Focus Area.....	28
Speed and Acceleration.....	29
Speed.....	29
Acceleration.....	29
Speed and Acceleration Tips.....	29
Font Scale.....	29
Timer.....	30
Editor.....	31
Editor Toolbar.....	31
File Options.....	31
Clear File.....	31
Source Editor.....	32
Edit Options.....	33
Cut, Copy and Paste.....	33
Cutting and Copying.....	33
Pasting.....	33
Find and Replace.....	34
Find.....	34
Replace.....	34
Find Options.....	35
Select All.....	36
Spell Check.....	36
Insert Options.....	37
Insert Image.....	37
Insert Flash Content.....	37
Insert Table.....	38
Table Context Menu.....	39
Cell Properties.....	39
Insert Horizontal Line.....	39
Insert Symbol.....	40
Insert Formula.....	41
Font Options.....	42
Caps Control.....	42
Font Color & Highlight.....	42
About CKEditor.....	43
Anchor Flag.....	44

Font Options (continuation).....	44
Font Style.....	44
Subscript & Superscript.....	44
Remove Formatting.....	44
Bullets & Numbering.....	44
Text Orientation.....	45
LTR & RTL Paragraphs.....	45
Paragraph Styles.....	45
Font Family.....	46
Font Size.....	46
Line Height.....	46
Word & Paragraph Counter.....	46
WYSIWYG Editing View.....	47
Context Menus.....	47
Editor Status Bar.....	48
Custom Styles Administration Pane.....	49
Create a New Style.....	49
Update a Style.....	51
Set Default Style.....	52
Reorder Styles.....	52
File Options.....	54
Create a New File.....	54
Saving Files.....	55
Edit File Name.....	55
Moving Files Across Machines.....	55
On the Computer We're Copying From.....	55
On the Computer We're Copying To.....	56
Prompter Window.....	58
Input Controls.....	58
Velocity Controls.....	58
Font Scale Controls.....	58
Timer & Playback Controls.....	58
Unique Inputs.....	58
How To Use Anchor Shortcuts.....	59
Support and Services.....	60
Distribution Contract: Inclusion with Teleprompter Hardware.....	60
Support Service Contract.....	60
Annex.....	61
Changes in this Cycle.....	61
2.3.3.....	61
2.3.2.....	61
2.3.1.....	61
2.3.0.....	61
Plans for the Next Two Releases.....	62



About Imaginary Teleprompter

Teleprompter by Imaginary Films, currently known as Teleprompter by Imaginary Sense, and from next release till the end of time as Imaginary Teleprompter and sometimes simply referred to as Teleprompter, is a professional, free software, easy to use Teleprompter, that can run on many operating systems, including Linux, Mac OS, and Windows. We created it to promote the democratization of media by enabling more people to produce quality contents, and to promote the adoption of free software solutions by the media industries.

About Free Software

Free software, as defined by the Free Software Foundation implies we, the software developers, give you, the user, the right to use, modify, and distribute the software at no cost, for any legal purpose, as long as due credit is given to the developers. This is done under the General Public License version 3.0, which requires modifications to be publicly disclosed, makes the software come with no warranty, requires a copy of the license to be shared with the software and a copy of the source code (the code that makes the program) to be shared along binaries (versions of the program that computers run). See the GPL3 at <https://www.gnu.org/licenses/gpl-3.0.en.html> for details.

Versions of Imaginary Teleprompter

Online Version

The online version exists for user testing purposes and helps provide instant, online, access to high quality teleprompter software in case of production emergencies.

A lot of efforts go into making sure the online version is as productive and feature rich as the standalone version. This is not always possible because of browsers' security limitations, which should not be bypassed. Therefore we recommend using the standalone version whenever possible.

Standalone Version

These are copies of Imaginary Teleprompter you download and run from your computer. Most efforts go into developing this version. It is the fastest and most feature rich version of Teleprompter.

Standalone is derived from the online version. Both standalone and online versions are free software, under the GPL v3, and shall remain free forever. See [About Free Software](#) for details.

Mobile Versions

We have created early prototypes of Imaginary Teleprompter for mobile devices, specifically the Android and iOS platforms. These are experimental versions, they've been stripped of GPL derivative code, remain under our copyright, and haven't been publicly released. See [how we manage copyright under the GPL](#) for details.

Mobile versions are solely used by Imaginary Sense and it's clients. They're incomplete and are currently only meant only to satisfy the needs of particular users.

Specifications

System Requirements

Some of the following specifications are RAW estimates. Imaginary Teleprompter may be able to run with lower resources where indicated with an asterisk.

- CPU: We recommend having least @1.8 GHz and 4 virtual cores, like a dual core with hyper threading, such as the Intel Core 2 Duo. Teleprompter could run well with lower end CPUs but tests have not been rigorous so we can't guarantee a good experience.
- CPU (ARM): Performance is dependent on the kind of ARM and how the binary is compiled.
 - 32 bit ARMHF architecture binaries are able to provide a smooth scroll at 720p resolution on a Raspberry Pi 3.
 - 32 bit ARM7L needs higher CPU frequencies because it uses software emulation for floating point (decimal) calculations. Teleprompter animations are computed at the subpixel level, resulting on a higher CPU consumption. Using these binaries on a Raspberry Pi 3 requires a maximum resolution of 640 x 480 for a slightly jittery animation.
 - ARM64 binaries are possible but we don't have the adequate hardware to make them or test it.
- RAM: 128 Mb free, recommended; 96 Mb free, minimum necessary to run.
- Storage: 278 Mb required (128.60 Mb for Teleprompter, ~128 for dependencies, plus 32 Mb contingency space)
- Video Card: 1 to 2 screen outputs. 640 x 480 minimum resolution, no maximum resolution. Minimum 16 bit color. Intel HD Graphics Series works good with a single screen. Dedicated graphics are advised for 2 screens.
- Internet: 56k modem or higher speed internet access required to use the math equation editor and the spell check functionality.
- Peripherals: Keyboard required, mouse is necessary to fully operate the software, but for all basic features a keyboard is enough.
- Additional peripherals: Teleprompter Interfaces that operate as a mouse wheel should be compatible. No brand is officially supported as of this release. Please see [Donate hardware to us](#), to help us increase support.

Officially Supported Web Browsers

- Desktop Browsers
 - Google Chrome
 - Mozilla Firefox
 - Apple Safari

- Mobile Browsers

No mobile browsers are officially supported yet

Officially Supported Platforms

- Windows
 - Windows 7 and later (x86 and amd64 architectures)
- Mac OS
 - OS X 10.9 (amd64 architecture)
- Linux
 - Ubuntu 12.04 (i686 and amd64 architecture)
 - Arch (Antergos and Manjaro) (i686 and amd64 architecture)
 - Fedora 21 (i686 and amd64 architecture)
 - AppImage (i686 and amd64 architecture)

Unofficially Supported Platforms

- BSD
 - FreeBSD (amd64 architecture)

Linux Dependencies

Imaginary Teleprompter uses the Electron Framework to work as standalone application under Linux, Mac OS and Windows. The following the dependencies are Electron dependencies that must be satisfied to run under Linux. You don't need to install Electron as a dependency on your system because it's included with Imaginary Teleprompter's package builds.

- | | | | |
|---------------|-------------------|-------------|-------------------------|
| • C-ares | • libevent libvpx | • nss | • libappindicator-gtk2 |
| • ffmpeg | • libxslt | • re2 | • libappindicator-gtk |
| • gtk3 | • libxss | • snappy | • libappindicator-sharp |
| • http-parser | • minizip | • libnotify | |

Supported File Formats

Teleprompter exports slightly customized HTML files, we refer to these as [Teleprompter HTML](#). These files can be used from version 2.3.0 onward. Support for more formats shall be added in future releases.

Teleprompter HTML

Teleprompter's HTML files are unique in two aspects:

1. Fonts and images use relative units of measurement such as *em* and *%*.
2. They go straight to contents, containing no header or body tags as of Teleprompter version 2.3.1.

Contribution Guidelines

Ways to Contribute

- [Promote the software](#)
- [Request a feature](#)
- [Report a bug or difficulty using the software](#)
- [Maintain Imaginary Teleprompter on a Linux repository](#)
- [Donate hardware to us](#)
- [Become a translator](#)
- [Add a feature or support a platform](#)

Promote the Software

If you found Teleprompter scouting the web for a good free teleprompter software, you'll probably realize one thing: "It needs exposition". Give us a big thank you by recommending it to your peers and giving us a mention on your videos or at Twitter. You'll make our day!

Request a Feature

While we have our ideas on what Imaginary Teleprompter should be, we want to hear from you. Your need might be somebody else's need, and your idea may be easier for us to develop than you might think.

The best way to request a feature is through GitHub at:

<https://github.com/ImaginarySense/Teleprompter-Core/issues>

But you may just as well contact us through email at: support@imaginary.tech or at Facebook.

Report a Bug or Difficulty Using the Software

We take bugs and criticism to heart and so far we've fixed every single thing that's been pointed out to us (except for the layout of our website, that has design flaws but we're not going to fix them yet). If you have anything to say regarding Teleprompter, we'll be happy to attend to your feedback.

The best way to report a bug is through GitHub at:

<https://github.com/ImaginarySense/Teleprompter-Core/issues>

But you may just as well contact us through email at support@imaginary.tech, or at Facebook:

<https://www.facebook.com/imaginarytech/>

Maintain Imaginary Teleprompter on a Linux repository

One of the best ways for Imaginary Teleprompter to be exposed to users are Linux repositories. We're a small, busy, team and cannot get behind all the distributions by ourselves. If you're a Linux enthusiast, like we are, please consider helping us get this software to the all the mayor distros.

1. First, familiarize yourself with our building process at:
<https://github.com/ImaginarySense/Teleprompter-Electron#installer-building-instructions>
2. Then, if you think you can help to create and maintain the packages for the distribution of your choice, write an email to Javier Cordero <javier@imaginary.tech> regarding your intentions.

Donate Hardware to Us

We want Imaginary Teleprompter to work with as many hardware interfaces used in the industry as possible. We also want it to work on as many platforms as possible. You could mail us any spare hardware to:

Javier Cordero Pérez (Teleprompter)
Urb. El Prado
75B Calle Israel Roldán Blas
Aguadilla, PR 00603-5852

It would be best if you communicate with us by e-mail beforehand, at support@imaginary.tech. We could use any of the following hardware:

- Remote Controls (USB, infrared, RF, Bluetooth 2.1 or Bluetooth 4.0 LE)
- Working Chromebook laptop (we'll start supporting this platform immediately if someone donates one)
- Working Apple computer that runs, at least, OS X 10.9
- Working, small, computer running the latest FreeBSD with Gnome 3 or MATE

Become a Translator

We're hoping that starting version 2.4.0 onward, Teleprompter can allow translations from the open source community into other languages other than English. Let us know if you're interested in contributing a translation at support@imaginary.tech, no compromises, it's just so we can let you know when our site is ready for such contribution.

Add a Feature or Support a Platform

HTML, CSS, Javascript, interest in the software, and a technical mind is all it takes to be a part of the team. Before publishing a fork, a feature or submitting a patch, you should get in touch with us to make sure what you intend to make will stay in a future release.

Write to us at support@imaginary.tech, regarding your interest to develop a feature, or support a platform, such as ARM architecture Linux distributions, or mobile operating systems.

If you intend to develop a feature and would like it to be included in a future release, please read [How to Join our Development Team](#).

How to Join our Development Team

1. Learn to use the software and think of ways how it could be programmed.
2. Successfully build distribution packages for your operating system following our building process at:
<https://github.com/ImaginarySense/Teleprompter-Electron#installer-building-instructions>
3. Read the source code to have a basic understanding of how the software was written and how it works. The most important files are index.html, teleprompter.html, js/editor.js, and js/teleprompter.js from [Teleprompter-Core](#); and main.js with package.json from [Teleprompter-Electron](#). CSS forms a huge part of the logistics behind Teleprompter's functionality, so you may also check css/editor.css, and css/teleprompter.css from [Teleprompter-Core](#).
4. Write to us at support@imaginary.tech, regarding your interest to develop a feature and how you might do it. We'll take your ideas into consideration and reply to you with what we think could be the best way to develop such a feature and more interaction will follow from there.

How We Manage Copyright Under the GPL

Each contributor retains copyright over their own work and voluntarily shares that copyright with Imaginary Sense when they make code contributions. Only Imaginary Sense and the original authors, Javier Oscar Cordero Pérez and Victor Alfonso Ortiz Alvarado, retain full copyright over the software.

Imaginary Sense, the original authors and all contributors to the software agree to give up certain rights as copyright holders under the General Public License version 3.0.

Developer Guidelines

Building Process

For Every Platform You Must First...

1. Install Git and NodeJS on your system.
2. Open a Terminal or Command Prompt at your desired build location.
3. Clone the Teleprompter-Electron repository and its submodules with the following command.

```
git clone --recursive https://github.com/ImaginarySense/Teleprompter-Electron.git
```

If by any chance you can't use Git, you may instead of steps 2 and 3 do the following:

2. Download Teleprompter-Electron's source from <https://github.com/ImaginarySense/Teleprompter-Electron/archive/master.zip> and extract to a desired location.
3. Download Teleprompter-Core's source from <https://github.com/ImaginarySense/Teleprompter-Core/archive/master.zip> and extract into the /app folder of the previously extracted file.

4. Move to the root folder of the Teleprompter-Electron project.

```
cd Teleprompter-Electron
```

5. Download dependencies and submodules.

```
npm run setup
```

6. Follow platform specific building steps. If builds are successful, you should find your binaries inside the *dist* folder.

For Linux

1. If you're building for Linux, depending what packages you intend to build you should install their dependencies as shown at: <https://www.electron.build/multi-platform-build#linux>. Commands and dependency names may vary across distributions. The following instructions assume you're using a Debian/Ubuntu derivative.

```
sudo apt-get install --no-install-recommends -y icsutils graphicsmagick
sudo apt-get install --no-install-recommends -y rpm      #To build rpm
sudo apt-get install --no-install-recommends -y bsdtar  #To build pacman
sudo apt-get install --no-install-recommends -y snapcraft #To build snap
```

2. Open Teleprompter-Electron's "package.json" in the text editor of your preference.
3. Locate the following lines:

```
"linux": {
  "target":
```

4. Under “target”, remove any targets that you don't intend to build for. (For example, you would leave deb for Ubuntu, rpm for Fedora, and pacman for Arch or Manjaro.) Use `tar.gz` for any other distros and `AppImage` if you wish to create a universal, portable, Linux app.
5. Save your changes.
6. Run the command that corresponds to your operating system's architecture:
 - To build 32 bit packages, run:
`npm run dist:linux32`
 - To build 64 bit packages, run:
`npm run dist:linux64`
 - To build both 32 and 64 bit packages, run:
`npm run dist:linux`
 - To build ARM7l packages, run:
`npm install -g electron-builder`
`npm run dist:linuxarm`

For Mac OS

Execute “npm run dist:osx”.

Building process only works on computers running Mac OS.

For Windows

- Execute “npm run dist:win64” to create a 64 bit installer.
- Execute “npm run dist:win32” to create a 32 bit installer.
- Execute “npm run dist:win” to create both 32 bit and 64 bit installers.

Development Cycle

Imaginary Teleprompter follows an Agile Project Management methodology. Most of the planning and management occurs across informal communications in the form of e-mails, personal meetings and a Facebook Messenger chat group.

Since the team also works on other projects from Imaginary Sense and outside of Imaginary Sense, the rigorousness of our development cycle varies from time to time. Nevertheless, no matter how much progress we've made during a sprint, new releases will be made on schedule and a one month testing phase will precede every release.

Release Dates

From version 2.3.0 forward, new versions of Imaginary Teleprompter are released every *August 1st* and every *January 31st*. Bug and security fixes are released sporadically all year long.

Sprints

Planning Phase

Up to a week for small features; as long as it takes for more complex features.

1. We pick a number of features, around 10 through 12 additions and improvements, from a list of all the improvements that could be done. Priority is given to features and bug fixes requested by the users and those that impact usability.
2. Smaller features are ordered by priority and split evenly across developers.
3. The more complex features, those that require research and experimentation, are usually worked across more than one sprint. and are split across developers regarding their understanding of existing code and their experiences with similar code. Developers shouldn't focus on more than two complex features at a time.
4. During this phase various prototypes of the more complex features are developed to determine which implementations work the best.

Development Phase

2 to 3 months prior the testing phase. May vary under special circumstances.

5. Developers work in parallel. Some prefer to develop locally, others in separate branches. All developers announce which files they're modifying in order to prevent conflicts when committing changes. Sometimes developers agree to wait for one another to finish before committing changes.

Despite most development being done in parallel, we do pair programming to help developers who're having difficulties with their code or understanding the requirements. Pair programming's been done both in person and through remote screen sharing sessions.

6. Developers test their own features and make sure they don't interfere with other features nor crash the program.
7. After testing is done, code that wasn't commented during the development of the features gets commented. Sometimes we developers are in a hurry and forget to follow this practice. Feel free to file a support ticket if you feel some specific code needs additional commenting to be understood.
8. Code gets uploaded to the GitHub development branches.
9. Other developers test the new feature, look at the code and give feedback to its developer.
10. Improvements are made to the code, usually by the author, sometimes by those who've given feedback, and these improvements get uploaded to the GitHub developer branches.

Testing Phase

1 to 2 months prior release

11. We agree to develop no new features during this phase.
12. Focus goes into improving existing code for efficiency and usability.
13. We test for extreme use cases.
14. Features that don't perform as desired are subjected to be disabled or removed for before release.
15. Additional comments are made to the code where required.

Coding Practices

1. Whenever possible a feature must be able to run from all supported web browsers and not depend of Electron in any way. A feature may be created exclusively for Electron only if there's no way to achieve it in all supported web browsers without violating browser security restrictions.
2. We write in pure JavaScript and avoid JQuery to maximize performance.
3. JQuery is used only to control JQuery libraries such as the Timer and JQuery/PEP.
4. Keep JavaScript, CSS and HTML decoupled.
5. Avoid using “!important” on CSS instructions.
6. Default values for DOM elements should be assignable from the HTML source, not the JavaScript.
7. Code is abstracted to enable re-usability and maintainability.
8. Teleprompter operations and the Editor follow a procedural programming paradigm.
9. Reusable elements have their own libraries and they're coded with an object oriented paradigm.

Syntax

1. camelCaseVariableNames
2. Brackets go next to function declarations, like this:

```
function methodName(requiredArgument1, .., optionalArgumentsObject) {  
    // Contents  
}
```
3. Comments are always welcome. Comments should describe purpose of code, not describe exactly what the code is doing unless this is not clear from simply reading the code itself.
4. Variable names and code should be descriptive in such way that comments aren't obligated for an English speaker to understand the code.
5. Every method must have a comment describing its purpose before declaration.
6. If a developer didn't comment his code and you're having trouble understanding it, please let us know.
7. Comments should always be used to describe arithmetic, mathematical functions and inter-process communication.
8. Comments and reference documentation should always be in English to enable more developers from outside of the team. Exceptions may be made with Spanish and other languages that the team speaks, but these should be translated to English, by a speaker of both languages, before release.

User Manual

Installation

Windows

Versions 2.3.x and Prior

1. Download the latest Windows binary from SourceForge:
<https://sourceforge.net/projects/teleprompter-imaginary-films/files/latest/download>
Get the current 32 bit installer from:
<https://sourceforge.net/projects/teleprompter-imaginary-films/files/v2.3.1/Teleprompter-32bit-2.3.1.exe/download>
2. Double click on the installer. The following Image will appear on the screen:



3. Once it's gone, Teleprompter has been successfully installed.

You will get the an error if you attempt to run the 64 bit version on a Windows 32 bit machine, and the following error if you run attempt to run Teleprompter on a machine Windows XP or earlier:

"Not a valid Win32 program."

Mac OS

Download the latest Mac OS binary from SourceForge:

<https://sourceforge.net/projects/teleprompter-imaginary-films/files/latest/download>

Open the .dmg file. The following window will appear:



Drag Teleprompter into the Applications folder to install it. Once it's been successfully copied, close the .dmg file.

Linux

AppImage is a standalone version of Teleprompter that works on all mayor Linux distributions. It's best to use your distribution's preferred installation method, if possible, instead of the AppImage; because those allow to fully remove the software afterwards. Also "imaginary-teleprompter" is not added to your terminal environment's program list when using the AppImage.

You will get the following error if you attempt to install the 64 bit version on a Linux 32 bit machine:

```
# ./imaginary-teleprompter
-sh: ./imaginary-teleprompter: cannot execute binary file: Exec format error
cannot execute binary file: Exec format error
```

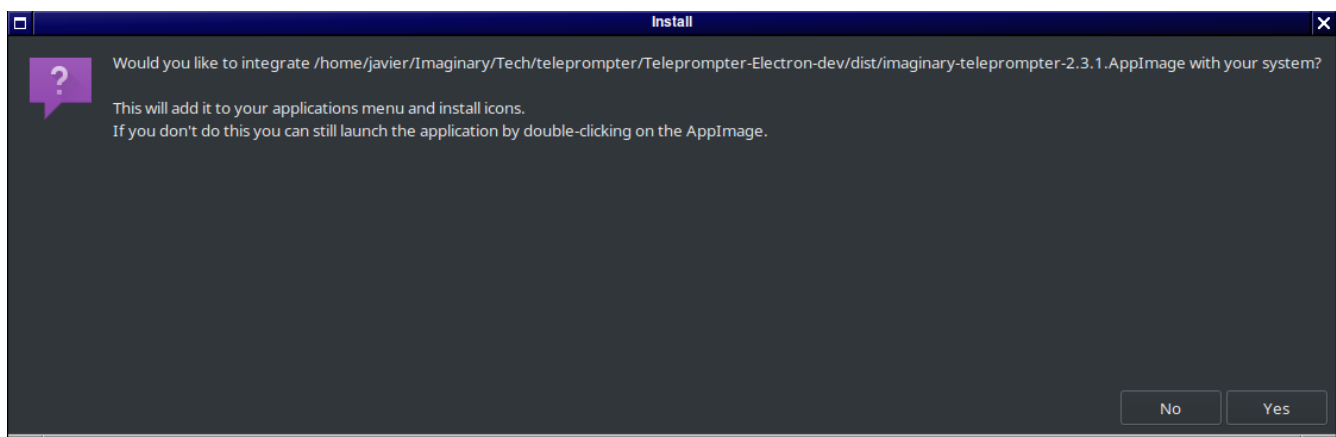
Get the latest 32 bit installers from:

<https://github.com/ImaginarySense/Teleprompter-Electron/releases>

AppImage

AppImage is a standalone version of Teleprompter that works on all mayor Linux distributions. It's best to use your distribution's preferred installation method if possible instead of the AppImage, because those allow to fully remove the software afterwards. Your distribution may create shortcuts in your desktop environments to where the AppImage resided at first execution. Those shortcuts are not automatically updated nor removed upon moving or deleting the AppImage. Also "imaginary-teleprompter" is not added to your terminal environment's program list when using the AppImage.

1. Download the latest Linux AppImage from SourceForge:
<https://sourceforge.net/projects/teleprompter-imaginary-films/files/latest/download>
2. Open the AppImage file.
3. Your distribution may prompt you to integrate it to the system for next execution. Imaginary Teleprompter will load immediately after you answer to this prompt.



System integration prompt in Manjaro Linux with WindowMaker.

Ubuntu / Debian

Download the current Debian package from SourceForge:

<https://sourceforge.net/projects/teleprompter-imaginary-films/files/v2.3.1/imagianry-teleprompter-2.3.1.deb/download>

Fedora

Download the current RPM package from SourceForge:

<https://sourceforge.net/projects/teleprompter-imaginary-films/files/v2.3.1/imagianry-teleprompter-2.3.1.rpm/download>

Arch

Download the current .pacman package from:

<https://sourceforge.net/projects/teleprompter-imaginary-films/files/v2.3.1/imagianry-teleprompter-2.3.1.pacman/download>

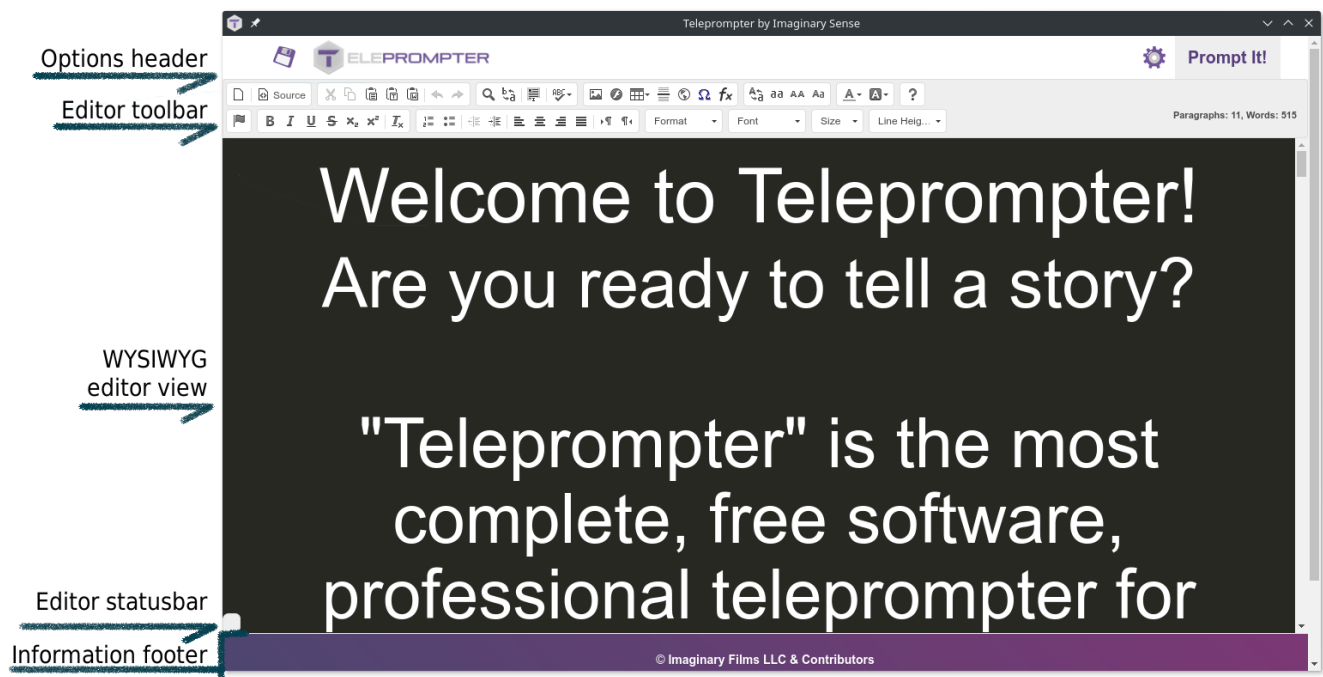
Manjaro (KDE, Octopi)

1. Download the latest .pacman package from:
<https://sourceforge.net/projects/teleprompter-imaginary-films/files/v2.3.1/imagianry-teleprompter-2.3.1.pacman/download>
2. Rename the downloaded file from 'imaginary-teleprompter-x.y.z.pacman' to 'imaginary-teleprompter-x.y.z.pkg.tar'
3. Start Octopi.
4. Click "File" > "Install local package..."
5. Select the downloaded file.
6. Confirm installation.
7. Enter your password when requested.

Editor Window

Imaginary Teleprompter's primary window. This is where the user prepares the contents to be prompted. It's composed of an options header, a rich text editor, and a footer with information about the software.

The rich text editor is composed of a toolbar, a "what you see is what you get" editing view, and the editor's status bar.




Options Header Bar

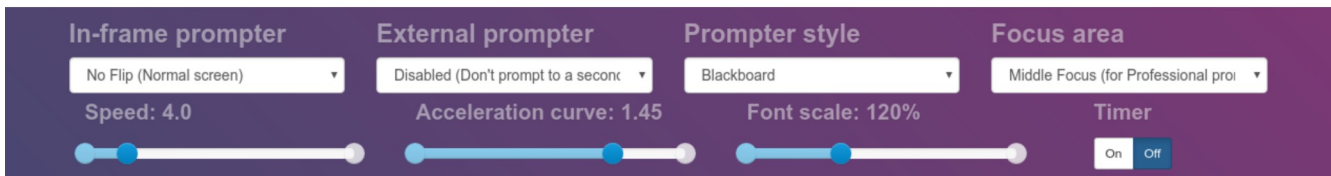


- **Save icon:** Opens the [File pane](#), where the user can create, rename, import and export files.
- **Teleprompter logo:** Links to Imaginary Teleprompter's [website](#).
- **Settings icon:** Opens and closes the [Setup Panel](#).
- **Prompt It option:** Starts [prompting](#). Same as pressing the F8 key.

Setup Panel

Allows setting up teleprompter instances. Click on the cog-wheel  to open and close the Setup Panel. All settings in the panel are saved automatically on program close, except for the current custom style.

See [Set Default Style](#) for more details about this...

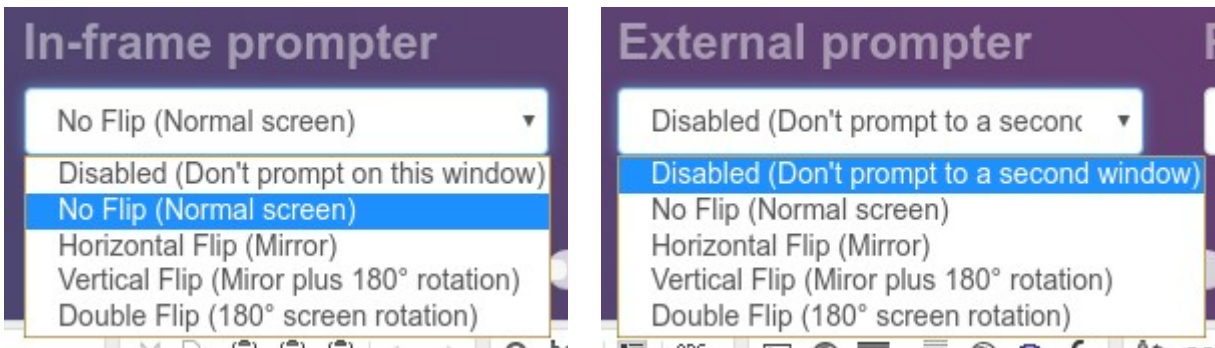


Open setup panel with default settings.

Screen & Mirror Controls

You can have up to two independent prompt outputs with Imaginary Teleprompter. These are meant to provide users with mirroring options across various hardware solutions. One output can be used for prompting a flipped image that's copied across multiple screens, while the other could prompt an image without flip, or a different flip, for monitoring purposes. Any or both outputs can accomplish one purpose or the other. Both outputs are rendered independently to provide the highest quality.

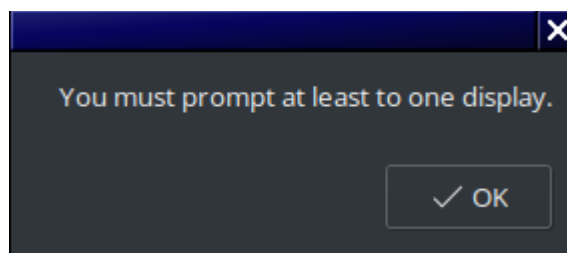
Mirroring Modes



Mirroring modes are the same in the In-frame prompter as in the External prompter.

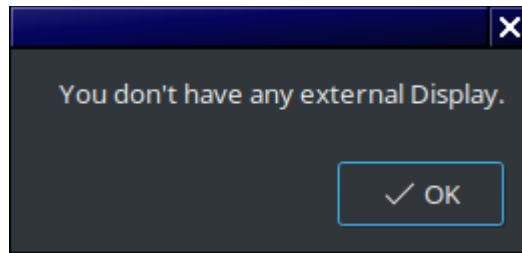
1. **Disabled:** Said prompter won't be instantiated.

At least one prompter must be enabled when you choose to start prompting.



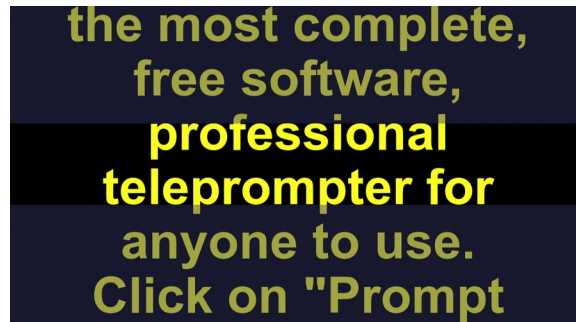
You must prompt at least to one display.

If you have only one screen and both instances are enabled, you will be forced to disable one. This last premise applies only to the Electron version of the software.



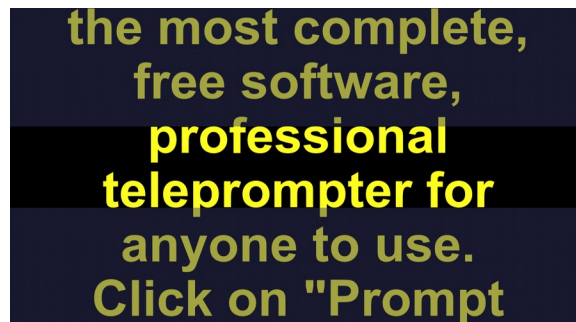
Electron only warning: You don't have a second display.

2. **No Flip:** Prompt is shown as is.



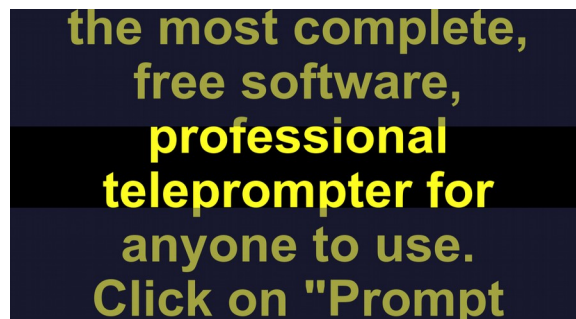
No flip.

3. **Horizontal Flip:** Prompt is flipped horizontally. Use this with a teleprompter mirror and a regular screen with its bottom looking towards the camera.



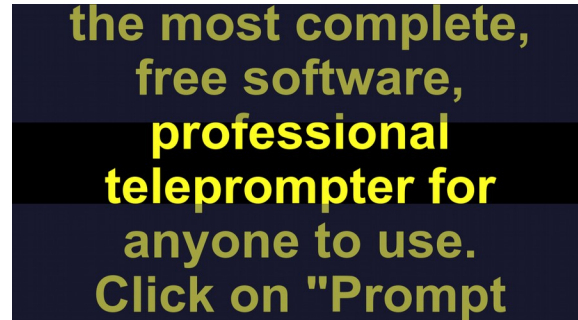
Horizontal flip.

4. **Vertical Flip:** Prompt is flipped vertically. Use this with a teleprompter mirror and a regular screen with its bottom looking away from the camera.



Vertical flip.

5. **Double Flip:** Prompt is flipped horizontally and vertically, which is the equivalent of rotating it 180°. Use this for on a prompter with no mirroring, where the screen is rotated on its head.

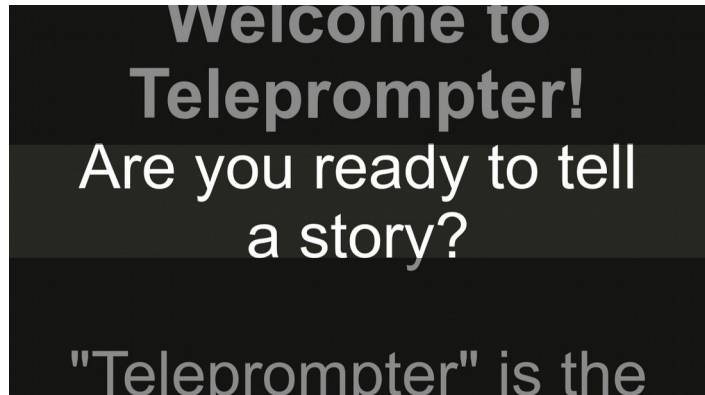


the most complete,
free software,
professional
teleprompter for
anyone to use.
Click on "Prompt

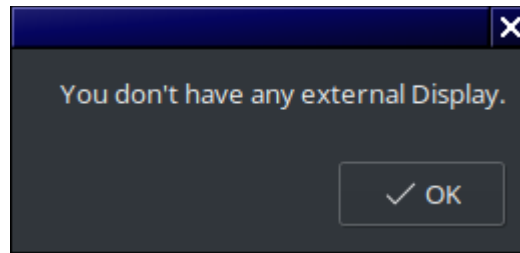
Double flip.

- This bar also causes the vertical center to be offset slightly towards the button. For this reason, we advise using the “In-frame” for monitoring and the “[External](#)” for your talents.
- The “In-frame” prompter loads faster than the “External” because there’s no need to load a separate window.
- Flipping instances causes no perceivable performance impact in either prompters.

External Prompter



- The External prompt loads your prompt on a new window.
- When one screen is present, the External prompt loads its window on top of the Editor window.
- When two screens are present, the External instance loads its window on the screen adjacent to the Editor window (or In-frame instance). This doesn’t always occur in the [online version](#) of this software, and is something we cannot control because of browsers’ security measurements. Use the [standalone version](#) to get all of Imaginary Teleprompter’s features.
- When more than two screens are present, the External instance loads its window on the first of the first two screens where the Editor window (or In-frame instance) isn’t located.
- You can tell the “External” instance from the “In-frame” because the In-frame shows the [Options Header Bar](#) and the “External” does not. The user must press the “Escape” key from the “External” instance in order to return Teleprompter to its prior state.
- We recommend using the In-frame for monitoring and the External for your talents. This is because the In-frame’s vertical center gets slightly displaced towards the bottom, while the External’s remains at the screen’s true center.
- The “External” prompter loads slower than the “In-frame” because it needs to open a separate window.
- Flipping instances causes no perceivable performance impact in either prompters.
- If you have only one screen present and you try to use both the In-frame and the External prompter, you will get the following error in the [standalone version](#) of this software:

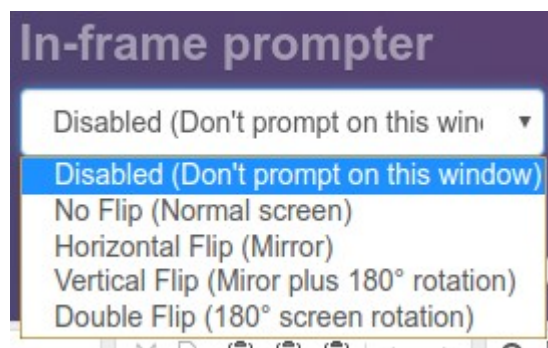


You don't have an external Display.

Live Update Mode

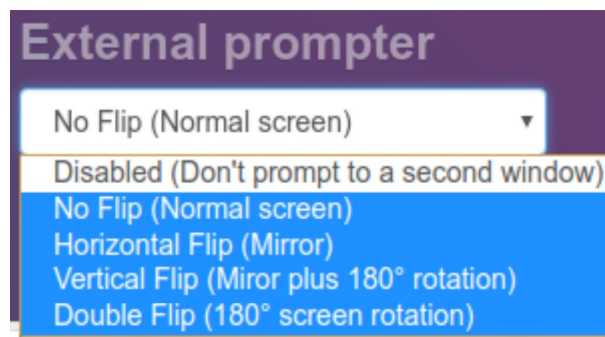
You can add new contents to the prompter, live, by following these steps:

1. Disabling the In-frame. This will leave the editor open when you begin prompting.



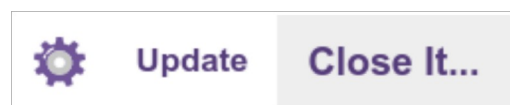
Select disabled on the "In-frame prompter".

2. Enable the External prompter. Select any one of the external options but “Disabled”.



Select any of the highlighted options.

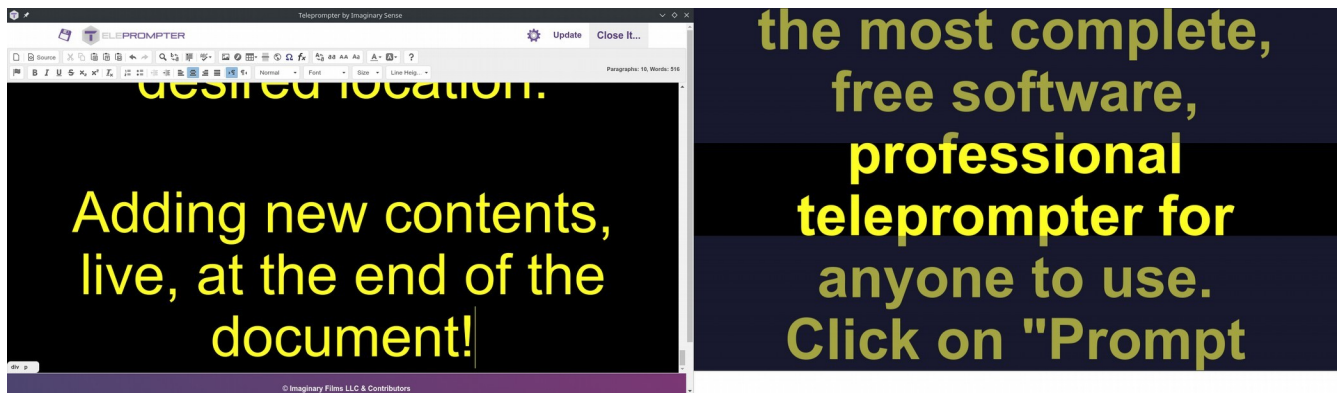
3. When you start prompting, the Editor will remain on the current screen, and the prompt will occur at the first screen Imaginary Teleprompter isn't occupying. The Options Header Bar will update to contain “Update” and “Close It...” options.



- Click on the Editor, or press Tab enough times, to focus it and make your changes. Click outside of the editor or into the prompter's screen to control the prompter with the keyboard.

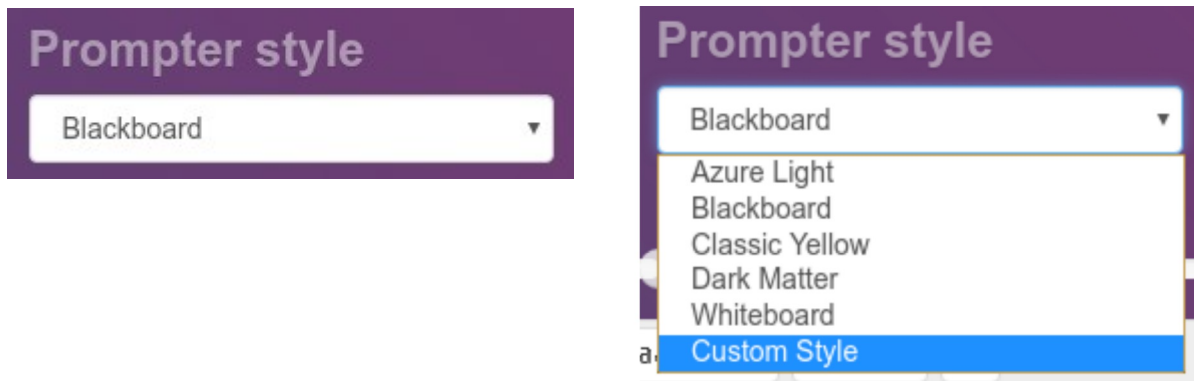
It is advisable to use the WASD keys to control the prompter while in Live Update Mode, to prevent the Editor window from scrolling.

- After you've made your changes, click on "Update" for your changes to take place in Teleprompter.



New contents being typed into the editor while the external prompter is on and scrolling at a constant speed.

Prompter Style

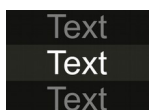


Imaginary Teleprompter comes with a variety of color combinations for reading. Styles, or color combinations, consist of a font color, a background color, and the color of a transparent overlay that covers the areas the talent should not read from.

Color Combinations for Version 2.3.1



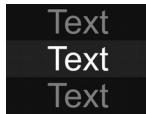
- Azure Light:** background: #000000; overlay: #202020; text: #229FFF
*Light azure (blue) text, over black background and dark gray overlay
Inspired on Star Wars: A New Hope original opening's color scheme.*



- Blackboard:** background: #000000; overlay: #202020; text: #229FFF
*White text, over soft gray background and black overlay.
Inspired on the Monokai color scheme.*



- **Classic Yellow:** background: #000000; overlay: #000000; text: #FFFF00
Yellow text, over a black background and black overlay
Inspired on popular Credit scrolls from the 1970s.



- **Dark Matter:** background: #222222; overlay: #000000; text: #FFFFFF
White text, over dark gray background and black overlay
A reference to the astronomical term.



- **Whiteboard:** background: #FFFFFF; overlay: #CCCCCC; text: #272822
Dark gray text, over white background and light gray overlay
A play on the Blackboard color scheme.

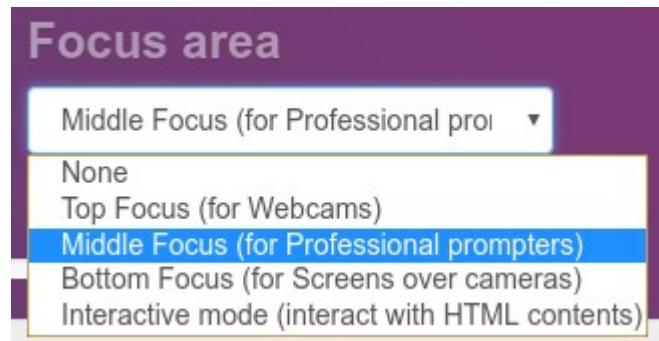
Custom Style

You can create new custom color schemes by selecting “Custom Style” from the Prompter style options. Selecting “Custom Styles” will load the *Custom Styles Administration Pane*.

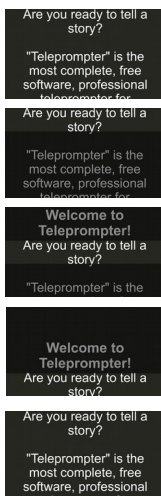
See [Custom Styles Administration Pane](#) for more details.

Focus Area

Focus Area or *Focus Modes* refers to the part of the screen that marks the area the talents should read from.



There are 5 focus modes in Teleprompter (older versions may have less):



1. **None:** Shows no overlay. The top of the screen is considered to be the point where the talent reads from.
2. **Top Focus:** Shows transparent overlay. The top of the screen is considered to be the point where the talent reads from.
3. **Middle Focus:** Shows transparent overlay. The middle of the screen is considered to be the point where the talent reads from.
4. **Bottom Focus:** Shows transparent overlay. The bottom of the screen is considered to be the point where the talent reads from.
5. **Interactive mode:** Same look as “None”. Differs in that mouse controls are removed to enable the user to interact with the hypertext being scrolled.

Speed and Acceleration

Every time the user presses Up or Down on the keyboard when prompting, velocity is decreased or increased one step, correspondingly. Scrolling the mouse wheel forward or backwards also increases or decreases one or more steps in velocity.

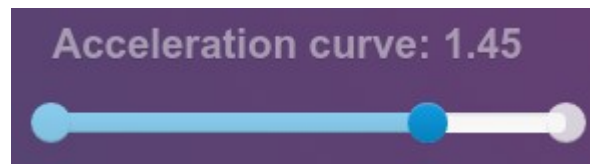
Actual velocity is relative to each teleprompter instance's resolutions. These are abstracted and managed internally to allow a consistent use experience across all kinds of screens and devices.

Speed



Controls steps' speed increments.

Acceleration

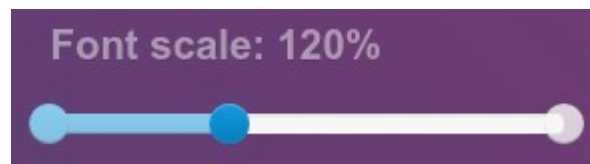


Controls steps' acceleration increments.

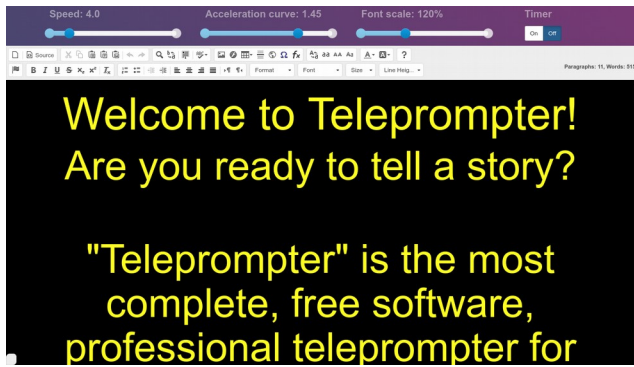
Speed and Acceleration Tips

- As the scale of the font increases, speed and acceleration steps should be increased as well.
- Lower speed increments combine well with higher acceleration increments.
- The lower the speed increments, the more steps it takes for the prompter to reach a desired speed.
- The higher the acceleration curve, the quicker the prompter will increase and decrease velocity after a few steps.

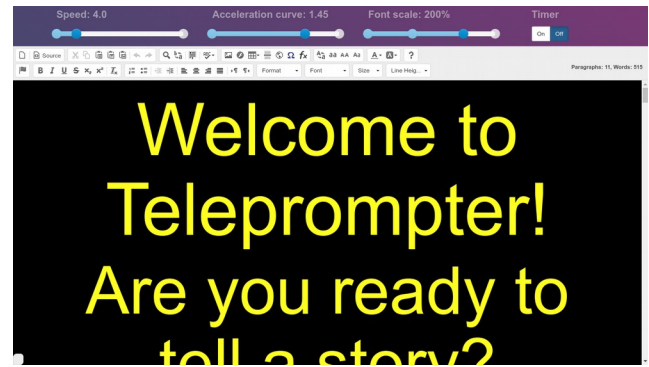
Font Scale



Scales all text with relative units of measurement. By default, all text in Imaginary Teleprompter's Editor uses relative units measurement.

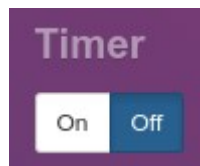


Font scale at default 120%.

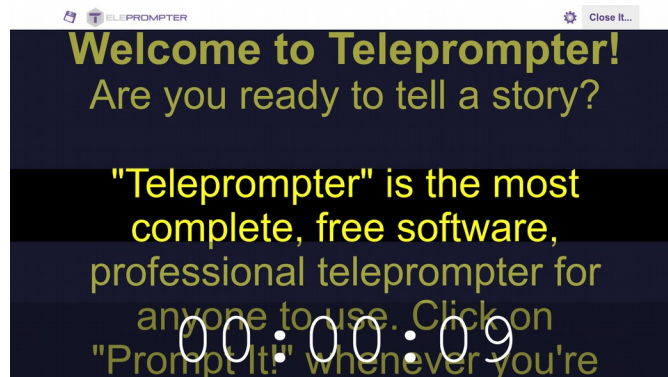


Font scale increased to 200%.

Timer



Enable or disable a timer for both in-frame and external prompters. The blue background indicates whether the timer is On or Off.



Timer, counting, at the bottom of the screen.

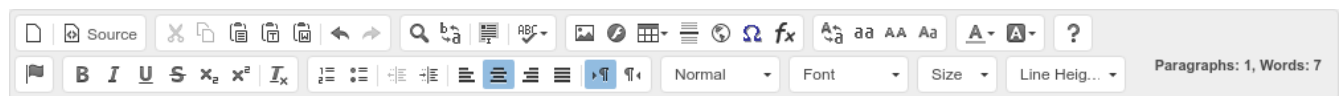
Editor

Imaginary Teleprompter is one of the few teleprompter software to come with a Rich Text Editor. Most teleprompter software allows editing plain text only and have no image support. We achieve this through the power of HTML, CSS, JavaScript, CKEditor and applied calculus.

CKEditor is a free software Rich Text Editor that we have incorporated into Teleprompter. We chose CKEditor over other editors because it is very feature rich, extensible and robust. It works very much the same way as any other word processor, such as Libre Office Writer and Google Docs.

Editor Toolbar


Use this Toolbar to make changes to the contents you'll prompt. It's useful to study and experiment with it's features, for here's where content creators can get the most creative.




File Options

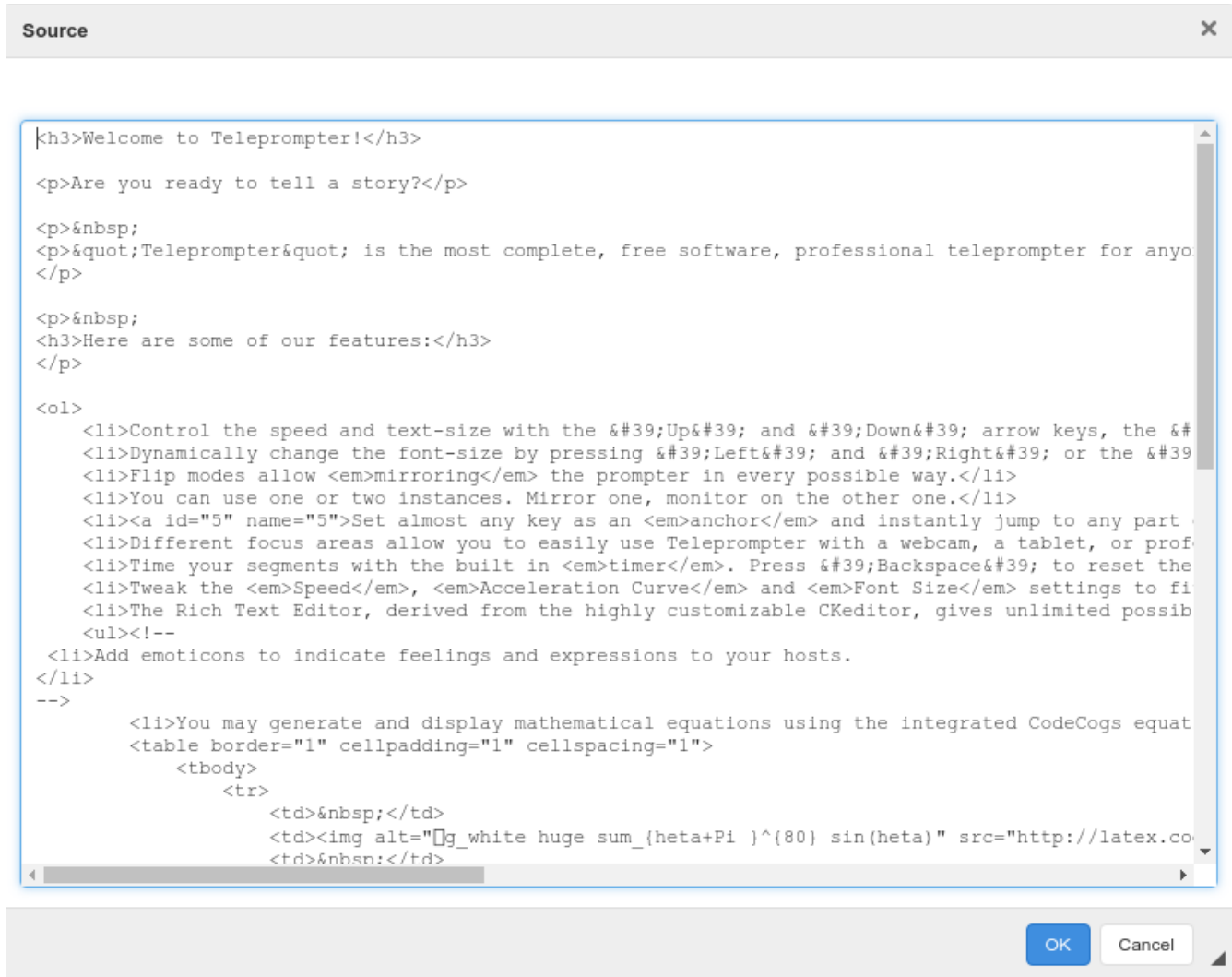


Clear File

 You can clear the contents of the current document by pressing this button. This does not create a new document. This button shall be removed in future releases because it causes confusion.

Source Editor

 **Source** Opens an HTML source code editor for the user to make changes to the contents in the regular editor. Advanced users could use the “Source Editor” along with “Interactive Mode” to use Teleprompter as a programmable interactive display.





Source code editor.

Edit Options

Cut, Copy and Paste






Cutting and Copying

-  Cut (CTRL-X): Removes selected contents and adds them to the clipboard.
-  Copy (CTRL-C): Does not remove selected contents but does adds them to the clipboard.

Pasting

These icons are present to be used in the [online version](#) of Teleprompter. The [standalone version](#) is based on the online version, so they're still present but don't really apply.

-  Paste (CTRL-V): Inserts the most recent contents of the clipboard into the location indicated by the keyboard's cursor.
-  Paste without format: Removes all format from text before pasting.
-  Paste from MS Word: Allows pasting text with MS Word formatting in older browsers.

Find and Replace

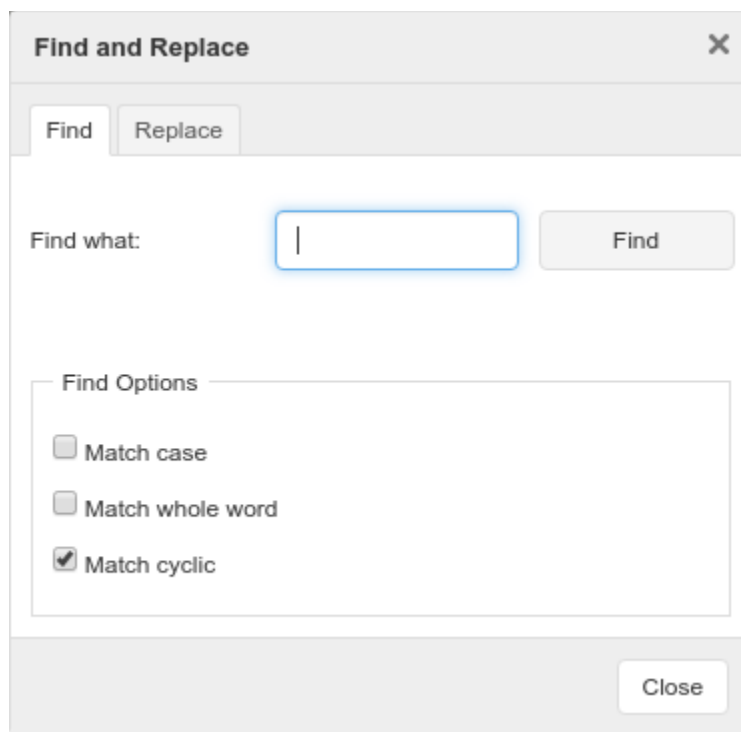


Find



Loads the Find and Replace dialog on its “Find” tab.

1. Type the text you intend to look for in the text-box to the left of the “Find” button.
2. Enable and disable options by checking and unchecking the check-boxes under Find Options.
3. Press the “Find” button to look for the next instance of your text in the document.
4. Once you’re done, click on “Close” or the “X” at the top right.



Find dialog.

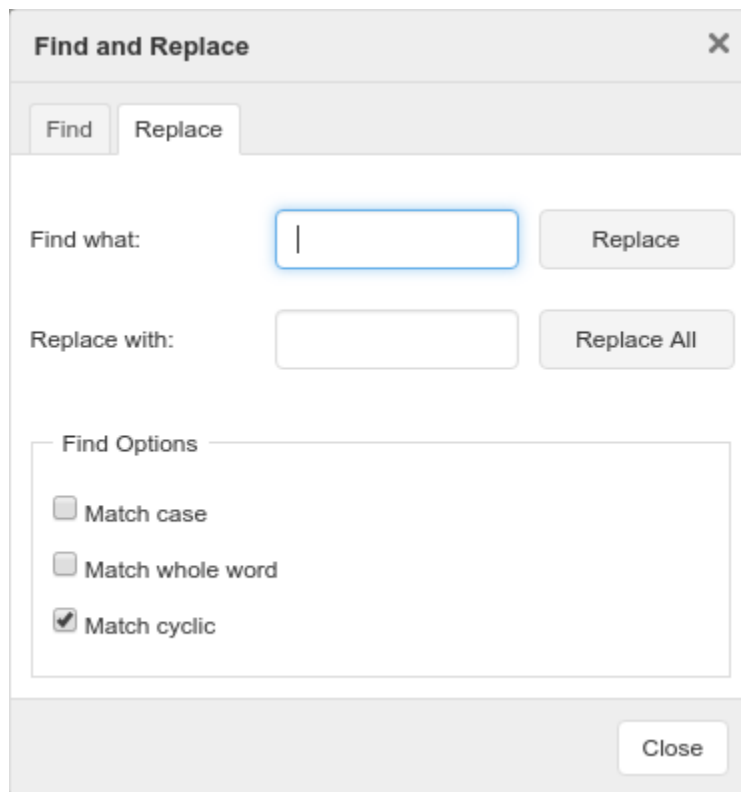
Replace



Loads the Find and Replace dialog on the “Replace” tab.

1. Type the text you intend to replace into the text-box to the left of the “Replace” button.
2. Type the text you intend to replace with into the text-box to the left of the “Replace All” button.
3. Enable and disable options by checking and unchecking the check-boxes under Find Options.
4. Press the “Replace” button to replace the next instance of text matching in the document.
5. Or press the “Replace All” button to replace all instances of text matching in the document.

6. Once you're done, click on "Close" or the "X" at the top right.



The image shows a "Find and Replace" dialog box with a title bar containing a close button (X). Below the title bar are two tabs: "Find" and "Replace". The "Find" tab is active. The dialog contains the following elements:


- Find what:** A text input field with a blue border and a vertical cursor.
- Replace with:** A text input field.
- Find Options:** A section containing three checkboxes:
 - ☐ Match case
 - ☐ Match whole word
 - ☒ Match cyclic
- Buttons:** A "Replace" button next to the "Find what" field, a "Replace All" button next to the "Replace with" field, and a "Close" button at the bottom right.

Find and replace dialog.


Find Options

1. **Match case:** Returns matches only when the capital letters and non capital letters match as.
2. **Match whole word:** Matches only to text that's being preceded and followed by white space.
3. **Match cyclic:** Searches begin at the location of the keyboard cursor. Enabling this makes the the algorithm search the rest of the document from the beginning, after the bottom has been reached.

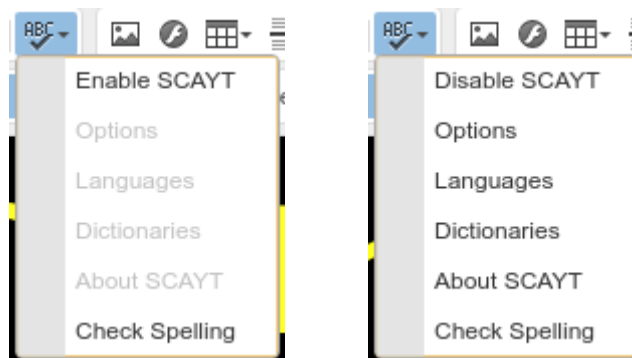
Select All

 Selects all of the contents in your editor. You may also press CTRL-A while the editor is focused.

Spell Check

 Opens the Spell Check options. This spell check works only with the [online version](#) and requires internet access to function. Future versions of Teleprompter will implement a different spell check for offline use with the standalone version.

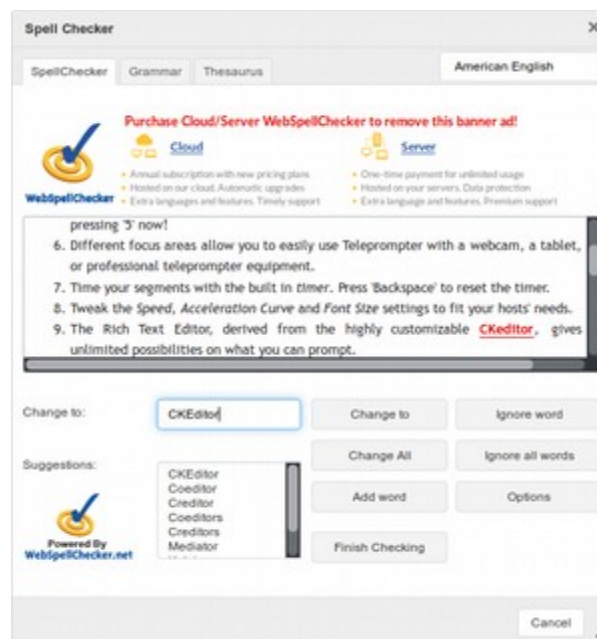
“Enable SCAYT” to have real time spell checking. This must be done once per session. SCAYT checks newly typed text, not that which was previously typed.



Default values.

SCAYT Enabled.

Click on “Check Spelling” to open the Spell Checker dialog.



Spell Checker Dialog

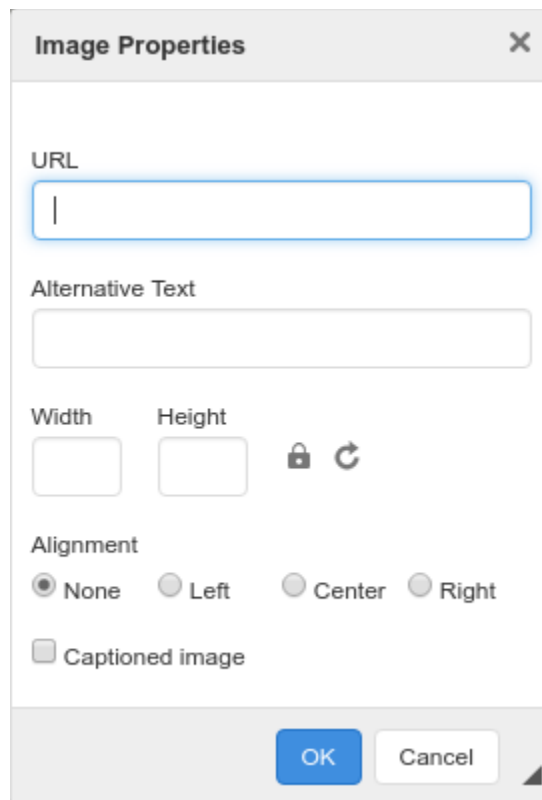
See http://scayt.com/user_manual/scayt_plugin_for_ckeditor4_user_manual.pdf for full documentation on how to use the spell check.

Insert Options

Insert Image

 Allows inserting images into Teleprompter.


- URL: Type the address to the image's location. The location can be somewhere in the internet or locally, in your computer. Please note that if a Imaginary Teleprompter file is exported, the linked images are not exported with it. We shall fix this in future releases.
- Alternative Text: Placeholder text to show if the image isn't found or currently unavailable.
- Width: The image's width measurement in pixels. This value is ignored and a default size of 100% available space is used instead.
- Height: The image's height measurement in pixels. This value is ignored and an automatic value that does not change the aspect ratio is used instead.
- Alignment: Because width is automatically set to 100%, any alignment value should result in identical outputs.




The image shows a dialog box titled "Image Properties" with a close button (X) in the top right corner. The dialog contains several input fields and options:

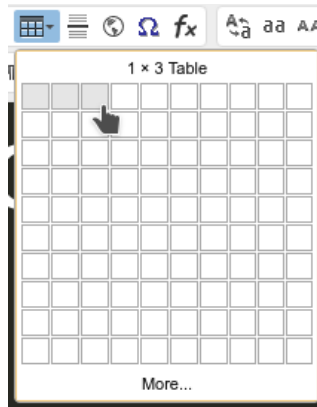
- URL:** A text input field with a vertical cursor.
- Alternative Text:** A text input field.
- Width:** A text input field.
- Height:** A text input field.
- Lock/Unlock:** A lock icon and a circular arrow icon.
- Alignment:** Four radio buttons labeled "None", "Left", "Center", and "Right". The "None" button is selected.
- Captioned image:** A checkbox.
- Buttons:** "OK" and "Cancel" buttons at the bottom right.

Insert Flash Content

 The ability to play flash contents is present but there's no official support for it. It works similar to [Insert Image](#). Please see Ck Editor's documentation for more details on how to use it.

Insert Table

 Pops a grid where the user may click any cell to create a table of the size indicated by the number of gray colored cells.



User creating a 1 by 3 cells table.

A user may press “More...” to setup a table from the Table Properties window.

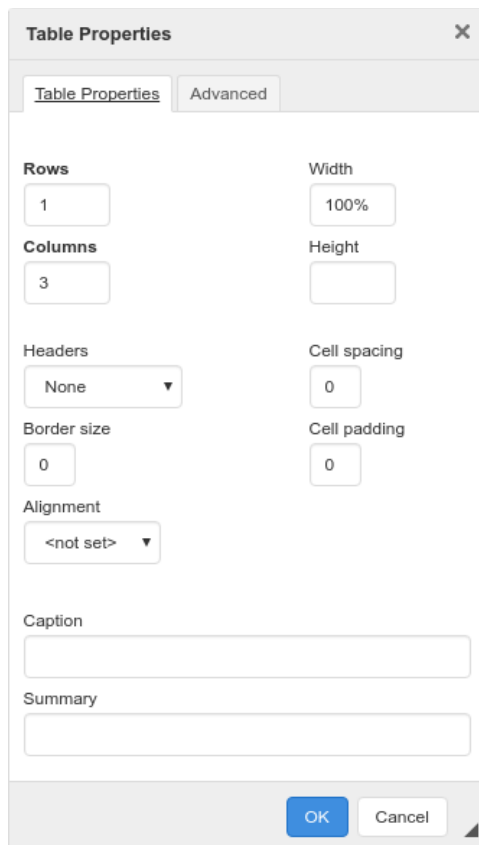
A screenshot of the 'Table Properties' window with the 'Table Properties' tab selected. It contains various settings for a table. The 'Rows' field is set to 1, and 'Columns' is set to 3. 'Width' is set to 100%, and 'Height' is empty. 'Headers' is set to 'None', 'Border size' is 0, and 'Alignment' is '<not set>'. 'Cell spacing' and 'Cell padding' are both set to 0. There are input fields for 'Caption' and 'Summary'. At the bottom are 'OK' and 'Cancel' buttons.

Table Properties	
Table Properties Advanced	
Rows	Width
1	100%
Columns	Height
3	
Headers	Cell spacing
None	0
Border size	Cell padding
0	0
Alignment	
<not set>	
Caption	
Summary	
OK Cancel	

Table Properties window.

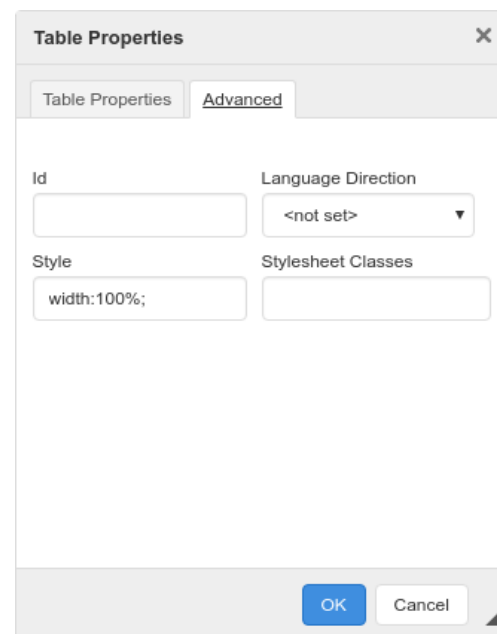
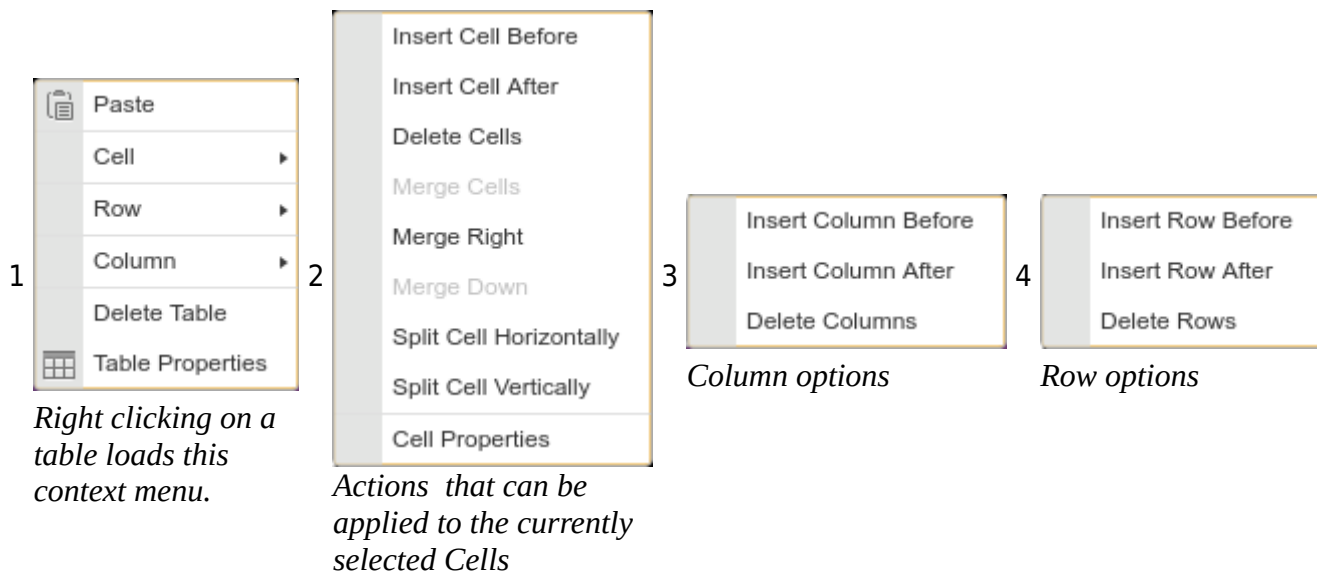
A screenshot of the 'Table Properties' window with the 'Advanced' tab selected. It contains settings for advanced table features. 'Id' is an empty field, and 'Language Direction' is set to '<not set>'. 'Style' is set to 'width:100%', and 'Stylesheet Classes' is an empty field. At the bottom are 'OK' and 'Cancel' buttons.

Table Properties	
Table Properties Advanced	
Id	Language Direction
	<not set>
Style	Stylesheet Classes
width:100%;	
OK Cancel	

Advanced Table Properties.

Table Context Menu



Cell Properties

The Cell Properties dialog box contains the following fields and options:


- Width:** Input field with a unit dropdown (currently set to 'pixels').
- Height:** Input field with a unit dropdown (currently set to 'pixels').
- Word Wrap:** Dropdown menu (currently set to 'Yes').
- Horizontal Alignment:** Dropdown menu (currently set to '<not set>').
- Vertical Alignment:** Dropdown menu (currently set to '<not set>').
- Cell Type:** Dropdown menu (currently set to 'Data').
- Rows Span:** Input field.
- Columns Span:** Input field.
- Background Color:** Input field with a 'Choose' button.
- Border Color:** Input field with a 'Choose' button.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom right.

Cell options: lookup HTML tables for more information.

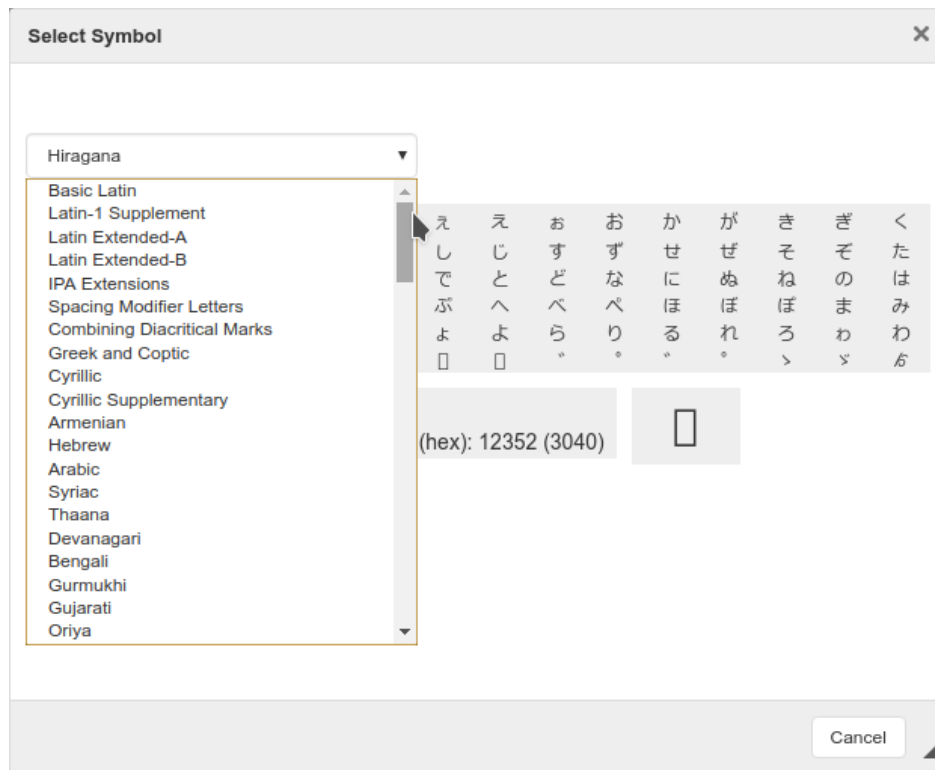
Insert Horizontal Line

Inserts a horizontal line that takes up the entire width of the usable editor area.

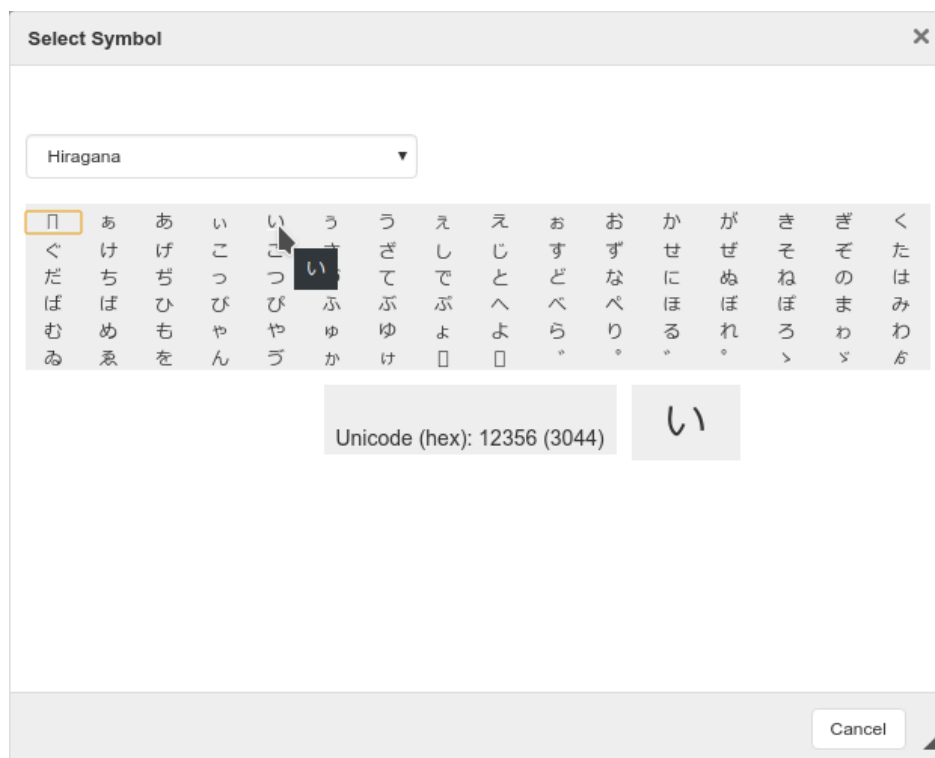
Insert Symbol

 Opens a dialog to insert any symbol printable by the user's computer.


1. Select the alphabet system to look for your symbol from the drop-down menu to the top left.



2. Then click on the symbol you wish to insert to insert it.

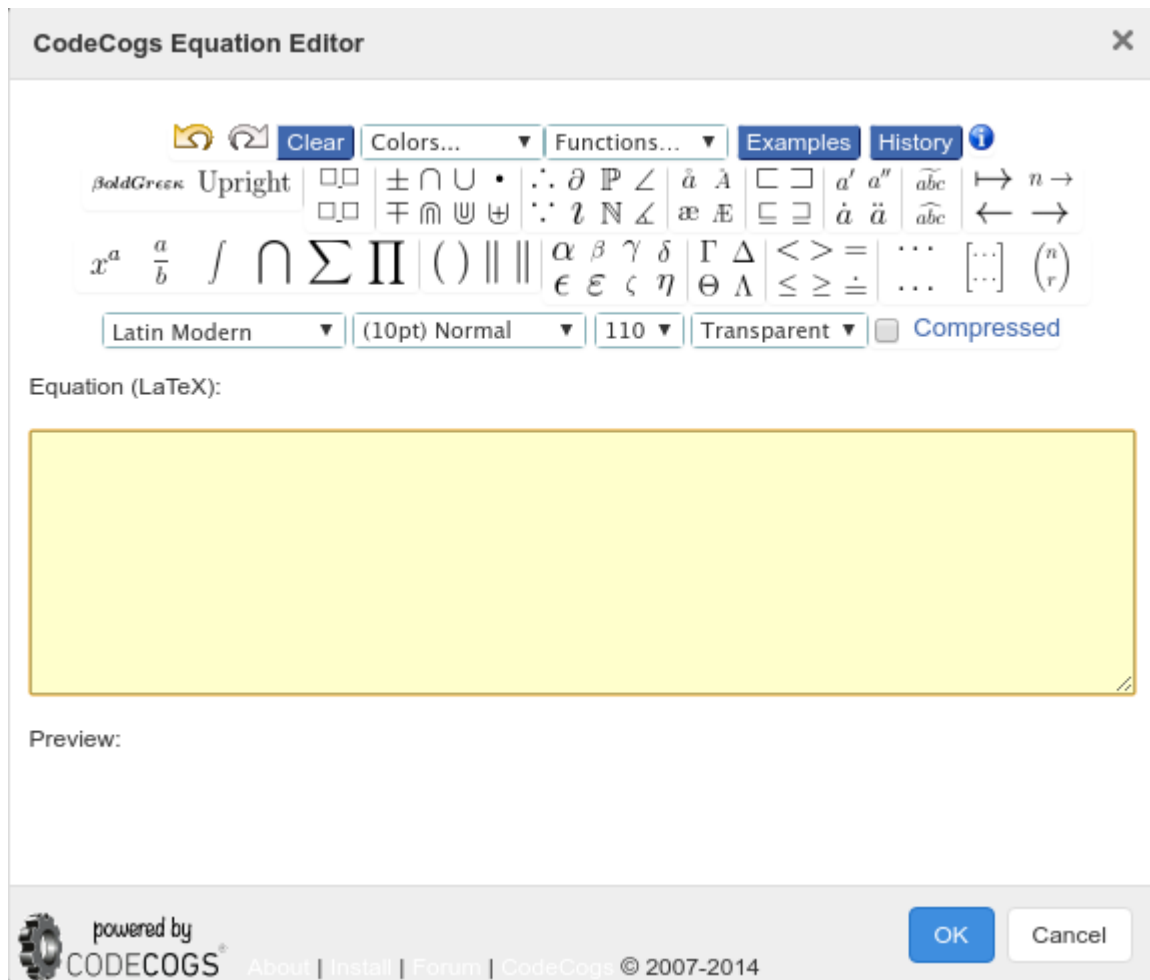


Insert Formula

 Opens the CodeCogs Equation Editor, which allows users to create mathematical equations and insert them into the editor. The user may press the symbols and lists to the upper area of the equation editor to insert symbols that will form the equation. They may also type it manually into the yellow text box, using LaTeX syntax. Learn more about LaTeX at <https://www.latex-project.org/>.

Above the Editor and bellow the symbols are settings regarding text size. The initial settings are CodeCogs' defaults. Increase font size to get an image with good enough quality for prompting.

Using CodeCogs requires an active internet connection.







CodeCogs Equation Editor.

Font Options


Caps Control

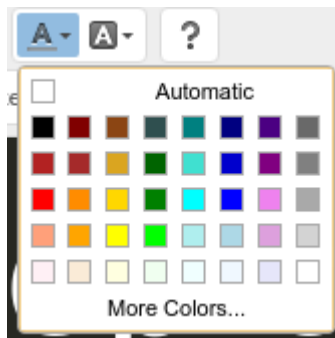


-  Exchanges capital letters for minuscule letters and vice versa.
-  Makes all letters minuscule letters.
-  Makes all letters capital letters.
-  Makes the first letter of each word capital, and the letters that follow, minuscule.

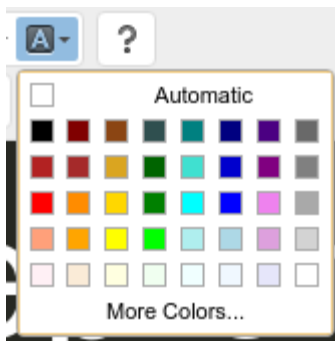
Font Color & Highlight



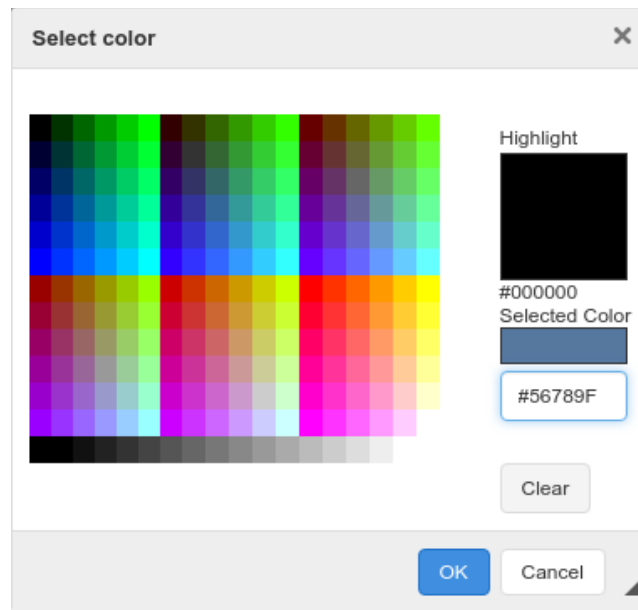
-  Drops down a color picker to choose font-color.



-  Drops down a color picker to choose highlight color.




When you press “More Colors” on either color picker, the following color picking dialog shows up:

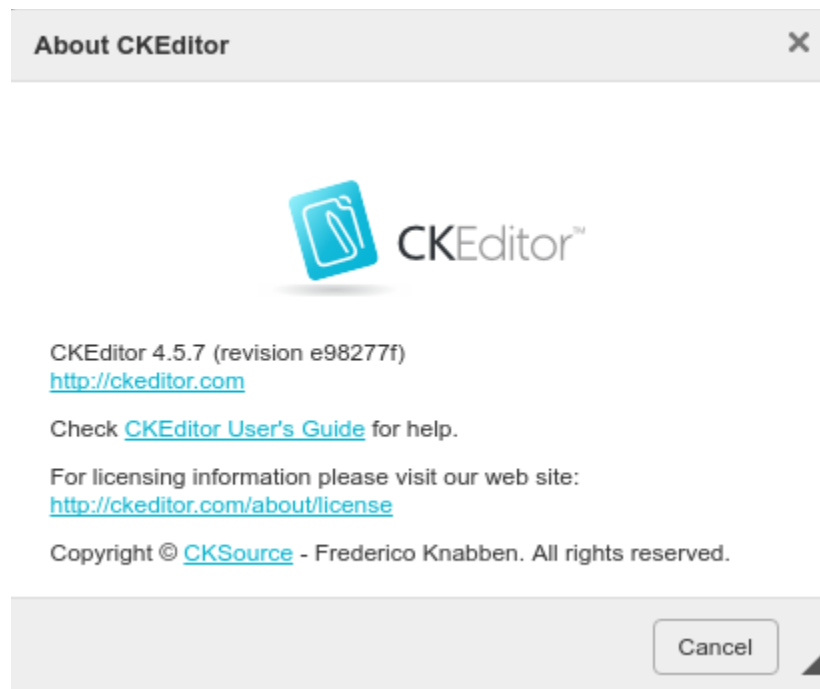


Color picking dialog

To choose a color that's not in the screen, type a hexadecimal color on the input field bellow the "Selected Color" preview and press enter for the preview to update.

About CKEditor

 Shows information about the version of CKEditor being used in Teleprompter.



"About CKEditor" screen in Imaginary Teleprompter versions 2.3.x

Anchor Flag



Pops an input to create HTML anchors in the document.

Please see [How to use anchor shortcuts](#) for more information about how to make use of this when prompting...

Font Options (continuation)

Font Style

B *I* U ~~S~~

- **B** Makes selected text, or text typed from the keyboard's cursor position, appear in **Bold**.
- *I* Makes selected text, or text typed from the keyboard's cursor position, appear in *Italics*.
- U Makes selected text, or text typed from the keyboard's cursor position, appear Underlined.
- ~~S~~ Makes selected text, or text typed from the keyboard's cursor position, appear ~~Strike-through~~.

Subscript & Superscript

x_2 x^2

- x_2 Makes selected text, or text typed from the keyboard's cursor position, _{subscript}.
- x^2 Makes selected text, or text typed from the keyboard's cursor position, ^{superscript}.





Remove Formatting



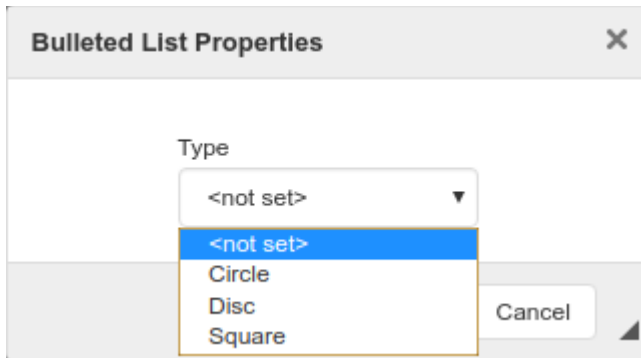
Removes all formatting from selected text.

Bullets & Numbering

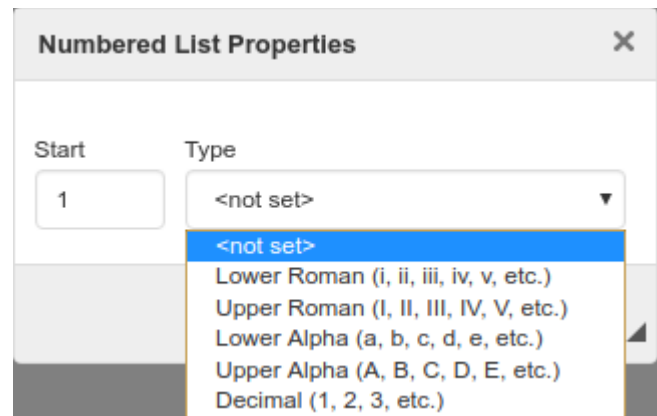


-  Starts an ordered list at the beginning of the line indicated by the keyboard's cursor.
-  Starts a bullet list at the beginning of the line indicated by the keyboard's cursor.
-  Removes indentation from the current paragraph.
-  Indents the current paragraph.

Right-click on a list to and select “Bulleted List Properties” or “Numbered List Properties” to control other details, such as the starting number of the bullet art.







Bulleted List Properties, listing possible bullet decorators.



Numbered List Properties, starting number set to one, and listing possible numbering systems.



Text Orientation



-  Orients the current paragraph towards the left.
-  Orients the current paragraph towards the center.
-  Orients the current paragraph towards the right.
-  Uses justified orientation for the current paragraph. We recommend not to use this feature frequently, as it makes text harder to read.

LTR & RTL Paragraphs



-  Formats the current paragraph for left to right typed languages.
-  .Formats the current paragraph for right to left typed languages


Paragraph Styles



Assigns pre-made styles to the line indicated by the keyboard's cursor.

Available styles include: headings, code containing boxes and DIV containers.

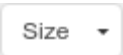
Font Family

 Assigns a font to the currently selected text, or text typed from the keyboard's cursor position.

Available fonts included in Teleprompter 2.3.x include:

- Arial
- *Comic Sans*
- Courier New
- Georgia
- Lucida Sans Unicode
- Tahoma
- Times New Roman
- Trebuchet MS
- Verdana


Font Size

 Assigns the font-size for the currently selected text, or text typed from the keyboard's cursor position.

Possible font size values go from 0.7 to 3.0 times the base size.

Font sizes in Imaginary Teleprompter use the *em* measurement unit, which is a unit of measurement relative to a pre-indicated font-size. This allows scaling the entire text of a document dynamically using the [Font Scale setting](#), it also allows properly synchronizing instances across screens with different scales and resolutions.

Line Height

 Assigns the line height for the currently selected text, or text typed from the keyboard's cursor position.

Possible line height values go from 1.0 to 2.0 times the base size.

Line heights in Imaginary Teleprompter use the *em* measurement unit, which is a unit of measurement relative to a pre-indicated font-size. This allows scaling the entire text of a document dynamically using the [Font Scale setting](#), it also allows properly synchronizing instances across screens with different scales and resolutions.

Word & Paragraph Counter

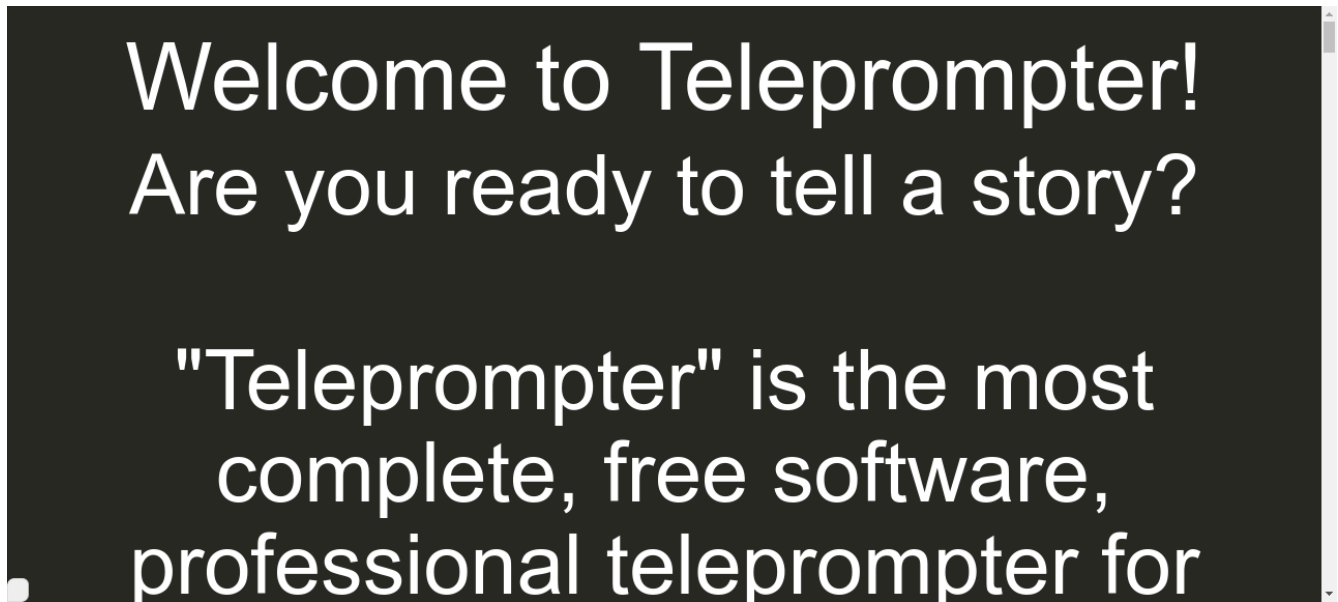
Counts the number of words and paragraphs in the document.

Paragraphs: 1, Words: 7

WYSIWYG Editing View

Allows the user to edit and see the contents as they would be prompted on screen. Clicking on the editor's contents places the keyboard's cursor there, allowing the user to insert new contents.

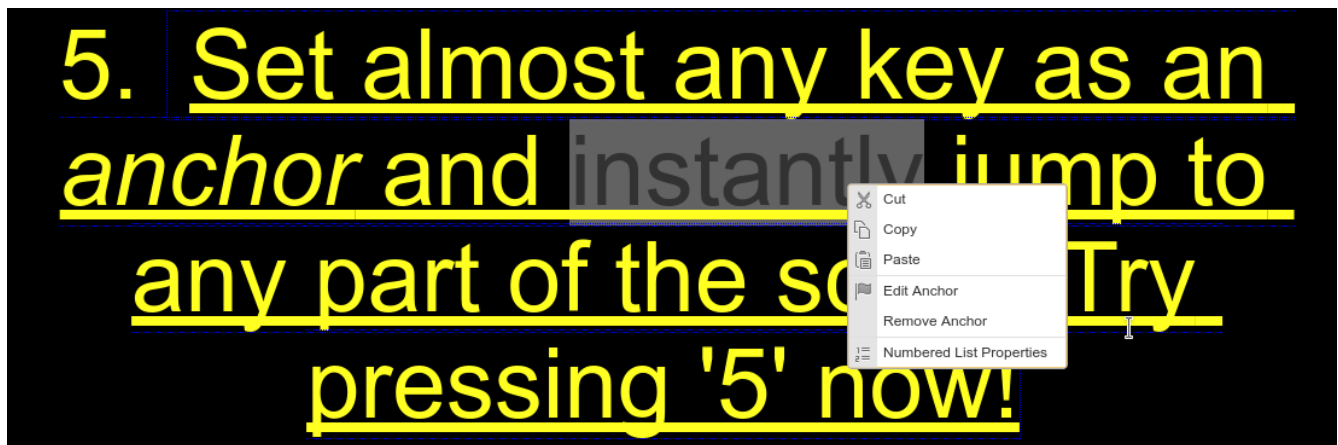
Only when the currently loaded document is the "Instructions" document, the user won't be allowed to edit and instead be prompted to create a new document. See [Creating a New File](#) for more information.



Editor's view with Instructions document loaded.

Context Menus

Right clicking on any part of the editing view loads a context menu. Context menus are called this way because their contents vary depending on the element the mouse or keyboard cursors are pointing to. Some of CKEditor's features are only accessible through context menus.

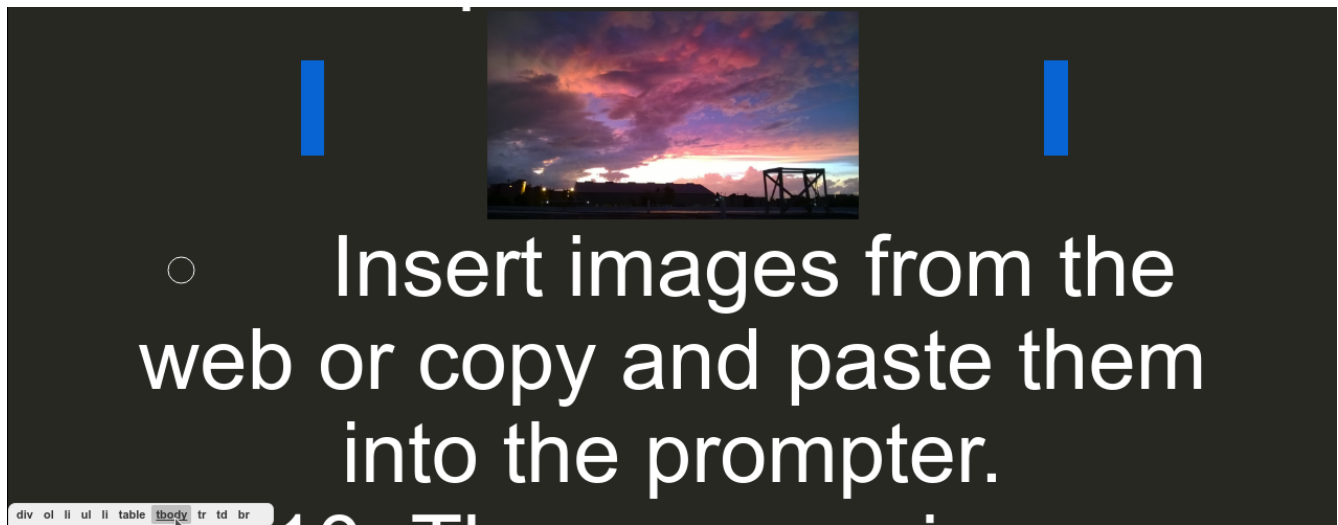


Example context menu. The user is selecting part of an anchored text, that is part of an ordered list.

Editor Status Bar



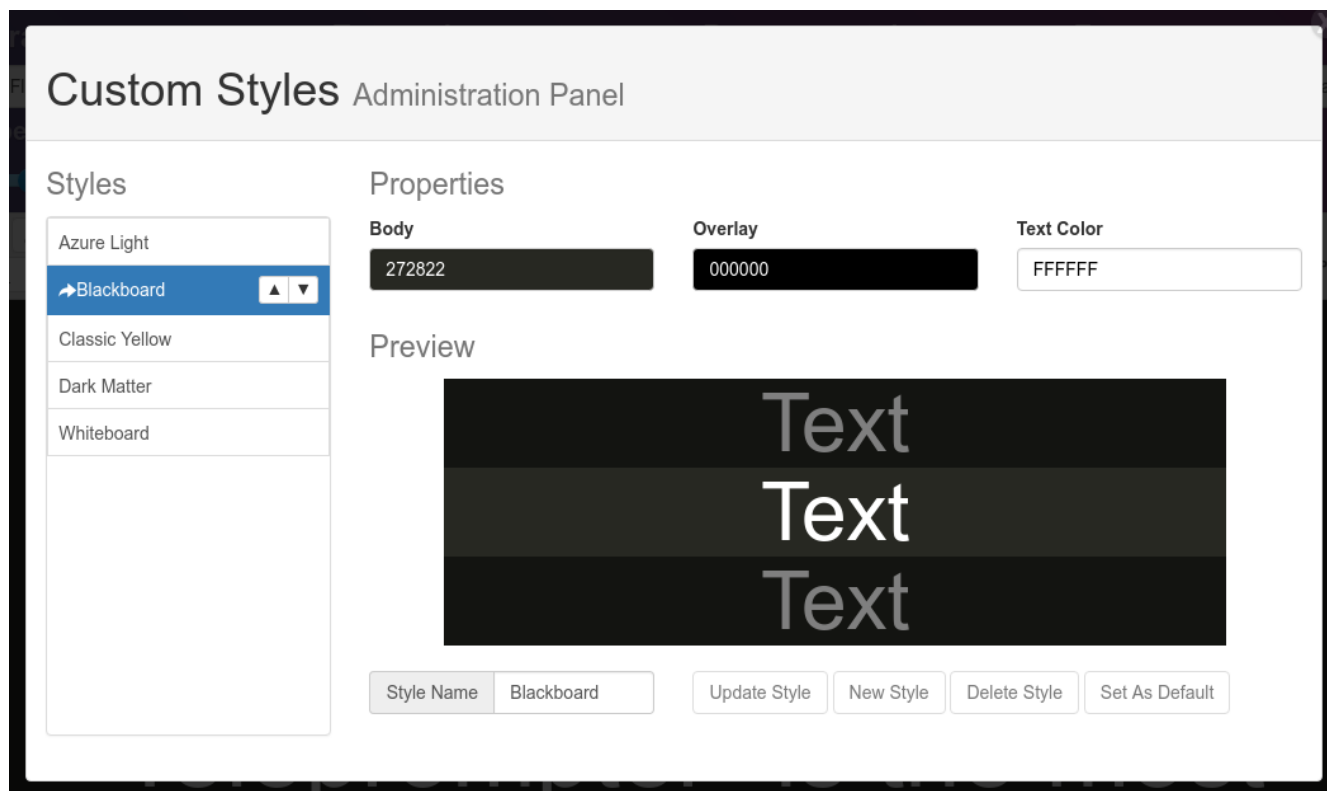
CKEditor comes with a status bar, to the bottom-left of the editor, that indicates the DOM hierarchy of the element positioned behind the keyboard's cursor. This feature is very useful for those knowledgeable in HTML and helps create more complex layouts, as shown in the following image:



Here the user is selecting the “tbody” element of an HTML table, showing that the table contains three cells: one to the left of the image, one containing the image, and another to the right of the image. Placing the image on the second cell of this table is a workaround to present it at a scale lower than 100% of the editable area’s width. Instead, the image takes 100% of the cell that contains it, which takes one third of the editable area.

Custom Styles Administration Pane

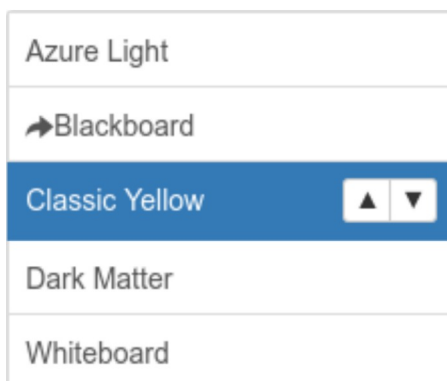
You can create custom color schemes by selecting “Custom Style” from the [Prompter Style](#) options. The following screen appears when you select “Custom Style”:



The screenshot shows the 'Custom Styles Administration Panel' interface. It is divided into two main sections: 'Styles' on the left and 'Properties' on the right. The 'Styles' section contains a list of styles: 'Azure Light', 'Blackboard' (selected with a blue highlight and a right arrow icon), 'Classic Yellow', 'Dark Matter', and 'Whiteboard'. Below this list is an empty text area. The 'Properties' section is further divided into 'Body', 'Overlay', and 'Text Color'. The 'Body' property has a value of '272822', 'Overlay' has '000000', and 'Text Color' has 'FFFFFF'. Below these properties is a 'Preview' section showing three lines of text: 'Text', 'Text', and 'Text', each on a different background color (light gray, dark gray, and black). At the bottom of the 'Properties' section are five buttons: 'Style Name' (with the text 'Blackboard' next to it), 'Update Style', 'New Style', 'Delete Style', and 'Set As Default'.

Create a New Style

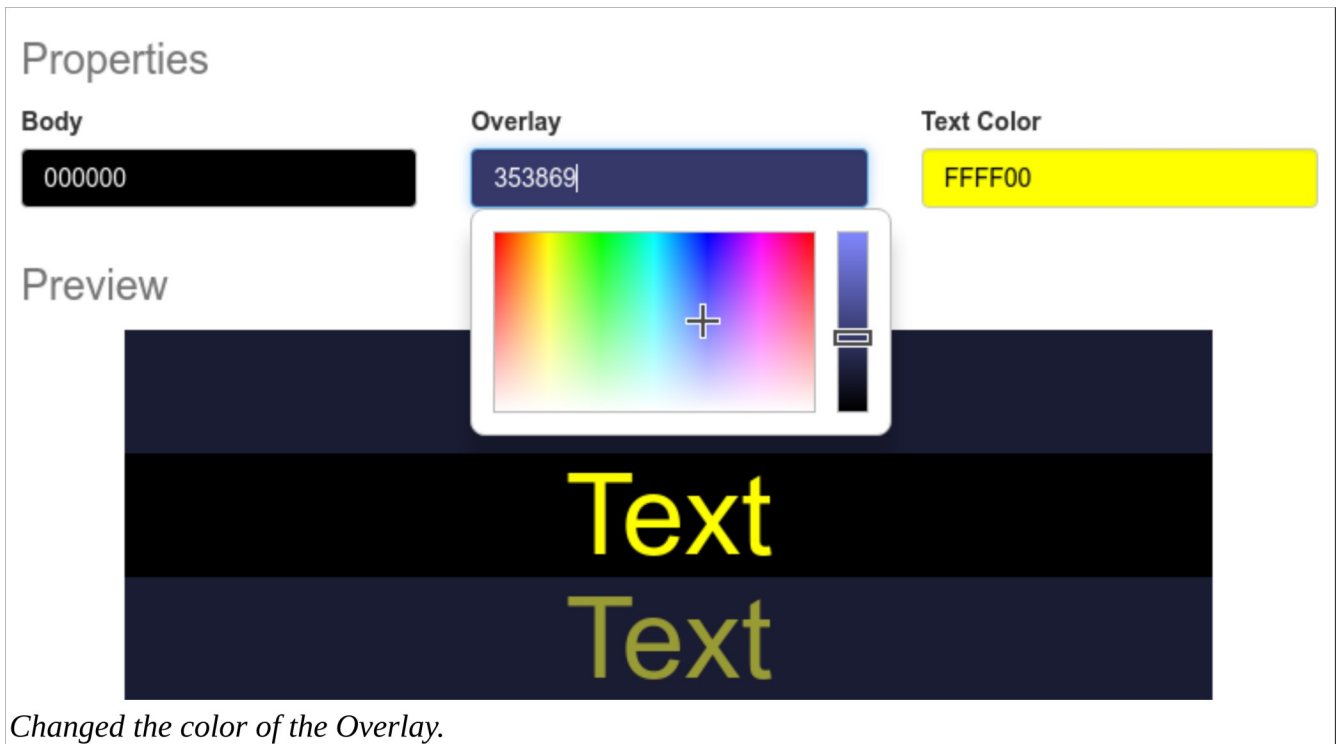
1. Select a style that you want to reference from on the “Styles” selection column, at left.



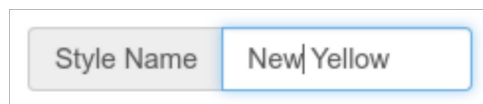
This is a close-up of the 'Styles' selection column. It shows a list of styles: 'Azure Light', 'Blackboard' (with a right arrow icon), 'Classic Yellow' (highlighted in blue with up and down arrow icons), 'Dark Matter', and 'Whiteboard'.

Selected "Classic Yellow" as reference.

2. Click on each of the color bars in the properties section to open the color picker and choose your colors.



3. On the bottom of the Custom Styles pane, click on the text box to the right of "Style Name" and type a name for your style.



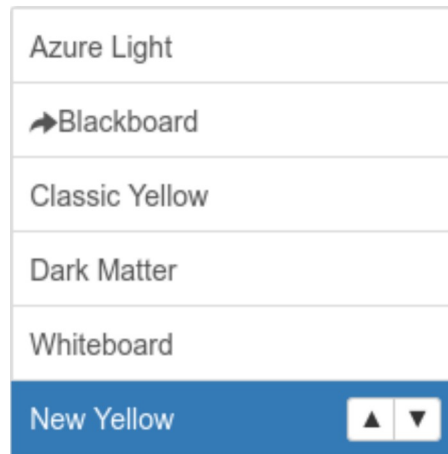
4. Once you've named the style, click on the "New Style" button to the right of "Style Name", to create it.



Update a Style

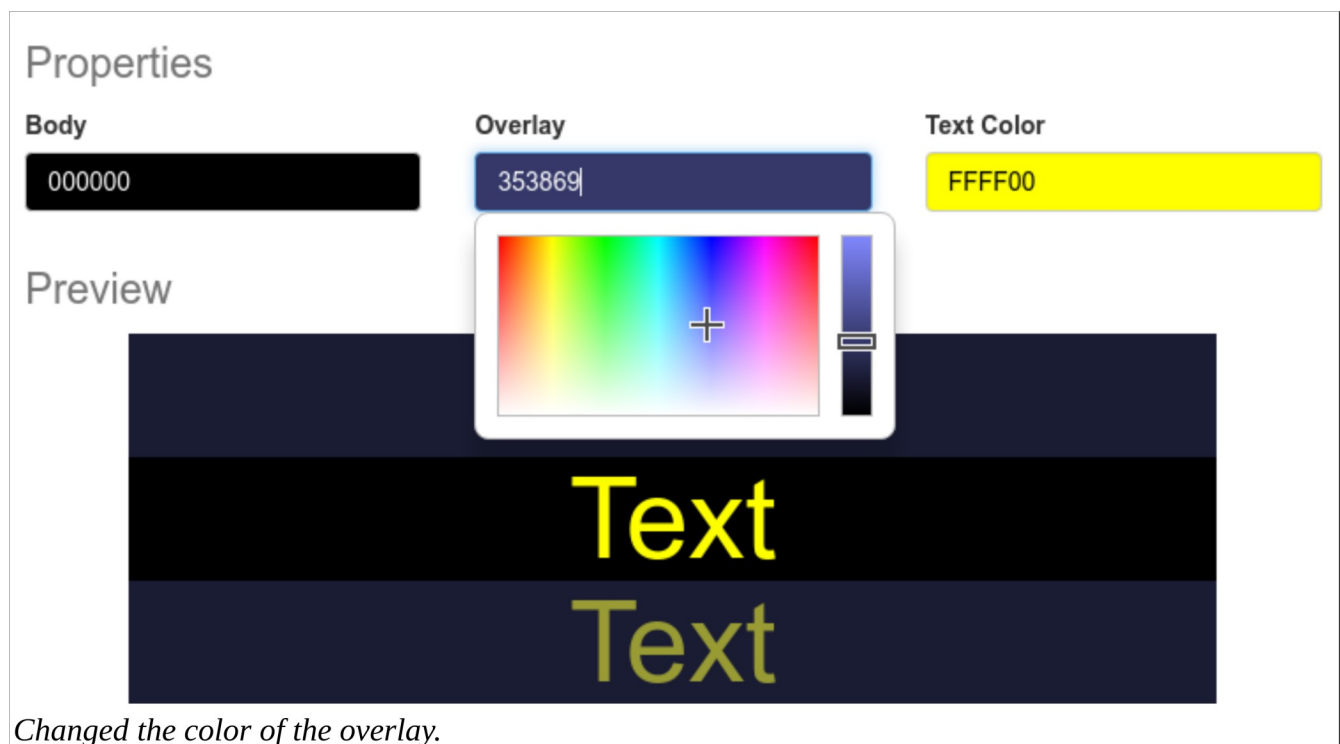
Say you missed step 2 while creating a new style, or you or your talent don't like how it looks in practice all you have to do is update its color pallet. Please note that only custom styles can be updated and presets cannot.

1. In the Custom Styles Administration Pane, select the style you want to update from on the "Styles" selection column, at left.



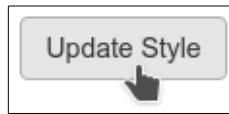
Selected "New Yellow" as the style to update.

2. Click on each of the color bars in the properties section to open the color picker and choose your colors.



Changed the color of the overlay.

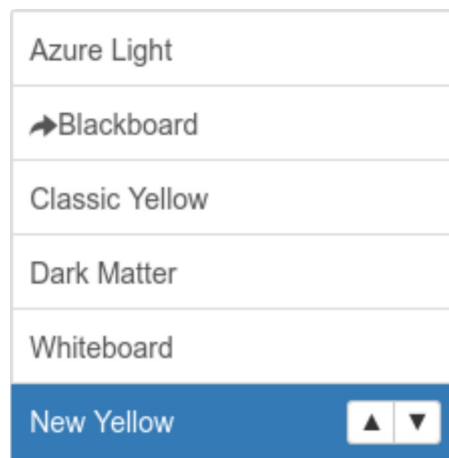
3. Once you've changed your pallet, click on the "Update Style" button to the right of "Style Name", to save your changes.



Set Default Style

The "Default Style" is the style that gets loaded when the program gets started. You may change the default style to any style, preset or custom, following these steps:

1. In the Custom Styles Administration Pane, select the style you want to update from on the "Styles" selection column, at left.



Selected custom style "New Yellow" to set as default.

2. Click on the "Set As Default" button to the bottom right of the Custom Styles Administration Pane.



Reorder Styles

*Warning: Version 2.3.1 contains a bug that prevents changes in order from manifesting when prompting. We apologize for the inconvenience. Please refrain from using this feature.

1. In the Custom Styles Administration Pane, select the style you want to move from on the "Styles" selection column, at left.
2. After selecting a style, two arrows will appear in the selection area.



3. Press the Up arrow to move this style before the previous style or press the Down arrow to move this style after the following style.

Custom Styles Administration Panel

Styles

Fantablack ▲ ▼

Azure Light

➔New Yellow

Blackboard

Dark Matter

Classic Yellow

Whiteboard

Properties

Body

000000

Overlay

000000

Text Color

000000

Preview

Style NameFantablack

Update Style

New Style

Delete Style


Set As Default

Custom styles sorted around.

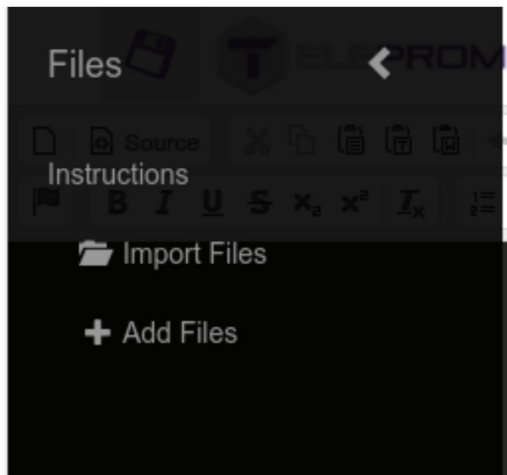
File Options

Create a New File

Imaginary Teleprompter loads the instructions file by default, in order to allow new users to learn to use the program quickly.

1. You can begin to create a new file in the following ways:
 - If the instructions are open, you can begin typing on the editor, Teleprompter will prevent any attempts to modify the instructions and prompt you to type the title of a new file.
 - You may open the Files side pane by pressing the Floppy icon  at the top left of the [Editor Window](#). Then click on “Add Files”, you’ll be prompted to type the title of a new file.

The Files pane will show to the left of the screen:

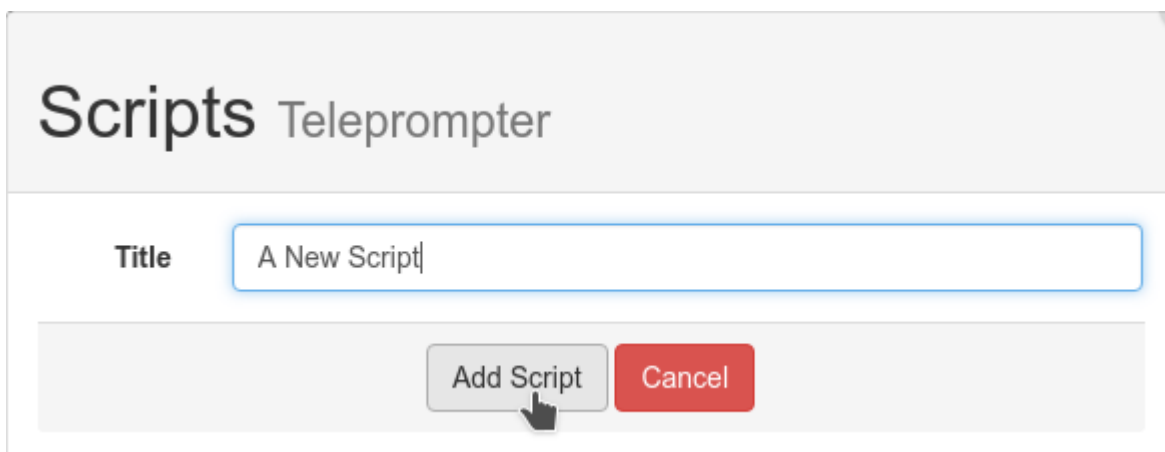


Files side pane

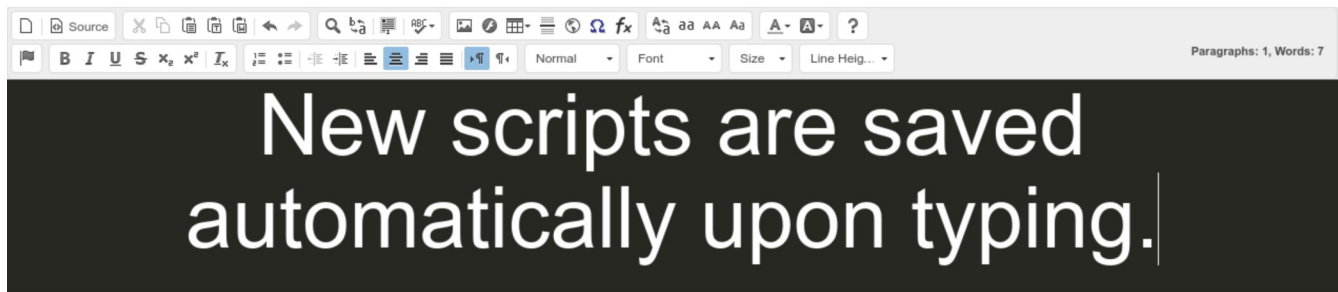


"Add files" option

2. You’ll be prompted to write a title for the new document. Once done typing press Enter or click on “Add Script”. If you don’t want to create a new document, press “Cancel” on the screen or the “Esc” key on your keyboard.




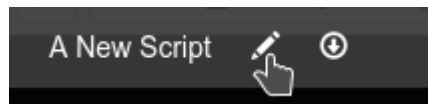
Saving Files



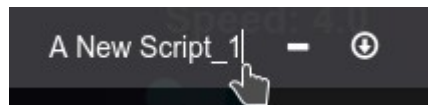
Scripts are saved automatically upon typing into the editor.

Edit File Name

1. Open the Files side pane by pressing the Floppy icon. 
2. Click the pencil icon next to the file you want to edit.




3. Type the changes to your title.

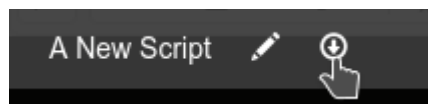


4. Press the “Enter” key when done to save the changes.

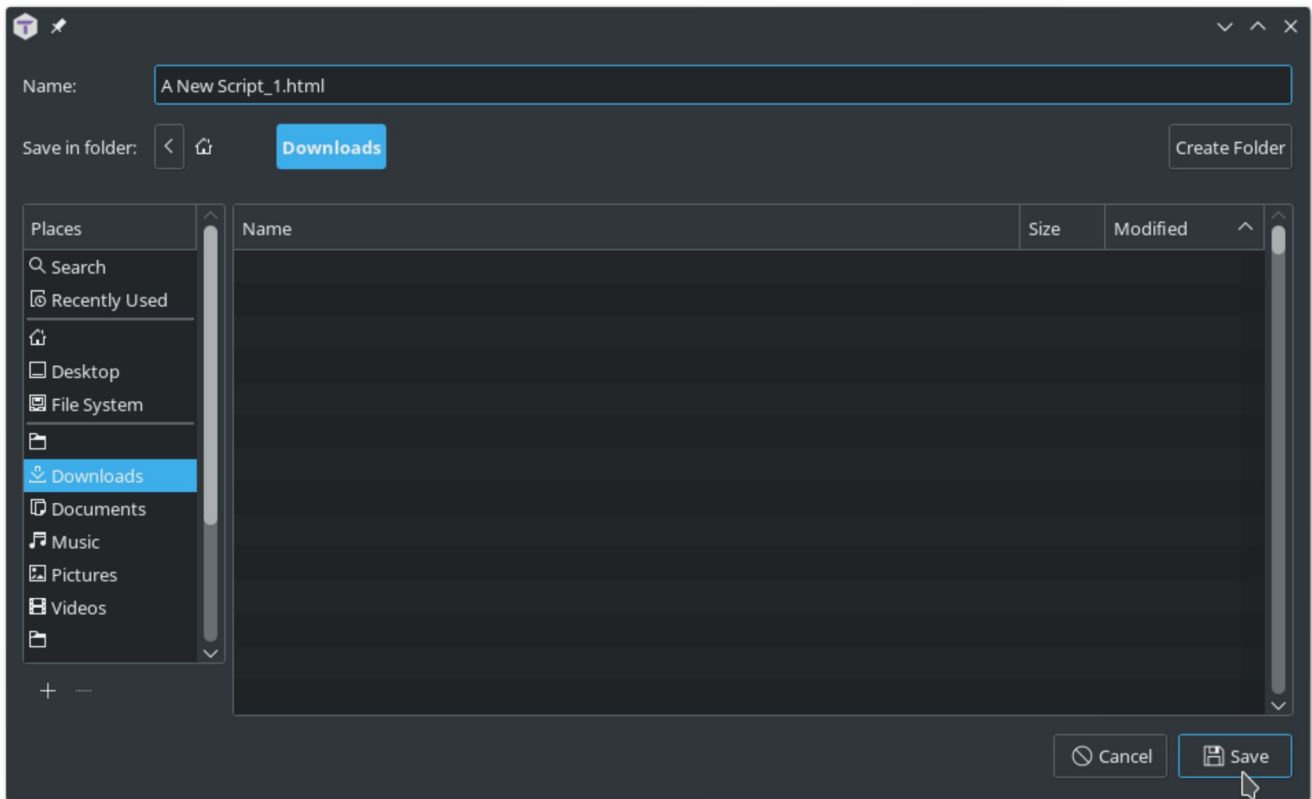
Moving Files Across Machines

On the Computer We’re Copying From...

1. Open the Files side pane by pressing the Floppy icon. 
2. Click the download icon (arrow signaling bottom inside of a circle) next to the file you want to move to another machine.




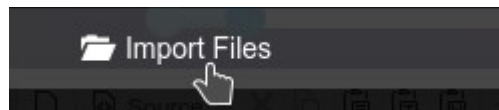
3. A save dialog will appear. The dialog will vary depending on your platform.



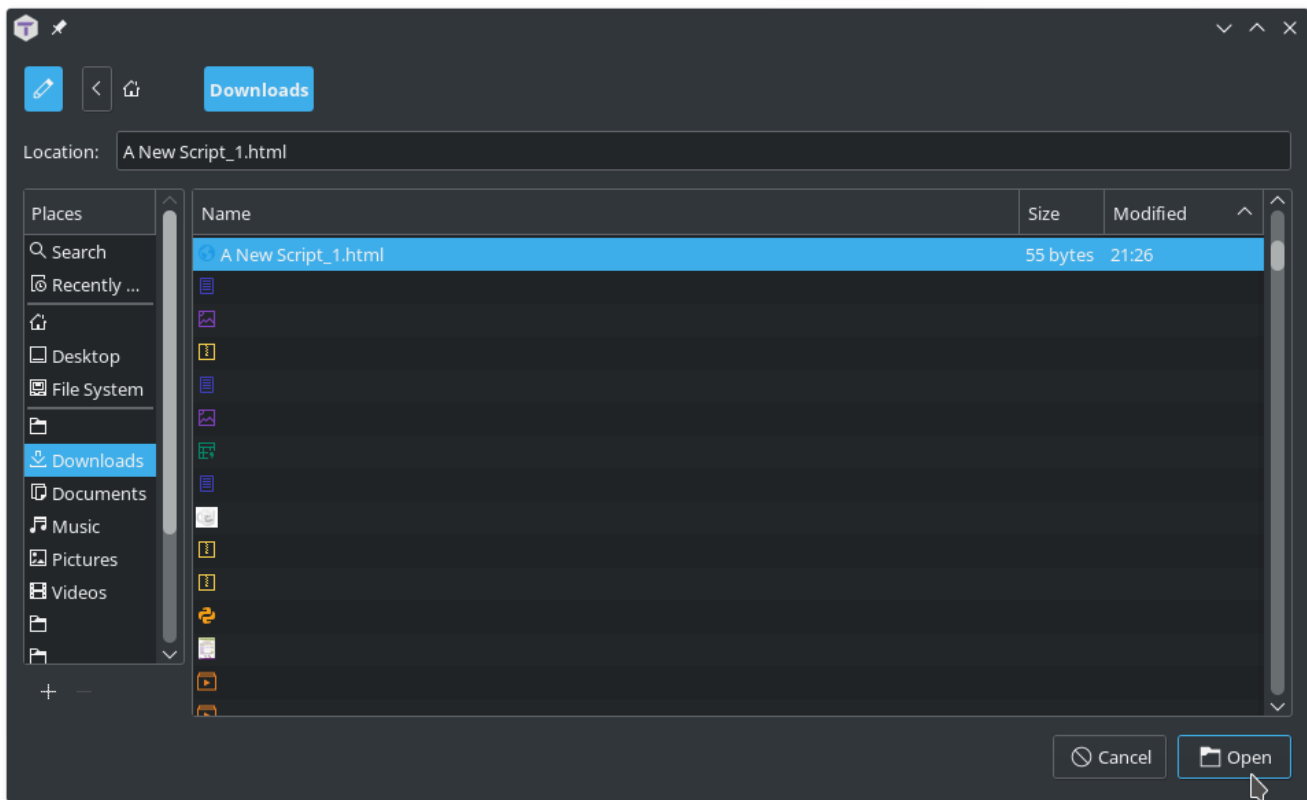
Save dialog under Linux with KDE.

On the Computer We're Copying To...

1. Open the Files side pane by pressing the Floppy icon. 
2. Click on the Import Script option.



3. A file dialog will load. The dialog will vary depending on your platform. Select all files you intend to import and click "Open". Unsupported files will not be opened. HTML files that do not come from Imaginary Teleprompter may not load properly. See [Supported File Formats](#) for more information...



4. Imported files get their names from the name of the files' name in the filesystem.



Prompter Window

Input Controls

Velocity Controls

- Increase velocity : Down, S
- Decrease velocity : Up, W

Font Scale Controls

- Increase font scale : Right, D
- Decrease font scale : Left, A

Timer & Playback Controls

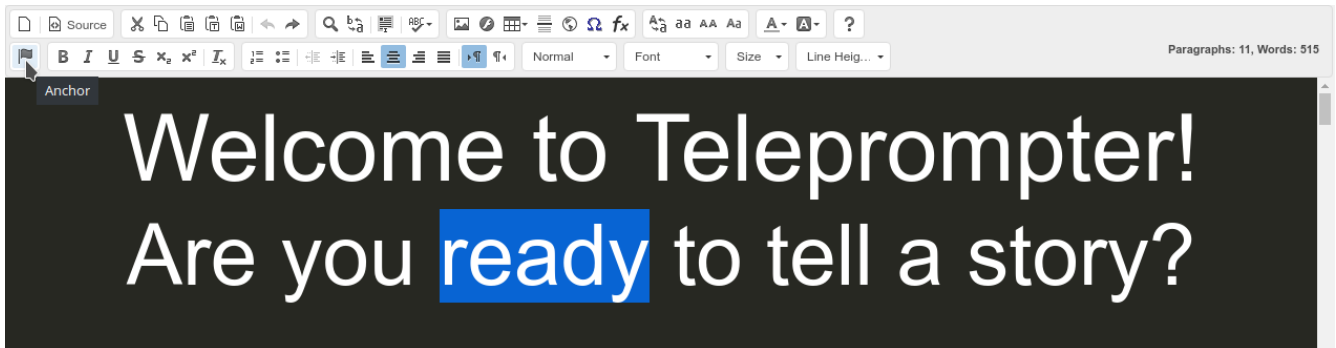
- Reset : Backspace
- Play/Pause : Space bar

Unique Inputs

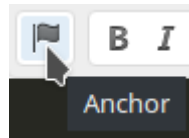
- Synchronize teleprompter instances : . *(use with care)*
- Start prompting : F8
- End prompting : ESC
- Enter/leave full screen : F11

How To Use Anchor Shortcuts

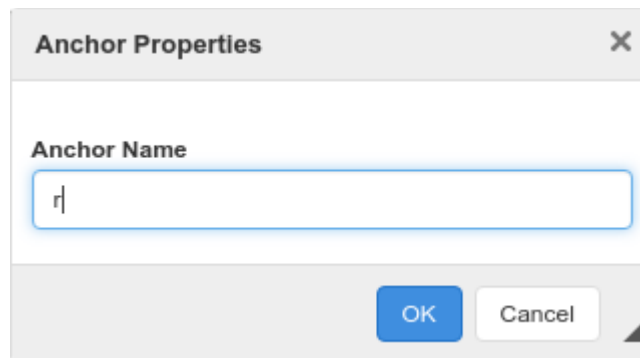
1. Select a keyword or line you want to jump to on your text in the editor.



2. Click on the "Anchor Flag Icon", on the [editor's toolbar](#).



3. A box named "Anchor Properties" should have appeared. Type any single key of your choice and click 'OK'.



Note: preassigned keys, such as WASD and Space bar will be ignored.



A thin dashed blue line now surrounds the previously selected area, marking it as anchored.

4. Repeat steps 1 through 3 for as many shortcuts as you wish.
5. When prompting, press the shortcut keys to jump into the desired locations.

Note that the anchored contents will jump to the bottom part of [Focus Area](#) (upper part if using vertical or double [Mirroring Modes](#)).

Support and Services

Distribution Contract: Inclusion with Teleprompter Hardware

While the GPL permits to sell copies of the software, to cover distribution costs and such, Imaginary Sense does not charge for its binaries. As a matter of fact, Imaginary Teleprompter may be included with teleprompter hardware manufactured and sold by third parties for free, as long as these parties comply with all the requirements of the GPL 3.0. If a party was interested in distributing Imaginary Teleprompter but cannot comply or does not wish to comply with the requirements of the GPL 3.0, they need to contact Imaginary Sense, and only Imaginary Sense, to reach an agreement for a distribution contract.

Reasons to enter in such an agreement could include:

1. To change Imaginary Teleprompter's branding
2. To remove credits
3. To not include a copy of the source code along the binaries
4. To avoid having to comply with the anti DRM restrictions of the GPL 3.0
5. To not disclose necessary changes required to support your hardware

Support Service Contract

If you were interested in us adding a new feature or supporting a specific platform to satisfy your particular needs, you may wish to contact us regarding a feature request and inquiry about a support contract. We should reach an agreement that could range from Imaginary Sense sponsoring your productions, to paying a fixed or periodical fee for the development of a feature.

Agreements vary depending on how much it costs us to develop such feature, how much value it adds to other users, whether these modifications are for personal or commercial use, and whether the modifications are to be shared under the GPL 3.0.

Annex

Changes in this Cycle

This release is all about usability. We've added more features and bring you a cleaner interface. Our branding changed from Imaginary Films to Imaginary Sense but we remain the same team, looking forward to bring great quality multi-platform software.

2.3.3

1. Fixed ghost process on window closed bug.

2.3.2

2. Added full 32 bit support
3. Added experimental ARM7L support.
4. Patched Electron security issues.

2.3.1

5. Fixes a bug that prevented the prompt's scroll from sometimes reaching the end.

2.3.0

1. Increased Performance
2. New File Management Features
3. New Speed Precision Controls
4. New Timer
5. New Fast Text Resize
6. New Custom Styles
7. New Focus Modes
8. Improved Instance Synchronization
9. Improved Live Update Feature
10. Improved User Interface

Plans for the Next Two Releases

1. Add support for Safari Mobile on iOS and Google Chrome Mobile on iOS and Android
2. Implement previous and next anchor navigation
3. Remove the editor window's scroll bar
4. Improve editor usability on touch devices
5. Implement unnamed anchors
6. Move footer information and options into About section of window menu
7. Add Full Screen option into window menu.
8. Add multi-platform SVG Emoticon Support
9. Implement software translations
10. Finish implementing image uploads manager
11. Implement generic Bluetooth LE remote control support
12. Improve keyboard and assistance technologies usability through the use of menu
13. Implement Local Spell Check
14. Improve Custom Styles usability
15. RTF file format support