

Stylistic sets allow you to access additional, optional parts of the font on a per-feature basis. In Fira Code, they allow you to enable/disable alternative character variants.

Fira Code v6 supports these stylistic sets:

Code	Default	Variant	Code	Default	Variant
cv01	a	ɑ	cv17	~	˜
cv02	g	ɡ	ss05	@	Ⓐ
cv03..06	i	ï i i i	ss04	\$	⸈
cv07..10	l	ł Ⱥ Ȼ ȼ	cv18	%	⌘
ss01	r	ʀ	ss03	&	⌘
zero, cv11..13	0	⦿ 0 0 0	cv15..16	*	⌘ *
cv14	3	Ⓕ	cv31	()	Ⓐ
onum	4679	4679	cv29	{}	⌘
			cv30		⌘

Code	No ligatures	Default	Variant
ss02	<= >=	⩷ ⩸	≤ ≥
cv19..20	<=	⩷	≤ ⇐
cv23	>=	⩸	≥
cv21..22	=<	=<	⩷ ≤
ss08	== === != !==	= ≡ ≠ ≐	== === ≠ ≐
cv24	/=	⩵	≠
ss09	>>= <<=   =  =	>>= <<=   =  =	⩵ ⇐   =  =
cv25	.-	.-	⋅-
cv26	:-	:-	⋮-
cv32	.=	.=	⋅=
cv27	[]	[]	□
cv28	{. .}	{. .}	{⋅ ⋅}
ss06	\\	\\	\\
ss07	=~ !~	=~ !~	≈ ≠
ss10	Fl Tl fi fj fl ft	Fl Tl fi fj fl ft	ℱ ℒ fi fj fl ft

You can enable stylistic sets **only if your editor supports it**.

## Atom

Go to `Atom` -> `Stylesheet` , add:

```
atom-text-editor {  
  font-feature-settings: "ss01", "ss02", "ss03", "ss04", "ss05", "ss06", "zero",  
  "onum";  
}
```

## iTerm2

Go to `iTerm2` -> `Preferences` -> `Advanced` and scroll to the `Drawing` section, then change:

Improves drawing performance at the expense of disallowing alphanumeric characters to belong to ligatures. to `No`

## Sublime Text

Go to `Sublime Text` -> `Preferences` -> `Settings` , add:

```
"font_options": ["ss01", "ss02", "ss03", "ss04", "ss05", "ss06", "ss07"]
```

For now Sublime Text [does not allow you to specify zero and onum features](#).

## VS Code

Since version [1.40](#), you can define in `settings.json` :

```
"editor.fontLigatures": "'ss01', 'ss02', 'ss03', 'ss04', 'ss05', 'ss06', 'zero',  
'onum'"
```

To avoid possible bugs, you must uninstall Fira Code 1.x before installing Fira Code 2+.

## CSS

```
.monaco-editor {  
  font-feature-settings: "ss01", "ss02", "ss03", "ss04", "ss05", "ss06", "zero",  
  "onum";  
}
```

## RStudio

Create or edit an [rstheme file](#) and add:

```
.ace_editor {  
  font-feature-settings: "ss01", "ss02", "ss03", "ss04", "ss05", "ss06", "zero",  
  "onum";  
}
```

## Kitty

Select which font variant you wish to use (Regular, Medium, Light, ...) and set that as your font on the kitty's config file ( `~/.config/kitty/kitty.conf` ), for instance:

```
font_family      Fira Code Regular
```

Then fetch the PS Name (the value between parenthesis) of the chosen font variant using kitty:

```
$ kitty + list-fonts --psnames | grep "Fira Code Regular"  
Fira Code Regular (FiraCode-Regular)
```

Finally use the PS Name setting the following key in the `kitty.conf` file:

```
font_features FiraCode-Regular +ss01 +ss02 +ss03 +ss04 +ss05 +ss07 +ss08 +zero +onum
```

## Wez's terminal

```
return {  
  harfbuzz_features = {"zero" , "ss01", "cv05"}  
}
```

Full instruction [here](#).

## Windows Terminal

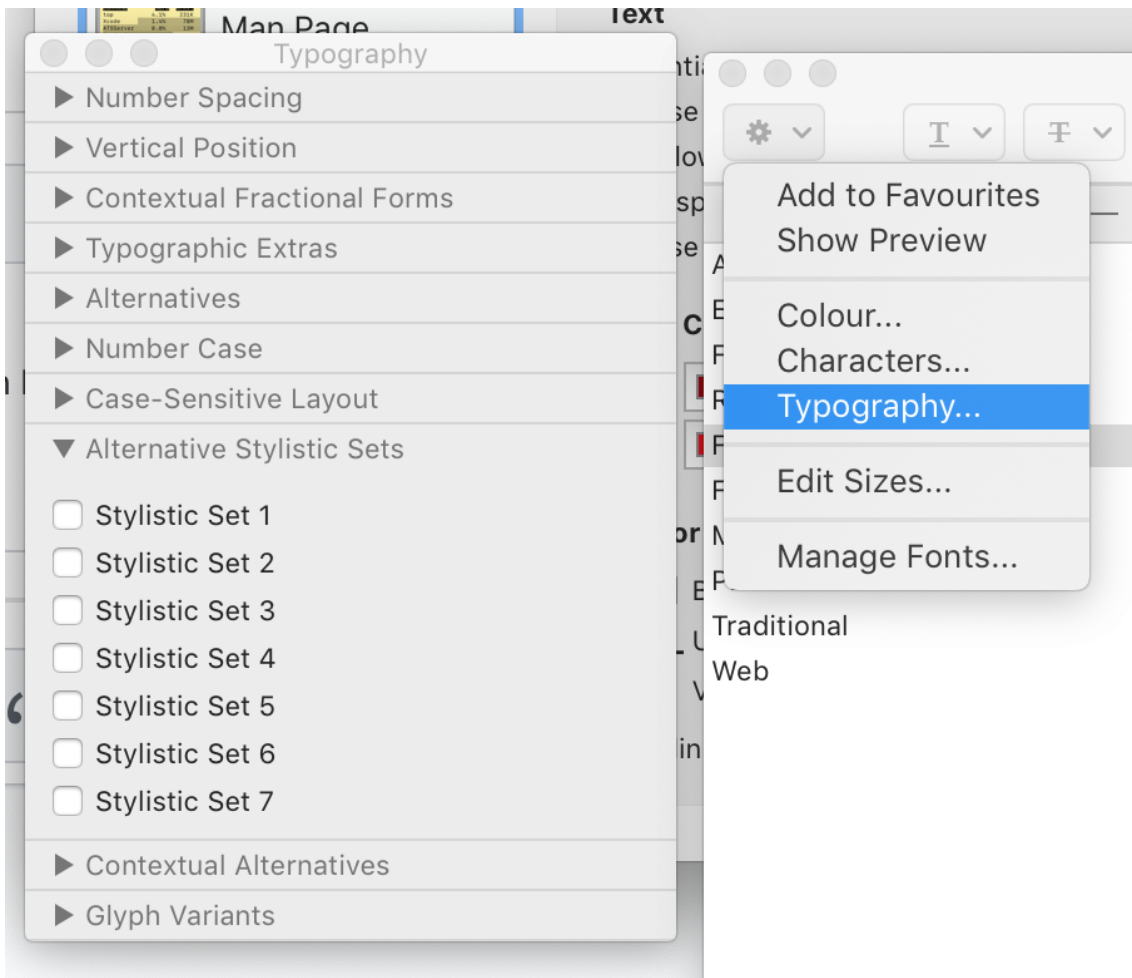
Open the Windows Terminal settings.json file. You will need to insert a [font object](#) for Fira Code in one or more profiles:

An example font object:

```
"font": {  
  "face": "Fira Code",  
  "features": {  
    "ss01": 1,  
    "ss02": 1,
```

```
"ss03": 1,  
"ss04": 1,  
"ss05": 1,  
"ss06": 1,  
"zero": 1,  
"onum": 1  
}  
}
```

## Some Apple apps



## Adobe apps, MS Word etc

<https://www.macworld.com/article/3052388/how-to-access-advanced-opentype-features-in-a-variety-of-mac-apps.html>

## Baking in stylistic sets into the font file

This might be an option if your editor does not let you choose stylistic sets on the fly

<https://github.com/twardoch/fonttools-opentype-feature-freezer>