

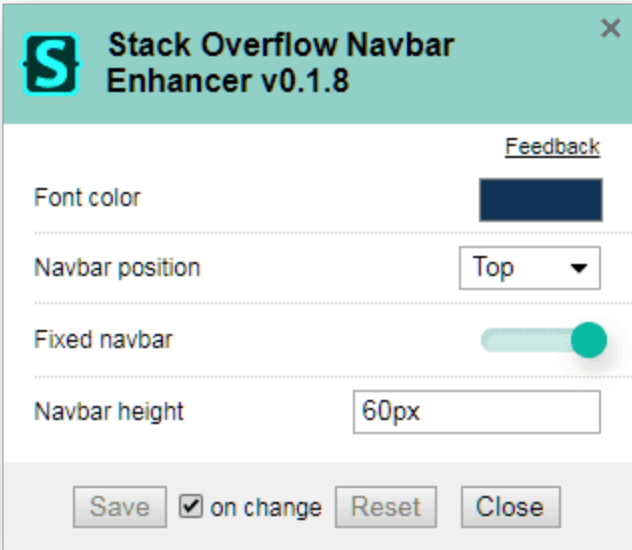
[github.com /openstyles/stylus/wiki/Writing-UserCSS](#)

Writing UserCSS

22-28 minutes

Overview

It's a standard userstyle, optionally [self-hosted](#), with metadata in a `/* comment */` at the beginning, where you can enable various advanced features like [preprocessor](#) and [user-customizable variables](#), which can be switched on the fly in the built-in config UI, for example:



► [View example source](#)

Installation

Open a UserCSS file or URL in the browser and click the Install style button in the Stylus' style installation page that should appear with the style contents and information, see the screenshot below. Optionally untick the [x] Check for updates to disable automatic updates for the style.

► [Screenshot](#)

Caveats:

- The contents must be [plain text](#) and not a [web page](#).
- The URL (or a local file name) must end with `.user.css`, `.user.styl`, or `.user.less`. It works on all sites by default, but you can limit it only to the [known style galleries](#) by disabling Open installer when navigating to a `.user.css` URL in the options of the extension.

Installing local files

- In Firefox, drag'n'drop the file into a browser tab label area or open it using the standard `CtrlO` or `CmdO` shortcut. The installation page will be opened in an additional tab.
- In Chrome/Chromium there are two methods:
 1. Drag'n'drop the file into an open Stylus page like the manager or the popup. Note that automatic updates won't work, use the method 2 instead. You can assign a hotkey to activate the extension so you'd press it to open the popup while dragging the file and drop it inside.
 2. Grant Stylus read-only access to the entire file system and then simply drag'n'drop the file into a new browser tab or open using the standard `CtrlO` (`CmdO`) shortcut the same way it's done in Firefox.

Granting access to local files in Chrome/Chromium

Table of Contents

- [Installation](#)
 - [Installing local files](#)
 - [Live reload \(on-the-fly previewing\)](#)
 - [Reinstallation](#)
- [Managing](#)
- [Configuring](#)
- [Creating](#)
 - [Change default template](#)
 - [Metadata](#)
 - [Choose a preprocessor](#)
 - [Linting](#)
 - [Applies to](#)
 - [Hosting a UserCSS](#)
 - [Badges](#)
- [Editing](#)
- [UserCSS Metadata](#)
 - [@name](#)
 - [@version](#)
 - [@namespace](#)
 - [@description](#)
 - [@author](#)
 - [@homepageURL](#)
 - [@supportURL](#)
 - [@updateURL](#)
 - [@license](#)
 - [@preprocessor](#)
 - [default](#)
 - [uso](#)
 - [less](#)
 - [stylus](#)
 - [@var](#)
 - [type](#)
 - [name](#)
 - [label](#)
 - [default value](#)
- [xStyle compatibility](#)
- [More examples](#)

Even though it's read-only, generally you shouldn't be doing it unless you trust the extension after examining its source code and practices or enjoy the danger.

1. Right-click the Stylus icon, click Manage extension.
...or open the browser menu → Extensions > Stylus → Details.
2. Enable the Allow access to file URLs switch.

Live reload (on-the-fly previewing) when developing styles locally in another editor/IDE

A Live reload checkbox is shown when installing a local file, i.e. a file:// URL. Stylus 2 and newer also support localhost and 127.0.0.1.

Enabling the Live reload checkbox will automatically apply your edits in an external editor/IDE (when the file is saved or auto-saved) to the matching web pages while the installer tab is open. Note that if the style wasn't yet installed in Stylus, live reload will commence only after you click the Install style button.

- While the feature is active, the errors will be shown at the top of the installer page, so if the style isn't applied as expected, either switch back or drag the tab into a separate window stacked beside the main one.
- While the feature is active, there's no need to open this style in Stylus' own editor. If you did it anyway and accidentally forgot to save your edits before enabling live reload, the extension will show a confirmation in the editor.

Chrome/Chromium requires additional setup to use one of the following methods:

1. Drag'n'dropping the file into an open Stylus page.

⚠ Requires Chromium 133 or newer.

⚠ In Chromium 129-132 you can go to chrome://flags and enable Experimental Web Platform features. Note that although the file observation API itself is conceptually safe as it sensibly grants access to an *individual file*, not the entire file system, but the other experimental features enabled by the flag may cause unexpected compatibility problems for websites, in which case see if disabling the flag helps first.

2. Navigating the browser to http://localhost/file.user.css URL.

⚠ Requires you to install a simple local http server.

⚠ Requires Stylus 2 or newer.

3. Enable access to the entire local file system as [explained above](#).

Reinstallation

Opening a UserCSS a second time will reveal a "Reinstall" button. Click to reinstall the UserCSS.

A *disabled* checked checkbox with "Install update from ..." will be shown. This indicates that the installed style is from the same source.

A warning will be displayed if the UserCSS is an older version of the one that is currently installed.

Managing

On the Manage page ("Installed Styles") each UserCSS entry will include one or more of the following icons:



In order from left to right, they represent:

- External link pointing to the source page.
 - Derived from the @homepageURL setting.
 - Hovering over the icon will show the URL in the tooltip.
 - This icon will not be present if no @homepageURL was defined.
- Delete style.
- Update style:
 - This checks the URL from where it was installed.
 - If the "Install update from ..." checkbox was left unchecked during installation, this icon will not be present.
- UserCSS configuration (see the [next section](#)).

Configuring

To configure a UserCSS, go to the Manage page and click on the "Configure" icon (Gear icon seen in the screenshot above) to open the dialog.

The contents of the configure dialog depend on the UserCSS @var `type` settings:

- `text` - A plain text input is provided.
- `color` - A color type input is shown along with a slider to set the opacity.
- `checkbox` - A checkbox will be provided; it is styled as a sliding switch.
- `select` - A dropdown select with preset options will be provided.
- `number` - A plain number input is provided (added in Stylus v1.4.23).
- `range` - A plain range slider is provided along with the current value and any units (if set) (added in Stylus v1.4.23).

Make the desired changes and click:

- "Save" to save your current selections.
- "Reset" to reset the configuration to the default settings.
- "Close" to close the popup.

Creating

To create a UserCSS format style:

- From the Manage page, check the "as UserCSS" checkbox next to the "Write new style" button.
- Click on the "Write new style" button.
- A basic UserCSS template will then be available.

Change default template

- To change the default UserCSS template, create a UserCSS style as described above.
- Within the default template, make any changes to the template to suit your needs.
- Click ▼ inside the Save button to expand the dropdown menu, then select Save as template.

Metadata


All UserCSS styles **require** the following metadata entries:

- @name
- @version
- @namespace
- @license (not required, but essential if you want to share your work)


See the full list of supported [UserCSS metadata](#) below.

Choose a preprocessor


The `@preprocessor` setting will determine how the code will be compiled and therefore what "CSS flavour" you use. These preprocessors can extend the standard CSS syntax by lots of convenient features. It is important to follow the syntax of the preprocessor you picked:


Pure [SASS](#)  syntax is currently not supported.

Linting

If enabled, the CSS linter will automatically switch to use stylelint. This is because stylelint supports a `"sugarss" syntax` , which closely approximates the Stylus language. However, stylelint does not correctly report all issues in the Stylus language format because of minor differences between the syntax.

Applies to

- Add additional entries by:
 - Typing any supported `@-moz-document`  matching function within the editor.
 - Or, clicking an "Add" button within the "Applies to" section.
- Remove entries by:
 - Deleting the `@-moz-document` matching function within the editor.
 - Or, clicking on the "Remove" button within the "Applies to" section. The last entry will not be removed.

Note: The documentation for @document reports that it is currently [only supported in Firefox](#)  and must include a -moz prefix. This is why we always refer to it as @-moz-document within this document.

Hosting a UserCSS

When linking to the .user.css file, you'll need to link to the "raw" version of the style.

- **GitHub Repo:** `https://raw.githubusercontent.com/{USER}/{REPO}/master/{NAME}.user.css`
- **Raw.GitHack:** `https://raw.githack.com/{USER}/{REPO}/master/{NAME}.user.css`
- **Bitbucket:** `https://bitbucket.org/{USER}/{REPO}/raw/HEAD/{NAME}.user.css`
- **GitLab:** `https://gitlab.com/{USER}/{REPO}/raw/master/{NAME}.user.css`
- **GitHub Gist:** `https://gist.github.com/{USER}/{SHA}/raw/{REVISION-SHA}/{NAME}.user.css`
Remove the {REVISION-SHA} to get the latest version.
- **Dropbox:** `https://dl.dropboxusercontent.com/s/{RANDOM}/{NAME}.user.css?dl=0`
Modify the public share link by changing the subdomain from www to dl and adding ?dl=0 to the end.

Special case:

If the linked file doesn't end in .user.css or .user.css? with parameters you have to add a dummy parameter like ?dummy=.user.css to the end of the URL:

- **GitLab Snippets:** `https://gitlab.com/{USER}/{REPO}/snippets/{ID}/raw?dummy=.user.css`

Badges

Add a badge to let users know your style can be installed directly

- Install directly with Stylus


```
[![Install directly with Stylus](https://img.shields.io/badge/Install%20directly%20with-Stylus-00adad.svg)](MY.USER.CSS)
```
- Install directly with Stylus


```
[![Install directly with Stylus](https://img.shields.io/badge/Install%20directly%20with-Stylus-285959.svg)](MY.USER.CSS)
```
- Install directly with Stylus


```
[![Install directly with Stylus](https://img.shields.io/badge/Install%20directly%20with-Stylus-238b8b.svg)](MY.USER.CSS)
```

Replace MY.USER.CSS with a direct link to your style in user.css format (e.g. GitHub's raw URL - `https://raw.githubusercontent.com/{user}/{repo}/master/my-style.user.css`).

Don't forget that the file must end in .user.css otherwise Stylus won't recognize it as such.

Editing

For general help on editing, see [Writing styles](#).

Usercss Metadata

@name

The name of your UserCSS style. **Mandatory.**
The combination of @name and @namespace must be unique for each UserCSS.

Leave @name empty to create a template for new UserCSS styles.

Example:

```
@name example site awesome dark
```

@namespace

The namespace of the UserCSS. **Mandatory.**
Helps to distinguish between styles with the same name. Usually, the author's nickname or homepage.
Can contain spaces and special characters.


The combination of @name and @namespace must be unique for each UserCSS.

Example:

```
@namespace https://github.com/octocat
```

@version

The version of your style. **Mandatory.**
This is used for the update check.

Follow [semver](#)  versioning for maximum compatibility with other extensions or style-processing utilities.

If you use Stylus 1.5.18 or newer, @version can contain any amount of dot-separated digits and CalVer (calendar versioning) with its 0-padded numbers.

Example:

```
@version 1.0.0
```

@description

A short significant description.

Example:

```
@description A UserCSS style to make people happy
```

@author

Info about the author of the UserCSS style.

Order of parameters is compulsory: name [<email optional>] [(URL optional)].

If an email address is included (wrapped in angle brackets), it will be shown as a "mailto" link on the install page.

If a URL is included (wrapped in parentheses), it will be shown as an external link on the install page.

Example:

```
@author My Name <my-email@my-site.com> (http://my-site.com)
```

@homepageURL

The project's homepage that is used in Stylus' Manage and Edit pages to link to UserCSS source. This is not an update URL. The update path is the URL the UserCSS was installed from.

Example:

```
@homepageURL https://github.com/octocat/Spoon-Knife
```

@supportURL

The URL the user can report issues to the UserCSS author. Displayed as "Feedback" link in the style's UserCSS configuration popup.

Example:

```
@supportURL https://github.com/octocat/Spoon-Knife/issues
```

@updateURL

When defined, this URL is used when updating the style otherwise the style is updated from where it was installed.

Example:

```
@updateURL https://github.com/octocat/Spoon-Knife/my-style.user.css
```

@license

Include a license based on the [SPDX license identifier](#) . **Essential.**

Example:

```
@license CC-BY-SA-4.0
```

If `no license` [↗](#) is included, it will be assumed that no one can use, copy, distribute, or modify the work.

@preprocessor

Applies a CSS preprocessor.
The preprocessors listed below are supported.

default

The default mode with [standard CSS syntax](#) [↗](#) is used if

- `@preprocessor default` is set or
- `@preprocessor` is missing and no variable is defined by `@var` or `@advanced`.

With the default preprocessor set, UserCSS variable definitions like

```
@preprocessor default
@var text  myFontSize      "Font size"      16px
@var color myBackgroundColor "Background color" #0b0
```

will generate a variable list, which is concatenated to each section of the style:

```
:root {
  --myFontSize: 16px;
  --myBackgroundColor: #0b0;
}
```

uso

This mode inspired by the [userstyles.org syntax](#) [↗](#) is used if

- `@preprocessor uso` is set or
- `@preprocessor` is missing and *there are* variables defined by `@var` or `@advanced`.

In addition to standard CSS features, [userstyles.org placeholders](#) [↗](#) (`/*[[variable-name]]*/`) will be replaced with their values recursively.

less

The UserCSS will be compiled following the [less syntax](#) [↗](#).

stylus

This setting will use the Stylus-language preprocessor on the UserCSS.

The UserCSS variable definition

```
@var text myFontSize "Select a font size" 14px
```

will be compiled to a Stylus-lang definition

```
myFontSize = 14px
```

which is then concatenated to the source code and compiled with Stylus-lang.

@var

Defines a live-switchable variable which will be compiled to valid CSS according to the preprocessor that is set.

`@var` is interchangeable with `@advanced` (used by `xStyle`).

The `@var` includes the following properties in this order:

```
@var <type> <name> <label> <default value>
```

All properties are required, but for the select type some [can be combined](#) if they're the same.

Example:

```
/*
@var <type> <name>   <label>           <default value> */
@var text  myBorder "Border style" 3px dotted blue
```


Do *not* include the keyword `!important` into the variable value.

Within the style, refer to the above variable as follows:

type

Choose one of these types to be used within the configuration menu

- `text` - Provides a plain text input.
- `color` - Provides color type input with alpha slider.
- `checkbox` - Provides a checkbox styled as slide switch.
- `select` - Provides a select element (same as dropdown and image types provided by `xStyle`).
- `range` - Provides a range slider with the current value, and any units, displayed (added in Stylus v1.4.23).
- `number` - Provides a plain number type input element with user input clamped to any min, max or step values (added in Stylus v1.4.23).

name

The name of the variable. Must match regular expression `/[\w-]+/` which means letters (small and caps), numbers, `_` and `-`.

label

The label describing a variable shown in the configuration popup. Must be enclosed with quotes (`'` or `"`) if the string contains spaces or special characters.

default value

The default value for the variable. Having a default value is mandatory.

- `text` - Include a string value.
If the string contains a colon (`:`), wrapping in single quotes *and* additional wrapping in double quotes is required.
- `color` - Use a hex (3, 4, 6 or 8-digit hex codes), `rgb` or `rgba` color.
- `checkbox` - Set `1` for checked or `0` for unchecked.
- `range` - Include an array of values (added in Stylus v1.4.23):
 - Set as `[default, min, max, step, units]`.
 - All items but the default value are optional.
 - The default, min, max and step values *must* be numeric (or null), or it will cause a parsing error.
 - Values set to null are ignored.
 - The units value - framed in (double)quotes - may be added in at any position since it's non-numeric; but the default, min, max and step values are set in order, e.g. `['px', 5, 0, 10, 1]`, `[5, 0, 'px', 10, 1]` or `[5, 0, 10, 'px', 1]`, etc, *will all* set the default to 5, min to 0, max to 10, step to 1 and the units to px.
 - Only the first units value entry will be used, any additional (non numeric) entries will be ignored, e.g. `[5, 0, 10, 1, 'px', 'em']` will only set the units to px. The `em` entry is removed.
 - ⚠ because we're parsing the meta data as JSON, all fractional values *must* include an integer value, e.g. `[.5]` will cause a parse error, so use `[0.5]`
 - ⚠ this variable option is not (yet) supported by other userstyle managers, so please make sure your users are aware of this fact.
- `number` - Same as `range` (added in Stylus v1.4.23). The only difference is that a plain number type input is rendered.
- `select`
 - Use a JSON format for the values set in the select's options.
 - Each `key:value` pair represents an option.
 - Default options can be indicated by adding an asterisk to the end of the key name (added in Stylus v1.4.23), e.g.
 - `["Arial", "Consolas*", "Times New Roman"]` ("`Consolas`" will be set as the default).
 - `"nearWhite*": "#ddd"` ("`nearWhite`" is set as the default).
 - `"near_white*:Near White": "#ddd"` ("`near_white`" is set as the default).
 - The key is composed with `OPTION_NAME:OPTION_LABEL`. Option names must be unique from each other:

```
@var select fontBkgd "Background" {
  "near_black:Near Black": "#222",
  "near_white:Near White": "#ddd"
}
```

- If the option name, label, and value are the same, it can be shortened to an array. For example:

```
/* ...
@preprocessor stylus
@var select fontName "Font name" ["Arial", "Consolas", "Times New Roman"]
... */
@-moz-document domain("example.com") {
  body {
    font-family: fontName !important;
  }
}
```

- If the value equals the name, it can be shortened to an array. For example:

```
/* ...
@preprocessor stylus
@var select theme "Theme" ["dark:Dark theme", "light:Light theme"]
... */
@-moz-document domain("example.com") {
  body {
    if theme = dark {
      ...
    } else if theme = light {
      ...
    }
  }
}
```

- Option name can be omitted:

```
@var select fontBkgd "Background" {
  "Near Black": "#111",
  "Near White": "#eee"
}
```

In this case, the option name (key) is used as option label.

Settings will be kept during updates if the option names / option keys don't change.

Use backtick (`) to enclose multi-line string.

Variable examples

```
/* ==UserStyle==
@name          test
@description   UserCSS example
@namespace     example.com
@author        me
@version       0.1.0
@preprocessor  stylus
@var checkbox  fontEnable "Font enabled" 1
@var text      fontSize   "Font size"     2.1em
@var color     fontColor  "Font color"     #123456
@var select    fontName   "Font name"      ["Arial", "Consolas*", "Times New Roman"]
@var select    fontBkgd   "Body background color" {
  "Near Black": "#111111",
  "Near White*": "#eeeeee"
}
@var text      bkgdImg    "Bkgd image"     "'http://example.com/bkgd.jpg'"
@var text      logoImg    "Logo image"     none
@var number    adOpacity  "Ad opacity"      [0.5, 0, 1, 0.1]
@var range     imgHeight  "Max image height" [50, 10, 200, 10, "px"]
==/UserStyle== */
@-moz-document domain("example.com") {
  if fontEnable {
    body {
      font-size: fontSize !important;
      color: fontColor !important;
      font-family: fontName !important; /* Consolas (without the "*") will be set as default */
      background-color: fontBkgd !important;
    }
  }
}
```



```
/* This approach will not allow setting the background-image to "none" */
background-image: url(bkgdImg) !important;
}
#logo {
  /* This approach will, but needs url('URL_HERE') written in the config input if set to a URL */
  background-image: logoImg !important;
}
#ad {
  opacity: adOpacity !important;
}
img {
  max-height: imgHeight !important; /* the 'px' is added to this value automatically */
}
}
```

xStyle compatibility

The [xStyle extension](#) also provides UserCSS metadata which differs slightly from our specification.

If writing a new style, make sure to set the @preprocessor appropriately. At the time of this writing, the Stylus language preprocessor is not supported by xStyle.

Our implementation supports use of:

For more details, see the [xStyle options](#) documentation.

xStyle Example:

```
/* ==UserStyle==
@name      test
@namespace tester
@version   0.1.0
@license   unlicense

@advanced text    a_text      "A Text"  "SomeWords"
@advanced color   a_color     "A Color"  #ffffff;
@advanced dropdown a_background "Your browser" {
  fx "Firefox" <<<EOT
background-color: red; EOT;
  cr "Chrome" <<<EOT
background-color: green; EOT;
}
@advanced image   a_background_image "Page background" {
  bg_1 "Background 1" "http://example.com/example.jpg"
  bg_2 "Background 2" "http://example.com/photo.php?id=_A_IMAGE_ID_"
}
==/UserStyle== */

@-moz-document domain("my-site.com") {
  div.message::after {
    content: /*[[a_text]]*/;
  }
  div.message {
    color: /*[[a_color]]*/
    /*[[a_background]]*/
    background-image: url(/*[[a_background_image]]*/);
  }
}
```

More examples

The settings on userstyles.org allow users to choose a default image or set a custom image. To realize this in UserCSS, a tricky combination of select and text is used:

- Using @var

```
/* ==UserStyle==
@name      Background image
@description How to add a default & custom background image
@namespace examples
```

```
@version      0.1.0
@preprocessor default
@var select bg-selected "Background image" {
  "Default image 1": "url(https://mysite.com/default-background-1.jpg)",
  "Default image 2": "url(https://mysite.com/default-background-2.jpg)",
  "Custom": "var(--bg-custom, none)",
  "None": "none"
}
@var text bg-custom "Custom Background; include url()" "url()"
==/UserStyle= */

@-moz-document domain("mysite.com") {
  body {
    background-color: #222 !important;
    background-image: var(--bg-selected) !important;
  }
}
```

- The same using @advanced:

```
/* ==UserStyle==
@name      Background image
@description How to add a default & custom background image
@namespace  examples
@version    0.1.0
@advanced dropdown bg-selected "Background image" {
  image1 "Default image 1" <<<EOT
  url(https://mysite.com/default-background-1.jpg) EOT;
  image2 "Default image 2" <<<EOT
  url(https://mysite.com/default-background-2.jpg) EOT;
  custom "Custom background" <<<EOT
  var(--bg-custom, none) EOT;
  noBG "None" <<<EOT
  none EOT;
}
@advanced text bg-custom "Custom Background; include url()" "url()"
==/UserStyle= */

@-moz-document domain("mysite.com") {
  :root {
    --bg-custom: /*[[bg-custom]]*/;
  }
  body {
    background-color: #222 !important;
    background-image: /*[[bg-selected]]*/ !important;
  }
}
```