

# Tuning my Unix Terminal



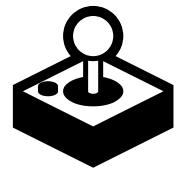
By Gustavo Adolfo Rodríguez

July 2020



Thanks for being here.  
This was made with ❤

# General terms..

My story boosting my efficiency  in the CLI while having fun  looking cooler 

Or...

- A brief intro...
- All that I'd have wanted to know before my first How-to... in Google

In Criollo terms..

Una descarga bien mela de  
pura chicanería caleña, sin  
mayores pretensiones, pa'  
entretenernos un rato en  
este encierro tan berraco



# Me: Gustavo Adolfo Rodríguez

- PSL:
  - April 2017
- Project:
  - MuleSoft Runtime Delivery  
(Java ecosystem)
- Role:
  -  60% Terminal, 15% Jenkins, 25% IDE...  
[DevOps]



# Agenda

- Intro 
- The root concepts 
- Styling 
- Shell features 
- Plugins 
- My favorites 
- Inspiration 
- Environment setup 
- Demo 
- Q&A 



# Intro

# Disclaimers

- I'm not an expert on:
  - This topic
  - On presentations. I get nervous  
- Political correctness
  - I'll try to do my best 

# Disclaimers

- I'm not trying to convince you of any tool, rather convince you to dive in this amazing world 
- Probably you'll have to Google or ask me a lot in case you want to get your hands dirty 

# Feedback 🎧

- Please fill the feedback survey and let us know if we should do a second part 📋
- Contact me directly with comments or doubts... You can't imagine how much that means to me
- Questions at the end 🙏



[https://s27389.pcdn.co/wp-content/uploads/AdobeStock\\_50345277.jpeg](https://s27389.pcdn.co/wp-content/uploads/AdobeStock_50345277.jpeg)



<https://www.supercars.net/blog/wp-content/uploads/2018/01/Bugatti-Chiron.jpg>

# What?

- The needed foundations 
  - There are thousands of tutorials online about this...  
But, in my journey, I noticed that few cover the base knowledge required 
- The cool and fun stuff 
  - My recommendations

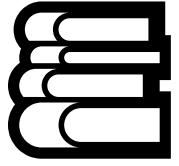
# Why? The real value!

- Efficiency
  - Fewer keystrokes to get a task done
- Productivity
  - Getting more task done
- Experience
  - Feeling better (What you feel when you tune your toys)

# How?

- Slides 😔
  - But this is going to be fast! 💨 (100+) 😊
- Demo 😃
  - This is going to be fun 🎉
- Starter kit 🎁
  - This is how you're going ❤️ this talk (and me 🎉 😄)

# Concepts



# Terminal, console, shell, CLI? 😕

160

Short answer:

- **terminal** = text input/output environment
- **console** = physical terminal
- **shell** = command line interpreter



Long answer:



Console and terminal are closely related. Originally, they meant a piece of equipment through which you could interact with a computer: in the early days of unix, that meant a teleprinter-style device resembling a typewriter, sometimes called a teletypewriter, or "tty" in shorthand. The name "terminal" came from the electronic point of view, and the name "console" from the furniture point of view. Very early in unix history, electronic keyboards and displays became the norm for terminals.

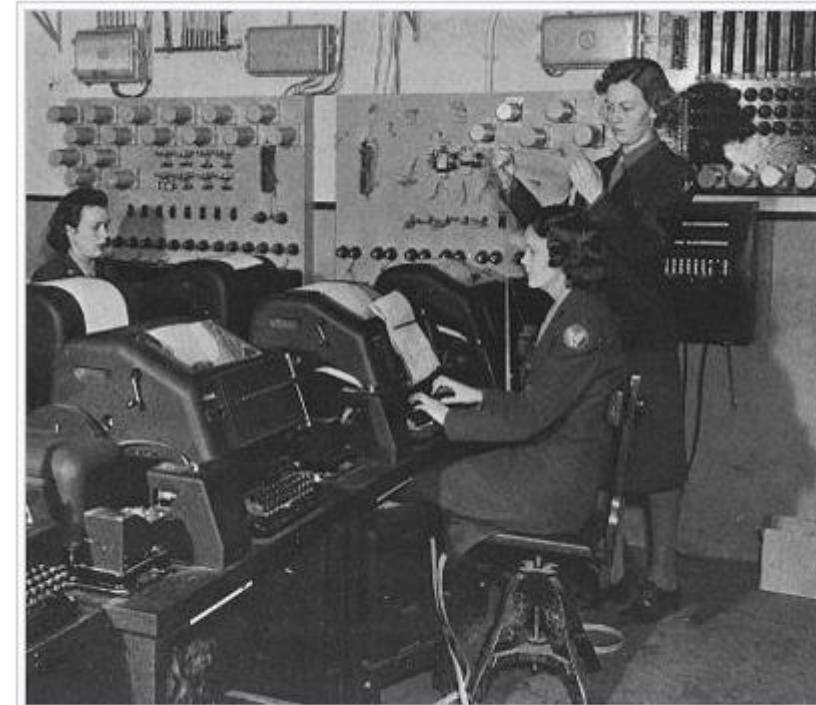
Ref: <https://askubuntu.com/a/506628/98631>

# Terminal

Something you can sit down at, and work like a boss.



Ref: <https://askubuntu.com/a/506880/98631>



Ref: <https://en.wikipedia.org/wiki/Teleprinter>

# Terminal

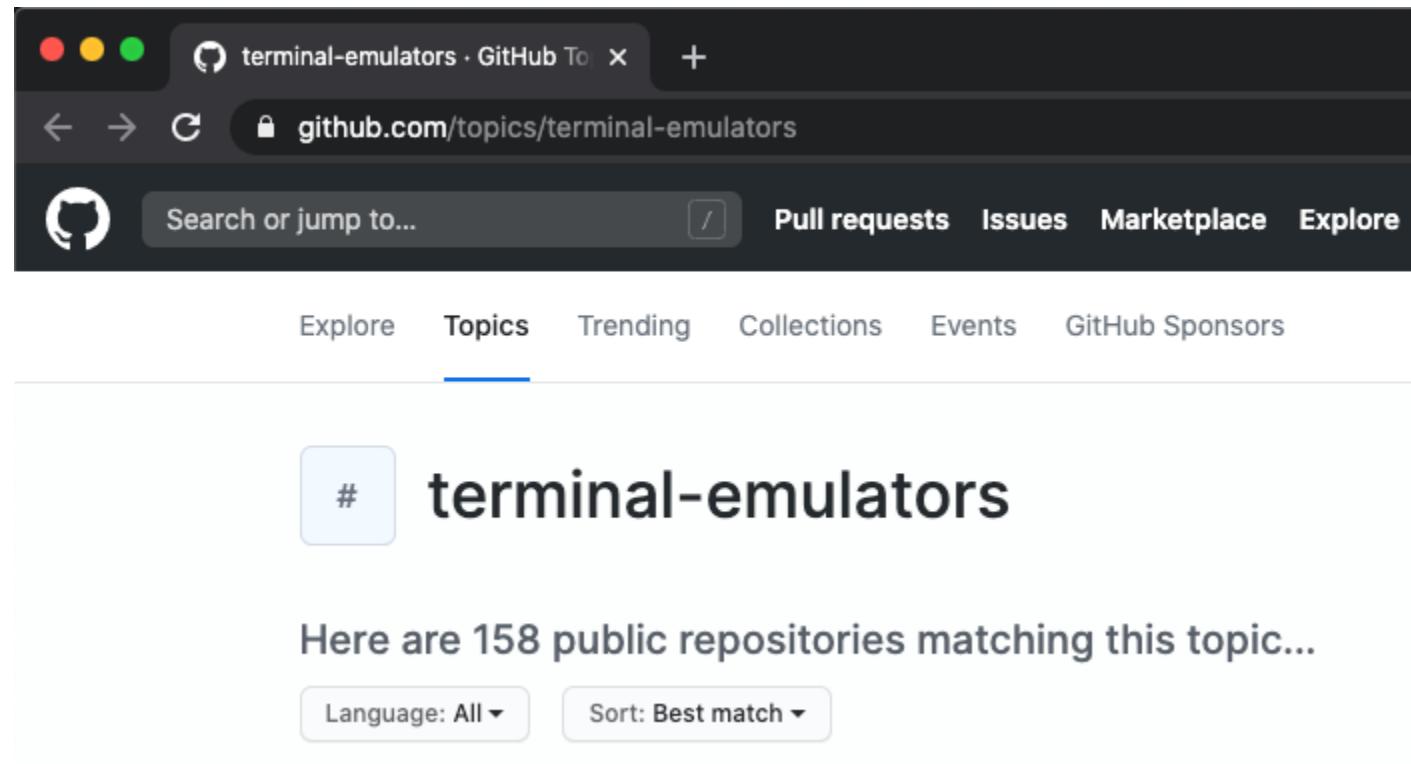


[https://youtu.be/AwqryPuwl\\_w](https://youtu.be/AwqryPuwl_w)

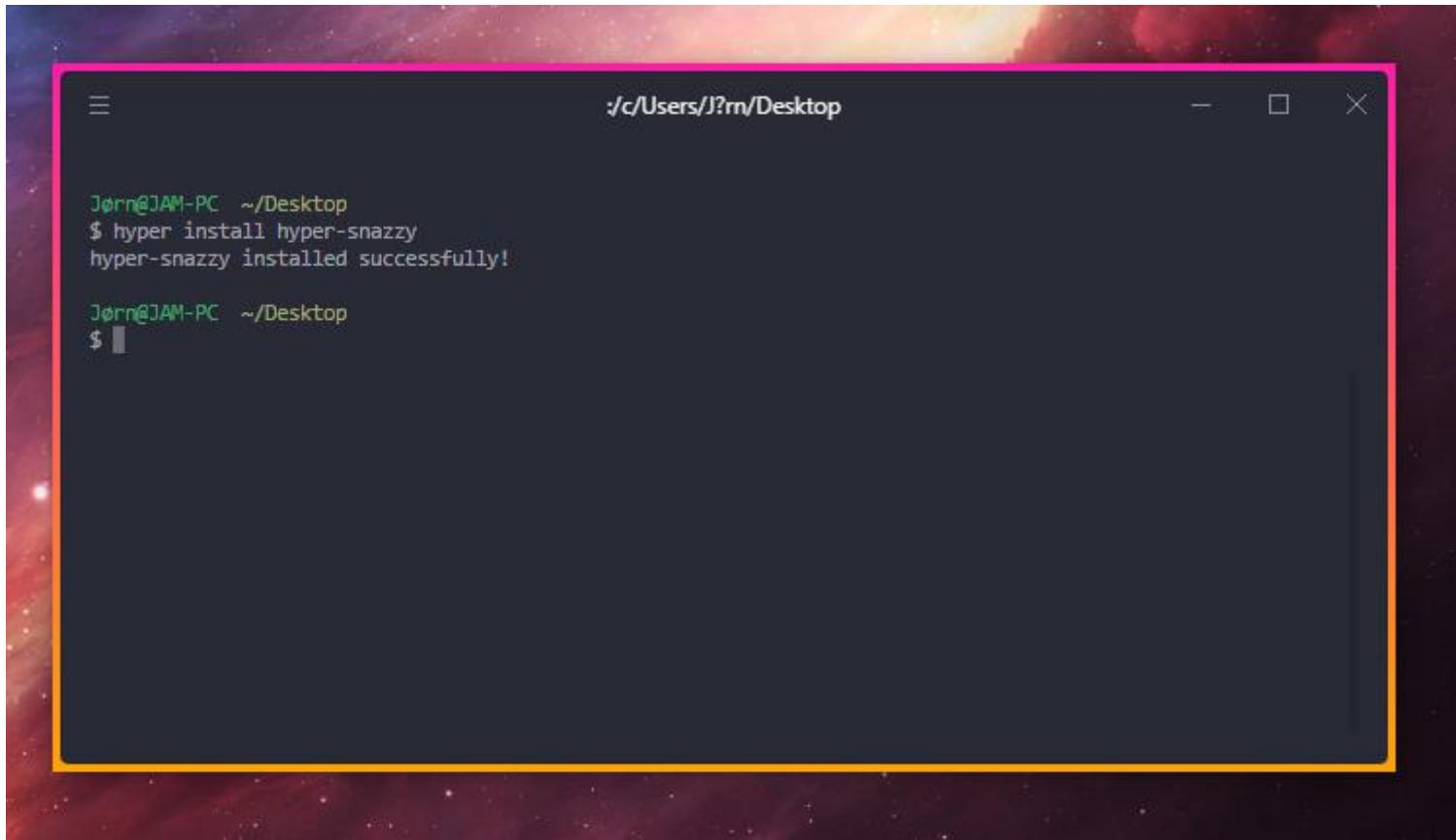


<https://youtu.be/TRxz4x45i54>

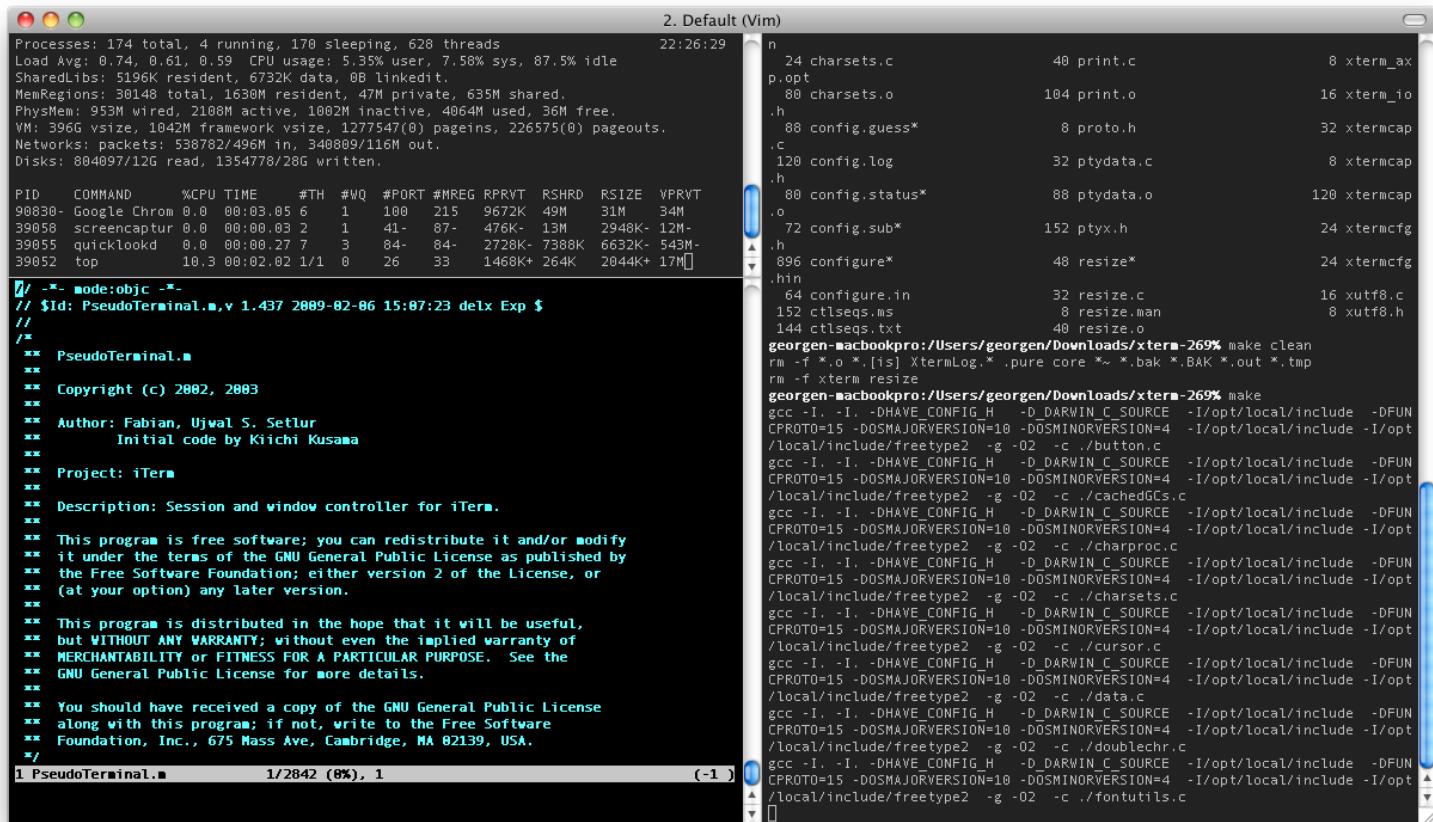
# Terminal emulators



# vercel/hyper



# gnachman/iTerm2



# microsoft/terminal



# Shell



# Shell

- Unix's precursor [Multics](#).
- Command processor



The [Multics glossary](#) may provide a clue. The shell is defined thus:

The Multics [command processor](#) used to be called the shell. This program is passed a command line for execution by the listener; it parses the line into a command name and arguments, locates the command and initiates it, and calls the command program with arguments that are PL/I character strings. It is simple to replace the default system supplied shell with a user-provided program, by calling cu\_set\_cp (see abbrev). A Unix shell includes the concepts of both shell and listener in the Multics sense.

The relationship between *shell* and *listener* makes sense both etymologically and [metaphorically](#) [1].

[1] "Since the shell is a symbol of authority, speech, and hearing, which is to say a symbol of prophecy..."

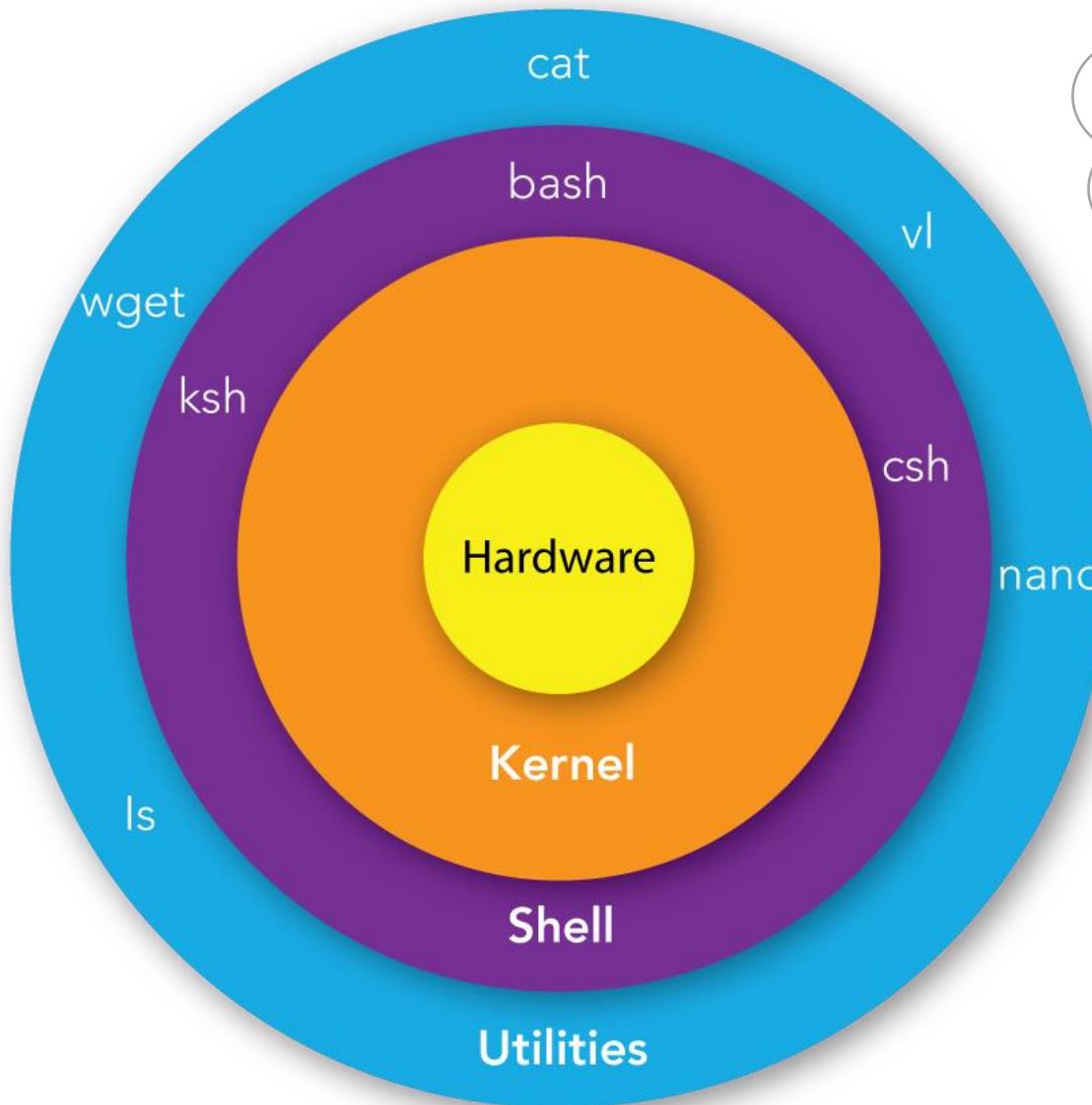
[share](#) [edit](#) [follow](#)

answered Jun 13 '11 at 19:30



<https://unix.stackexchange.com/a/14939/217673>

# Shell



Origins of  
encapsulation?

# Shells

## Unix

- Sh: Bourne Shell
- Bash: Bourne Again Shell
- Csh: C Shell
- Ksh: Kron Shell
- Zsh
- Tcsh
- Fish Shell
- Ion Shell

## Windows

- CMD
- PowerShell

# Shells for this talk:



## Z shell



 GitHub Archive Program | The Gi X +

← → C 🔒 archiveprogram.github.com ⭐ 🌐

# Archive Program

## FEATURED PROJECTS



# Explore some of the top projects archived in the 2020 Arctic Vault program

---

[dotnet/core](#)

[rails/rails](#)

[Homebrew/brew](#)

[apache/hadoop](#)

[fastai/fastai](#)

[rust-lang/rust](#)

[ruby/ruby](#)

[zeromq/libzmq](#)

[robbyrussell/oh-my-zsh](#)

[github/gh-ost](#)

---

[torvalds/linux](#)

[docker/machine](#)

[php/php-src](#)

[v8/v8](#)

[jimweirich/builder](#)

[golang/go](#)

[facebook/react](#)

[postgres/postgres](#)

[xamarin/xunit](#)

[rspec/rspec](#)

---

[python/cpython](#)

[openssl/openssl](#)

[twbs/bootstrap](#)

[Alamofire/Alamofire](#)

[vercel/next.js](#)

[angular/angular](#)

[CocoaPods/CocoaPods](#)

[microsoft/MS-DOS](#)

[grafana/grafana](#)

[libgit2/libgit2](#)

---

[bitcoin/bitcoin](#)

[nodejs/node](#)

[microsoft/TypeScript](#)

[gatsbyjs/gatsby](#)

[WordPress/WordPress](#)

[jquery/jquery](#)

[jupyter/notebook](#)

[Netflix/chaosmonkey](#)

[graphql/graphql-js](#)

Many more

# GitHub Arctic Code Vault



**Gustavo Adolfo  
Rodríguez Libreros**  
garodriguezlp

Passionate software developer made in @UnivalleCol and Salsa music lover

[Edit profile](#)

13 followers · 13 following · 50

 [@psicorp](#)  
 Cali, Colombia

---

#### Highlights

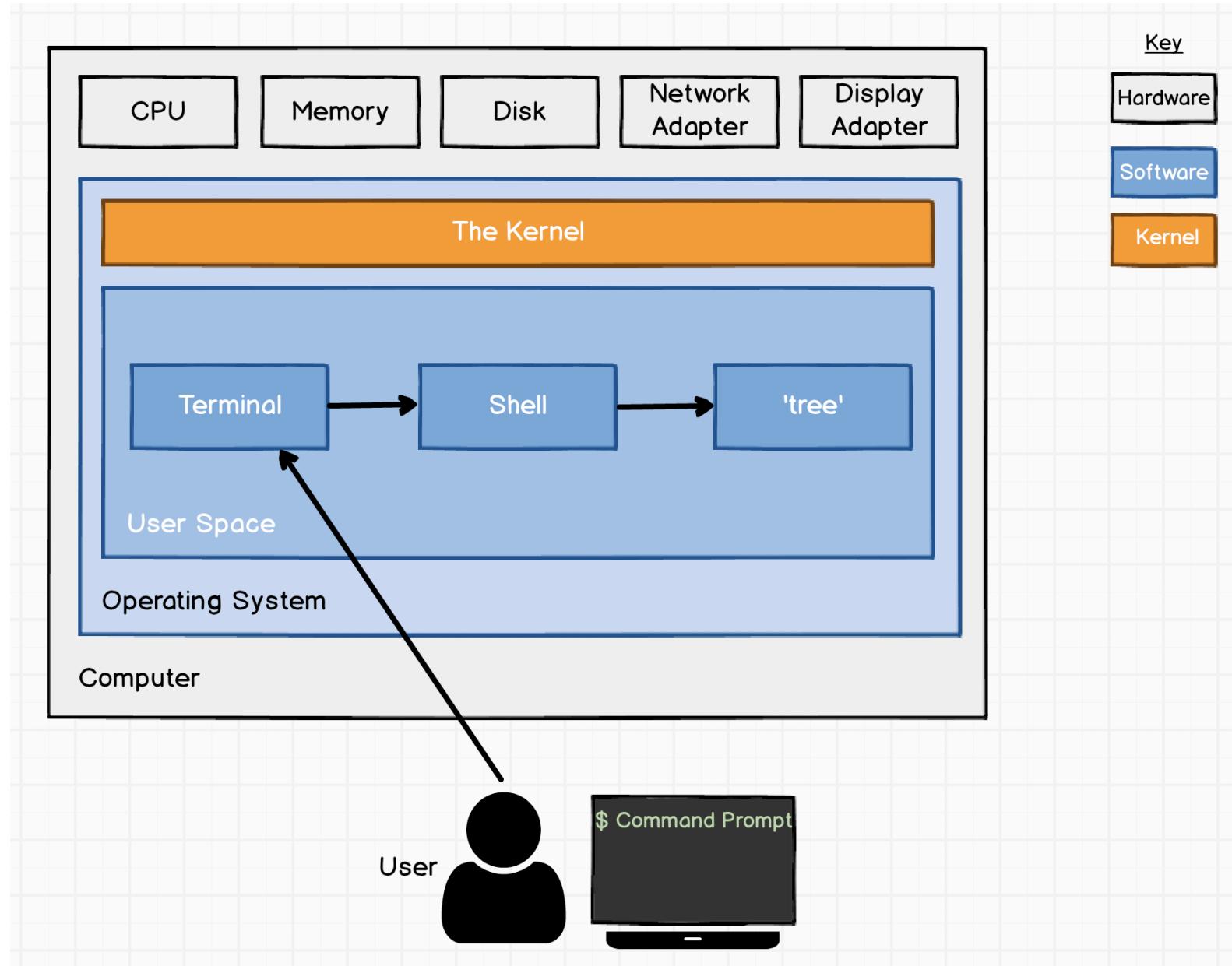
\* Arctic Code Vault Contributor

---

#### Organizations



# Summary



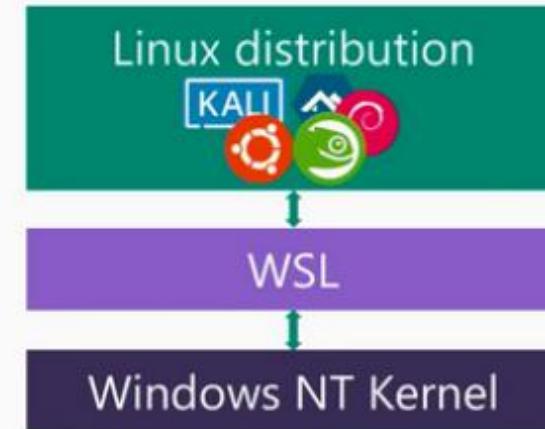
<https://dwmkerr.com/effective-shell-part-5-understanding-the-shell/>

# I use Windows 😭

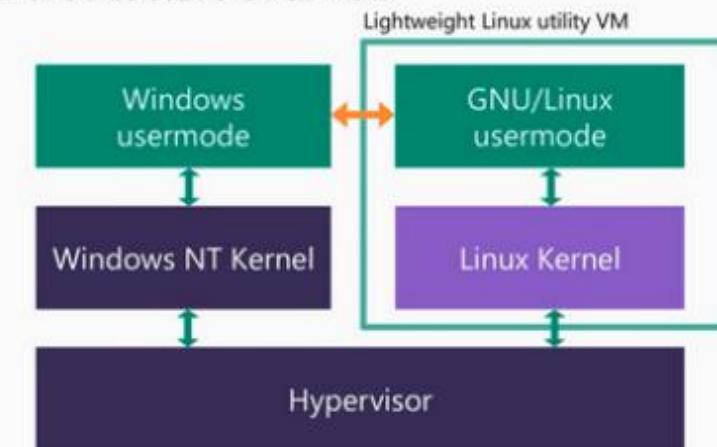


<https://docs.microsoft.com/en-us/windows/wsl/>

## WSL architecture

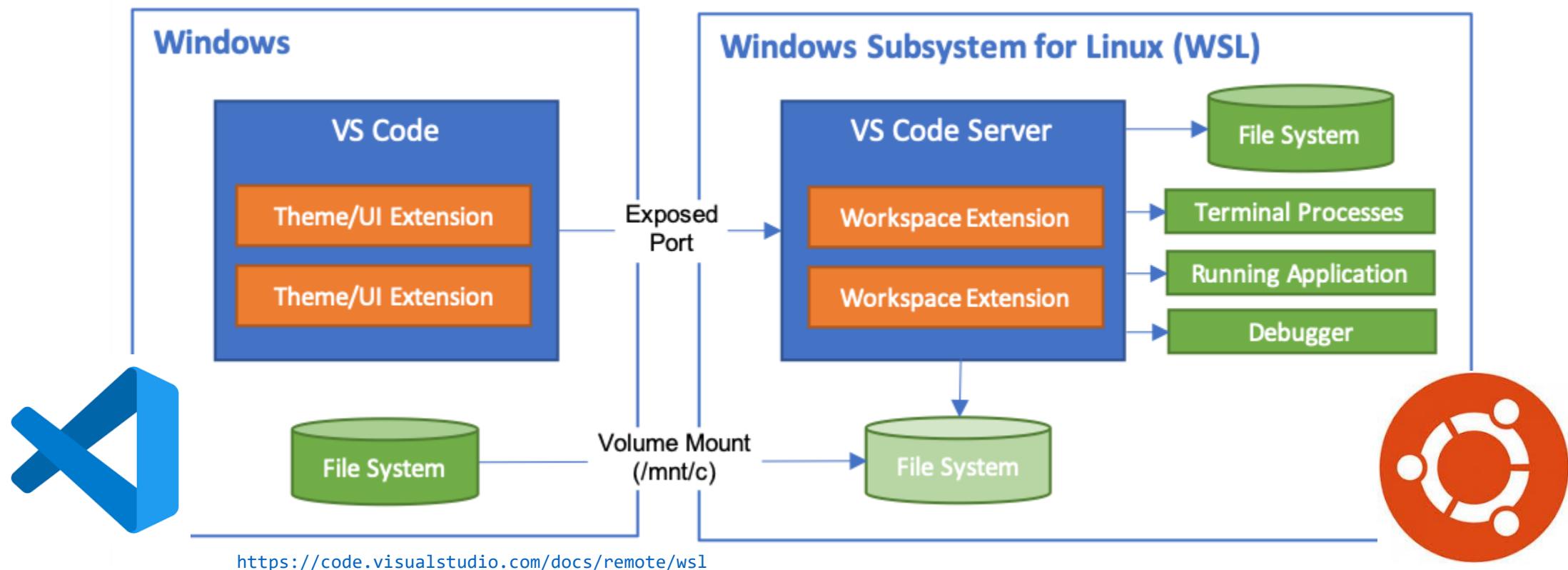


## WSL 2 architecture overview



[https://4.bp.blogspot.com/-e\\_FaEsP7nH8/Xca4QMFTjQI/AAAAAAAERA/r59YN5UCX1wdNsfpNy-sjjyTQ4o\\_dtbjACLcBGAsYHQ/s1600/wsl1-wsl2.png](https://4.bp.blogspot.com/-e_FaEsP7nH8/Xca4QMFTjQI/AAAAAAAERA/r59YN5UCX1wdNsfpNy-sjjyTQ4o_dtbjACLcBGAsYHQ/s1600/wsl1-wsl2.png)

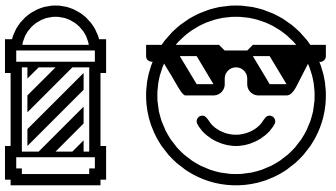
# I use Windows



# So, What are the decision to be made?

- Terminal emulator: Hyper
- Shell: zsh
- Tools
- Style

# Styling



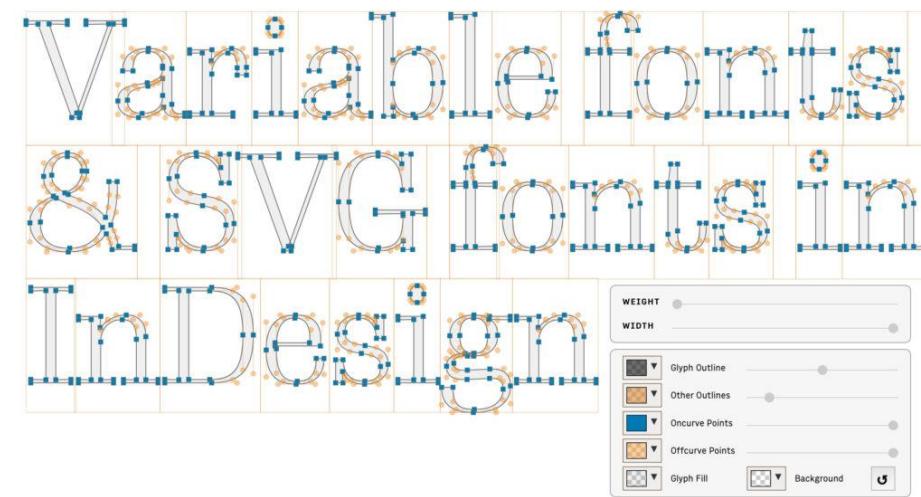
Font and Color

Fonts α-Ω

# Fonts



You can add your own to the OS 😎

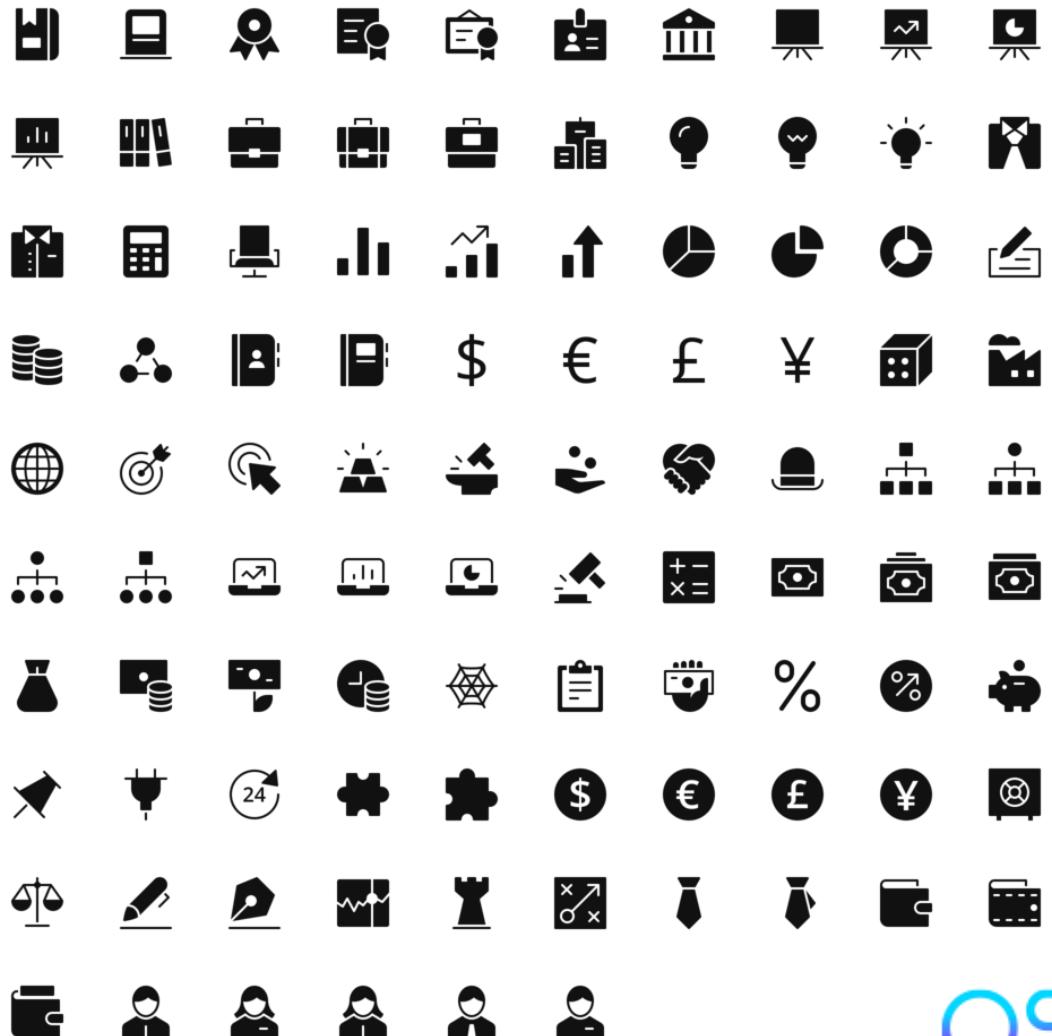


# Glyph (a.k.a Icons)

- In typography, a **glyph** is an elemental symbol within an agreed set of symbols, intended to represent a readable character for the purposes of writing

a a a  
aabaa  
a a a

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





## JOEL ON SOFTWARE

YOUR HOST



I'm Joel Spolsky, a software developer in New York City.

OCTOBER 8, 2003 by JOEL SPOLSKY

# The Absolute Minimum Every Software Developer Absolutely, Positively Must Know About

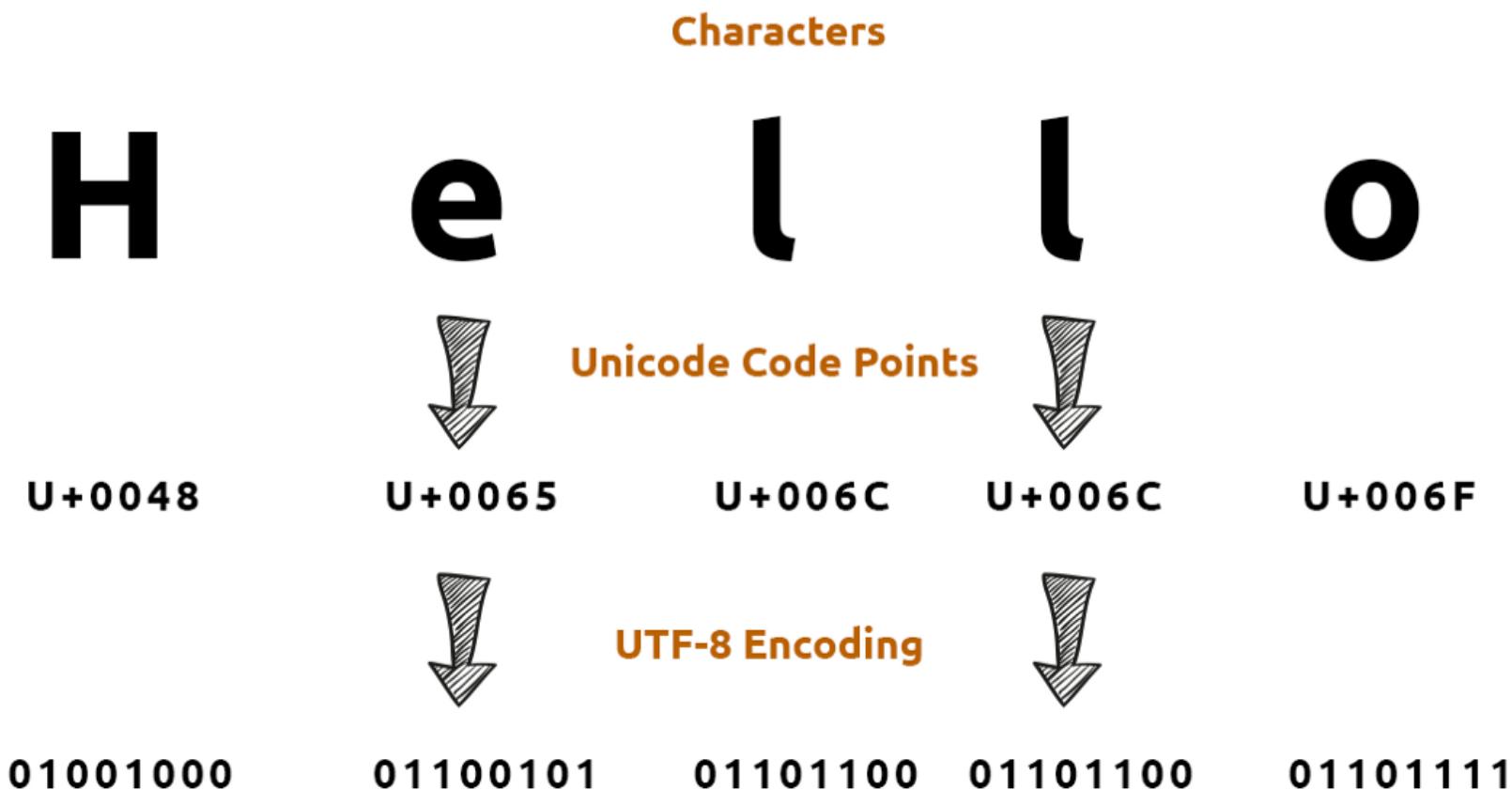
## Unicode and Character Sets (No Excuses!)

TOP 10, NEW DEVELOPER, NEWS

> “if you are a programmer working in 2003 and you don’t know the basics of characters, character sets, encodings, and Unicode, and I catch you, I’m going to punish you by making you peel onions for 6 months in a submarine. I swear I will.” Joel Spolsky

- MS Excel
- Stack Overflow
- Trello
- Glitch

# Character Encoding



[https://miro.medium.com/max/1200/0\\*02gVrncQJf55othp](https://miro.medium.com/max/1200/0*02gVrncQJf55othp)

ASCII: Good...  
but, just for  
English 😊

USASCII code chart															
Regular ASCII Chart character codes 0 - 127															
Extended ASCII chart character codes 128 - 255															
Character Codes															
Character Table															
More images															

## ASCII

ASCII, abbreviated from [American Standard Code for Information Interchange](#), is a character encoding standard for electronic communication. ASCII codes represent text in computers, telecommunications equipment, and other devices. [Wikipedia](#)

**Classification:** ISO 646 series

**Extensions:** Unicode; ISO/IEC 8859 (series); KOI-8; OEM (series); Windows-125x (series); Others

**Alias(es):** ISO-IR-006, ANSI\_X3.4-1968, ANSI\_X3.4-1986, ISO\_646.irv:1991, ISO646-US, us, IBM367, cp367

[Feedback](#)

# Unicode 1

ε	≈	₩	ଓ	❑	☺	❑	○	·	ػ
U+03B5	U+2248	U+1F911	U+0CA0	U+2751	U+1F617	U+2B1B	U+3007	U+2027	U+055E
ڏ	ڦ	ڦ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ	ڻ
U+30C4	U+2690	U+FF61	U+FEED	U+263A	U+1F47D	U+1F9E1	U+27AB	U+1D55	U+1F499
)	݂	Everyone in the world should be able to use their own language on phones and computers.							
U+207E	U+0E9F								
ܖ	ܭ	<a href="#">LEARN MORE ABOUT UNICODE</a>							
܄	܄	܄	炎	鄉	܄	܄	܄	܄	܄
U+12E4	U+30EE	U+2193	U+27E8	U+708E	U+9109	U+0C95	U+1F609	U+063A	U+04B9
܄	܄	܄	܄	܄	܄	܄	܄	܄	܄
܄	܄	܄	܄	܄	܄	܄	܄	܄	܄
U+4E14	U+0642	U+0C39	U+53E3	U+0B14	U+30FB	U+51F8	U+02C7	U+4E09	U+01D0



[ADOPT A CHARACTER](#)

The screenshot shows the main page of the Unicode website. It features a large blue "Uni" logo followed by "UNICODE". Below the logo is a grid of various emoji characters. To the right of the grid is a "More images" button. The page also includes a "Share" icon and a "Wikipedia" link.

## Unicode

Unicode is an information technology standard for the consistent encoding, representation, and handling of text expressed in most of the world's writing systems.

[Wikipedia](#)

**Encoding formats:** [UTF-8](#), [UTF-16](#), [GB18030](#); Less common: [UTF-32](#), [BOCU](#), [SCSU](#), [UTF-7](#)

**Standard:** Unicode Standard

**Language(s):** International

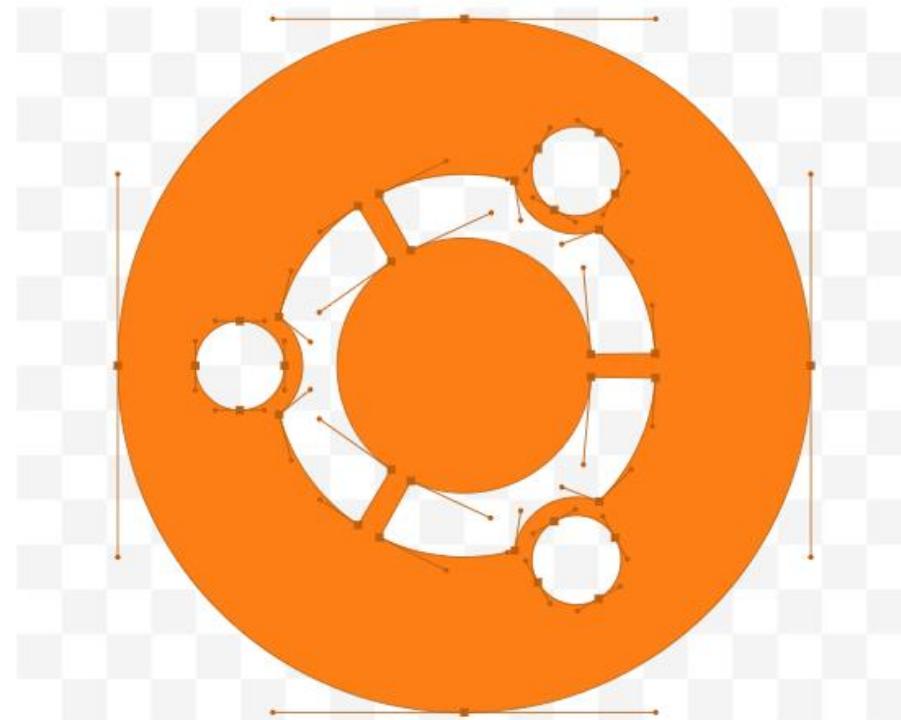
**Alias(es):** Universal Coded Character Set (UCS)

[Feedback](#)

Try: echo '\uf31b'

ubuntu

Brands Style (fab) • f7df • <i class="fab fa-ubuntu"></i> •



Thanks to the community for voting this icon into Font Awesome!

<https://fontawesome.com/icons/ubuntu?style=brands>

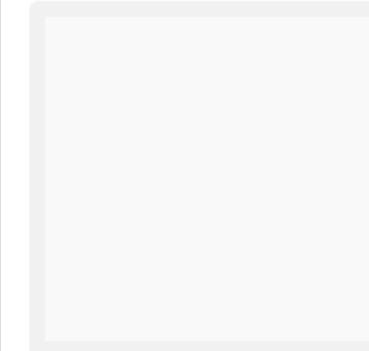
:( Why do I need  
to know about  
encoding?

Charbase

A visual unicode database

← U+F7DE INVALID CHARACTER

## U+F7DF: INVALID CHARACTER



Tweet

Your Browser	
Index	U+F7DF (63455)
Class	Unassigned (Cn)
Block	Private Use Area
Java Escape	"\uf7df"
Javascript Escape	"\uf7df"
Python Escape	u"\uf7df"
HTML Escapes	&#63455; &#xf7df;
URL Encoded	q=%EF%9F%9F
UTF8	ef 9f 9f
UTF16	f7df

<https://charbase.com/f7df-unicode-invalid-character>

psl  
Now part of Perficient

Got it! Stop talking and tell me where to get cool fonts 😠

Sorry, 😅 There is one more thing to know before we get there... 💡

#

# ligatures

Here are 51 public repositories matching this topic...

Language: All ▾ Sort: Best match ▾

A grid of various ligature characters from the FiraCode font, including symbols like equals, less than or equal to, and greater than or equal to.

tonsky / FiraCode

Star 49.1k

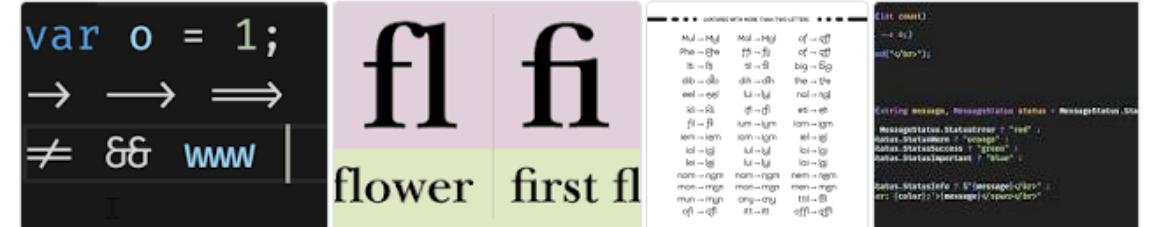
Code Issues Pull requests

Free monospaced font with programming ligatures

font ligatures programming-ligatures

Updated 5 days ago Clojure

# Font ligatures



Typographic **ligatures** are when multiple characters appear to combine into a single character. Simplistically, when you type two or more characters and they magically attach to each other, you're using **ligatures** that were supported by your OS, your app, and your **font**. Jul 20, 2017

[www.hanselman.com](http://www.hanselman.com) › blog › MonospacedProgrammi... ▾

Monospaced Programming Fonts with Ligatures - Scott ...

About Featured Snippets Feedback

# Font ligatures

<b>Fira Code v5</b>	<b>Fira Mono</b>
ligatures: <input checked="" type="button"/> ON	ligatures: <input type="button"/> NO
<b>Common</b>	
Arithmetics <code>++ -- ≈ &amp;&amp;      =</code>	<code>++ -- /= &amp;&amp;      =</code>
Scope <code>→ ⇒ :: _</code>	<code>-&gt; =&gt; :: __</code>
Equality <code>= ≡ ≠ ≡ = ≡ ≡ ≡ ≠ ≡ ≠ =</code>	<code>= ≡ ≡ !≡ =/= ≡ ≡ ≡ ≡ !≡ ≡</code>
Comparisons <code>≤ ≥ ≤ ≥ ⇌</code>	<code>&lt;= &gt;= &lt;= &gt;= &lt;=&gt;</code>
Comments <code>/* */ // ///</code>	<code>/* */ // ///</code>
Escaped chars <code>\n \\</code>	<code>\n \\</code>
Bit operations <code>&lt;&lt; &lt;&lt;&lt; &lt;= &gt;&gt; &gt;&gt;&gt; ≫= ≫ ≈</code>	<code>&lt;&lt; &lt;&lt;&lt; &lt;= &gt;&gt; &gt;&gt;&gt; &gt;=  = ^=</code>
Hexadecimal Ex <code>0xFF 1920x1080</code>	<code>0xFF 1920x1080</code>

<https://github.com/tonsky/FiraCode/blob/master/extras/ligatures.png>

Try it:

```
> curl -s https://raw.githubusercontent.com/garodriguezlp/zsh-bootstrap/master/demo/ligatures.sh | bash
```

# Ok, here's a good source for discovering programming fonts



The screenshot shows a web browser window titled "Programming Fonts - Test Drive". The address bar says "programmingfonts.org". On the left, there's a sidebar with font selection buttons (S, V, S, B) and a search bar. Below that is a list of fonts:

- 3270 (Ricardo Bánffy (2015) — sans)
- Agave (agaric (2013) — sans)
- Anka/Coder (Andrey Makarov (2010) — sans)
- Anonymous Pro (Mark Simonson (2009) — serif)
- APL2741 (Adrian Smith (2003) — serif)
- APL385 (Adrian Smith (2003) — sans)
- Aurulent Sans Mono (Stephen G. Hartke (2007) — sans)

At the bottom of the sidebar, there are buttons for "Size" (16), "1.4", "monokai", and "JavaScript". A footer at the bottom of the sidebar says "98 Programming Fonts" and "Test drive all the programming fonts!".

The main area of the browser shows some sample code in a monospaced font:

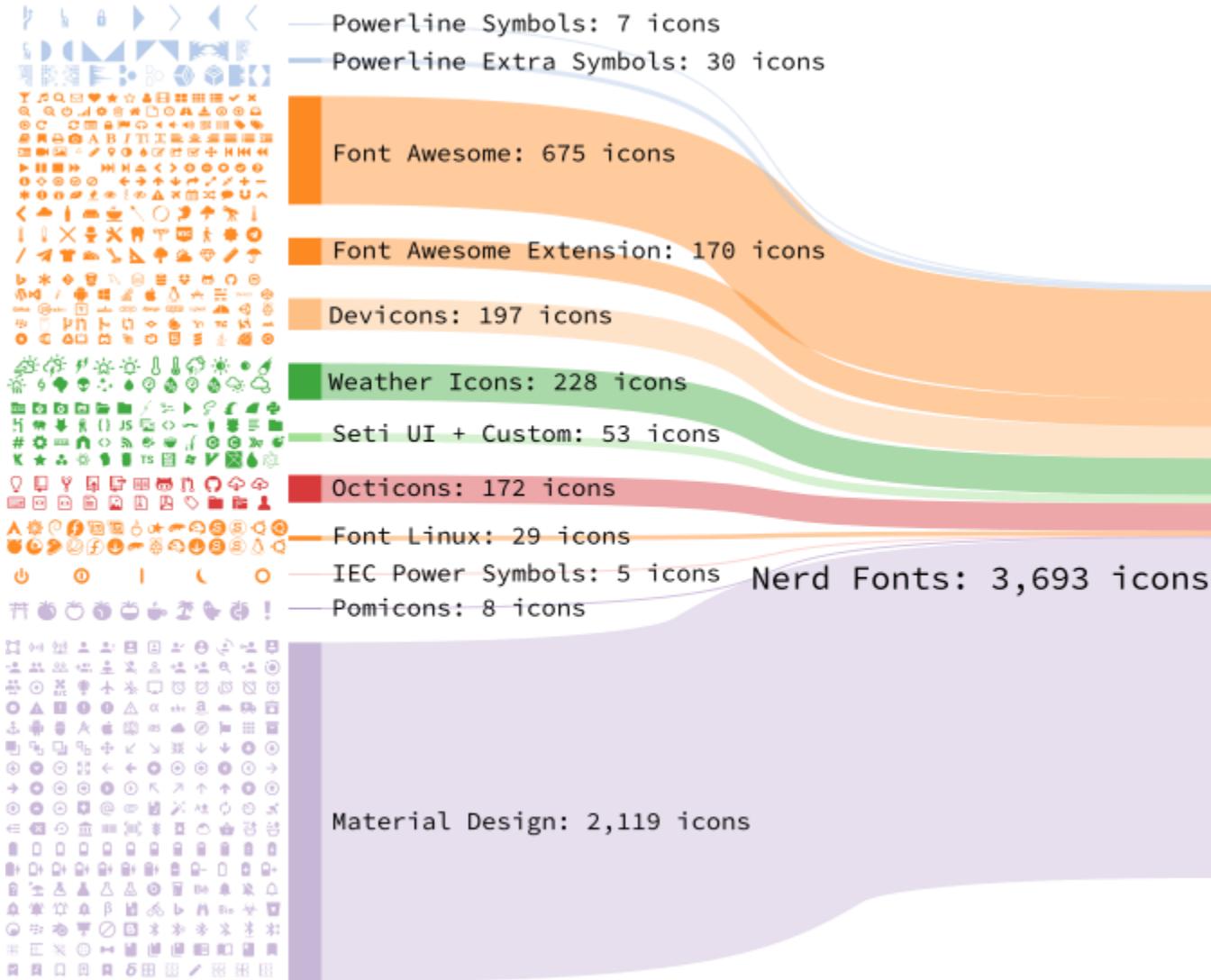
```
1 // Type some code →  
2  
3 console.log "o008 iILL1 g9qCGQ ~--+⇒";  
4  
5 function updateGutters(cm) {  
6     var gutters = cm.display.gutters,  
7         __specs = cm.options.gutters;  
8  
9     removeChildren(gutters);  
10  
11    for (var i = 0; i < specs.length; ++i) {  
12        var gutterClass = __specs[i];  
13        var gEl = gutters.appendChild(  
14            elt(  
15                "div",  
16                null,  
17                "CodeMirror-gutter " + gutterClass  
18            )  
19        );  
20        if (gutterClass == "CodeMirror-linenumbers") {  
21            cm.display.lineGutter = gEl;  
22            gEl.style.width = (cm.display.lineNumWidth || 1) +  
23            "px";  
24        }  
25    }  
26}
```

<https://www.programmingfonts.org/>

But, those won't be enough for us...  
we need them with steroids 💪😎  
(that means... nice Glyphs 😎)



# Nerd Fonts



# So, what's a patched font? 😞

The screenshot shows a web browser displaying the [nerdfonts.com/font-downloads](https://nerdfonts.com/font-downloads) page. The top navigation bar includes links for HOME, FEATURES, FONTS DOWNLOADS, CHEAT SHEET, CONTRIBUTORS, and CHangelog. Below the navigation, there are two sets of font previews. Each set includes a sample of lowercase and uppercase letters, as well as a series of special characters and symbols. Each preview section has a green 'Download' button and a purple 'Preview on ProgrammingFonts.org' button.

Font 1 (Left): abcdefghijklmnopqrstuvwxyz  
ABCDEFGHIJKLMNPQRSTUVWXYZ  
oO08 iI1L1 {} [] g9qCGQ ~-+=>

Info: derived from the x3270 font, a modern format of a font with high nostalgic value

Font 2 (Right): abcdefghijklmnopqrstuvwxyz  
ABCDEFGHIJKLMNPQRSTUVWXYZ  
o008 iI1L1 {} [] g9qCGQ ~-+=>

Info: A small, monospace, outline font that is geometrically regular and simple

Your favorite font + 3K of cool icons 😎

# Please read the FAQ

← → C 🔒 [github.com/ryanoasis/nerd-fonts/wiki/FAQ-and-Troubleshooting](https://github.com/ryanoasis/nerd-fonts/wiki/FAQ-and-Troubleshooting)

What do these acronym variations in the font name mean: **LG** , **L** , **M** , **S** , **DZ** , **SZ** ?

- LG - Line Gap
- L - Large
- M - Medium
- S - Small
- DZ - Dotted Zero
- SZ - Slashed Zero

This particularly applies to Meslo at the moment:

Meslo has changed it's name to Meslo LG which now includes three variants: small, medium and large.

LG stands for Line Gap, so there's one variant for smaller vertical line spacing, more towards Apple's Menlo, a normal line gap (which equals Meslo v0.1) and a large gap, which is more than twice the space of Apple's Menlo.

In addition to Regular, there's Italic, Bold and Bold Italic font styles included for each LG variant.

source: <https://github.com/andreberg/Meslo-Font>

<https://github.com/ryanoasis/nerd-fonts/wiki/FAQ-and-Troubleshooting>

Color 



# 50% of the story



## ANSI Escape sequences (ANSI Escape codes)

---

[ANSI Escape sequences](#) | [VT100 / VT52 ANSI escape sequences](#) | [VT100 User Guide](#)

---

These sequences define functions that change display graphics, control cursor movement, and reassign keys.

ANSI escape sequence is a sequence of ASCII characters, the first two of which are the ASCII "Escape" character 27 (1Bh) and the left-bracket character " [ " (5Bh). The character or characters following the escape and left-bracket characters specify an alphanumeric code that controls a keyboard or display function.

ANSI escape sequences distinguish between uppercase and lowercase letters.

Information is also available on [VT100 / VT52 ANSI escape sequences](#).

<http://ascii-table.com/ansi-escape-sequences.php>

Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0	00	000	NULL	32	20	040	&#032;	Space	64	40	100	&#064;	@	96	60	140	&#096;	`
1	1	001	Start of Header	33	21	041	&#033;	!	65	41	101	&#065;	A	97	61	141	&#097;	a
2	2	002	Start of Text	34	22	042	&#034;	"	66	42	102	&#066;	B	98	62	142	&#098;	b
3	3	003	End of Text	35	23	043	&#035;	#	67	43	103	&#067;	C	99	63	143	&#099;	c
4	4	004	End of Transmission	36	24	044	&#036;	\$	68	44	104	&#068;	D	100	64	144	&#100;	d
5	5	005	Enquiry	37	25	045	&#037;	%	69	45	105	&#069;	E	101	65	145	&#101;	e
6	6	006	Acknowledgment	38	26	046	&#038;	&	70	46	106	&#070;	F	102	66	146	&#102;	f
7	7	007	Bell	39	27	047	&#039;	'	71	47	107	&#071;	G	103	67	147	&#103;	g
8	8	010	Backspace	40	28	050	&#040;	(	72	48	110	&#072;	H	104	68	150	&#104;	h
9	9	011	Horizontal Tab	41	29	051	&#041;	)	73	49	111	&#073;	I	105	69	151	&#105;	i
10	A	012	Line feed	42	2A	052	&#042;	*	74	4A	112	&#074;	J	106	6A	152	&#106;	j
11	B	013	Vertical Tab	43	2B	053	&#043;	+	75	4B	113	&#075;	K	107	6B	153	&#107;	k
12	C	014	Form feed	44	2C	054	&#044;	,	76	4C	114	&#076;	L	108	6C	154	&#108;	l
13	D	015	Carriage return	45	2D	055	&#045;	-	77	4D	115	&#077;	M	109	6D	155	&#109;	m
14	E	016	Shift Out	46	2E	056	&#046;	.	78	4E	116	&#078;	N	110	6E	156	&#110;	n
15	F	017	Shift In	47	2F	057	&#047;	/	79	4F	117	&#079;	O	111	6F	157	&#111;	o
16	10	020	Data Link Escape	48	30	060	&#048;	0	80	50	120	&#080;	P	112	70	160	&#112;	p
17	11	021	Device Control 1	49	31	061	&#049;	1	81	51	121	&#081;	Q	113	71	161	&#113;	q
18	12	022	Device Control 2	50	32	062	&#050;	2	82	52	122	&#082;	R	114	72	162	&#114;	r
19	13	023	Device Control 3	51	33	063	&#051;	3	83	53	123	&#083;	S	115	73	163	&#115;	s
20	14	024	Device Control 4	52	34	064	&#052;	4	84	54	124	&#084;	T	116	74	164	&#116;	t
21	15	025	Negative Ack.	53	35	065	&#053;	5	85	55	125	&#085;	U	117	75	165	&#117;	u
22	16	026	Synchronous idle	54	36	066	&#054;	6	86	56	126	&#086;	V	118	76	166	&#118;	v
23	17	027	End of Trans. Block	55	37	067	&#055;	7	87	57	127	&#087;	W	119	77	167	&#119;	w
24	18	030	Cancel	56	38	070	&#056;	8	88	58	130	&#088;	X	120	78	170	&#120;	x
25	19	031	End of Medium	57	39	071	&#057;	9	89	59	131	&#089;	Y	121	79	171	&#121;	y
26	1A	032	Substitute	58	3A	072	&#058;	:	90	5A	132	&#090;	Z	122	7A	172	&#122;	z
27	1B	033	Escape	59	3B	073	&#059;	;	91	5B	133	&#091;	[	123	7B	173	&#123;	{
28	1C	034	File Separator	60	3C	074	&#060;	<	92	5C	134	&#092;	\	124	7C	174	&#124;	
29	1D	035	Group Separator	61	3D	075	&#061;	=	93	5D	135	&#093;	]	125	7D	175	&#125;	}
30	1E	036	Record Separator	62	3E	076	&#062;	>	94	5E	136	&#094;	^	126	7E	176	&#126;	~
31	1F	037	Unit Separator	63	3F	077	&#063;	?	95	5F	137	&#095;	_	127	7F	177	&#127;	Del

# ANSI Graphics Mode



Try it: echo -e "\e[31mHello  
World in Red\e[34m and blue ;)"

Esc[Value;...;Valuem

## Set Graphics Mode:

Calls the graphics functions specified by the following values. These specified functions remain active until the next occurrence of this escape sequence. Graphics mode changes the colors and attributes of text (such as bold and underline) displayed on the screen.

### Text attributes

- 0** All attributes off
- 1** Bold on
- 4** Underscore (on monochrome display adapter only)
- 5** Blink on
- 7** Reverse video on
- 8** Concealed on

### Foreground colors

- 30** Black
- 31** Red
- 32** Green
- 33** Yellow
- 34** Blue
- 35** Magenta
- 36** Cyan
- 37** White

### Background colors

- 40** Black
- 41** Red
- 42** Green
- 43** Yellow
- 44** Blue
- 45** Magenta
- 46** Cyan
- 47** White

Parameters 30 through 47 meet the ISO 6429 standard.

# ANSI 16 Colors



Campbell

```
Ubuntu x + 
Background | Foreground colors
-----|-----
ESC[40m | [31m [32m [33m [34m [35m [36m [37m
ESC[40m | [1;30m [1;31m [1;32m [1;33m [1;34m [1;35m [1;36m [1;37m
-----|-----
ESC[41m | [30m [31m [32m [33m [34m [35m [36m [37m
ESC[41m | [1;30m [1;31m [1;32m [1;33m [1;34m [1;35m [1;36m [1;37m
-----|-----
ESC[42m | [30m [31m [32m [33m [34m [35m [36m [37m
ESC[42m | [1;30m [1;31m [1;32m [1;33m [1;34m [1;35m [1;36m [1;37m
-----|-----
ESC[43m | [30m [31m [32m [33m [34m [35m [36m [37m
ESC[43m | [1;30m [1;31m [1;32m [1;33m [1;34m [1;35m [1;36m [1;37m
-----|-----
ESC[44m | [30m [31m [32m [33m [34m [35m [36m [37m
ESC[44m | [1;30m [1;31m [1;32m [1;33m [1;34m [1;35m [1;36m [1;37m
-----|-----
ESC[45m | [30m [31m [32m [33m [34m [35m [36m [37m
ESC[45m | [1;30m [1;31m [1;32m [1;33m [1;34m [1;35m [1;36m [1;37m
-----|-----
ESC[46m | [30m [31m [32m [33m [34m [35m [36m [37m
ESC[46m | [1;30m [1;31m [1;32m [1;33m [1;34m [1;35m [1;36m [1;37m
-----|-----
ESC[47m | [30m [31m [32m [33m [34m [35m [36m [37m
ESC[47m | [1;30m [1;31m [1;32m [1;33m [1;34m [1;35m [1;36m [1;37m
```

cinnak@roll:~\$ █

Name	FG Code	BG Code	VGA <sup>[nb 2]</sup>	W
Black	30	40	0,0,0	
Red	31	41	170,0,0	1
Green	32	42	0,170,0	0
Yellow	33	43	170,85,0 <sup>[nb 7]</sup>	1
Blue	34	44	0,0,170	0
Magenta	35	45	170,0,170	1
Cyan	36	46	0,170,170	0
White	37	47	170,170,170	1
Bright Black	90	100	85,85,85	1
Bright Red	91	101	255,85,85	2
Bright Green	92	102	85,255,85	0
Bright Yellow	93	103	255,255,85	2
Bright Blue	94	104	85,85,255	0
Bright Magenta	95	105	255,85,255	2
Bright Cyan	96	106	85,255,255	0
Bright White	97	107	255,255,255	2

[https://en.wikipedia.org/wiki/ANSI\\_escape\\_code](https://en.wikipedia.org/wiki/ANSI_escape_code)

```
git clone https://github.com/garodriguezlp/zsh-bootstrap.git
cd zsh-bootstrap
./demo/colors.sh
```

ANSI color escape sequences are used to change the foreground and background colors of text. The sequences are typically composed of the escape character (ASCII 27, or \u001b), followed by a color index (0-255 for foreground, 40-47 for background), and a reset sequence (0m for foreground, 40m for background). The color index is multiplied by 3 to get the final value.

The following table shows the color combinations produced by the script:

	Normal	Bright	Dark	Light	Yellow	Cyan	Magenta	Blue	Red	Green	White
Normal:	Hello World!										
Bright:	Hello World!										
Normal:	Hello World!										
Bright:	Hello World!										
Normal:	Hello World!										
Bright:	Hello World!										
Normal:	Hello World!										
Bright:	Hello World!										
Normal:	Hello World!										
Bright:	Hello World!										
Normal:	Hello World!										
Bright:	Hello World!										
Normal:	Hello World!										
Bright:	Hello World!										

Try it: curl -s https://raw.githubusercontent.com/garodriguezlp/zsh-bootstrap/master/demo/colors.sh | bash



# The other 50%



```
padding: '12px 14px',  
  
// the full list. if you're going to provide the full color palette,  
// including the 6 x 6 color cubes and the grayscale map, just provide  
// an array here instead of a color map object  
colors: {  
    black: '#000000',  
    red: '#C51E14',  
    green: '#1DC121',  
    yellow: '#C7C329',  
    blue: '#0A2FC4',  
    magenta: '#C839C5',  
    cyan: '#20C5C6',  
    white: '#C7C7C7',  
    lightBlack: '#686868',  
    lightRed: '#FD6F6B',  
    lightGreen: '#67F86F',  
    lightYellow: '#FFFA72',  
    lightBlue: '#6A76FB',  
    lightMagenta: '#FD7CFC',  
    lightCyan: '#68FDFE',  
    lightWhite: '#FFFFFF',  
},  
~,
```

<https://github.com/vercel/hyper/blob/master/app/config/config-default.js#L75>

## Creating your own color scheme

Color schemes can be defined in the `schemes` array of your `settings.json` file. They are written in the following format:

JSON

```
{  
    "name" : "Campbell",  
  
    "cursorColor": "#FFFFFF",  
    "selectionBackground": "#FFFFFF",  
  
    "background" : "#0C0C0C",  
    "foreground" : "#CCCCCC",  
  
    "black" : "#0C0C0C",  
    "blue" : "#0037DA",  
    "cyan" : "#3A96DD",  
    "green" : "#13A10E",  
    "purple" : "#881798",  
    "red" : "#C50F1F",  
    "white" : "#CCCCCC",  
    "yellow" : "#C19C00",  
    "brightBlack" : "#767676",  
    "brightBlue" : "#3B78FF",  
    "brightCyan" : "#61D6D6",  
    "brightGreen" : "#16C60C",  
    "brightPurple" : "#B4009E",  
    "brightRed" : "#E74856",  
    "brightWhite" : "#F2F2F2",  
    "brightYellow" : "#F9F1A5"  
},
```

<https://docs.microsoft.com/en-us/windows/terminal/customize-settings/color-schemes>

I'm a techie, not an artist   
What can I do?

Don't worry my friend ,  
there're many awesome themes and  
tools out there   

Dracula — Dark theme for 107+ + X

draculatheme.com

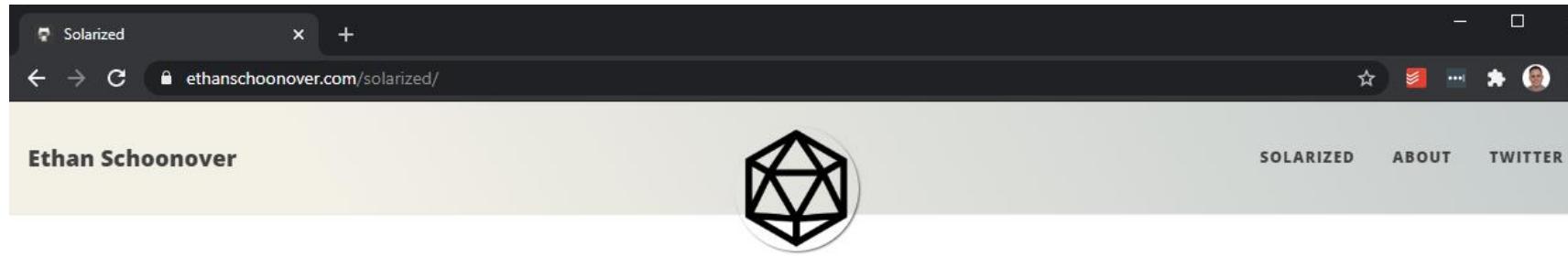
About Contribute Dracula PRO

The image shows a screenshot of a web browser window displaying the Dracula theme homepage. The browser's interface is also styled with the Dracula theme, featuring a dark background with light-colored text and icons. The main content area features a large, stylized logo of a vampire with white skin, black hair, and a pink bow tie, set against a dark circular background. Below the logo, the word "Dracula" is written in a large, bold, purple font. Underneath "Dracula", the text "Dark theme for Emacs" is displayed in a smaller, white font. At the bottom of the page, there are two GitHub-style buttons: "Star" with the number 12,972 and "Fork" with the number 1,153.

Dracula

Dark theme for Emacs

Star 12,972 Fork 1,153



**Note: SITE RESTORATION IN PROGRESS**

## Solarized

Precision colors for machines and people

A circular graphic featuring a snippet of code in the center. The code appears to be written in a programming language like Scala or Java, dealing with file operations and data structures. The background of the circle is a gradient of green and blue, and the text is in a monospaced font.

Monokai Pro

x +

← → C monokai.pro

☆ ⚡ ... 🧩 🧮 :



# Monokai Pro

BEAUTIFUL FUNCTIONALITY FOR PROFESSIONAL DEVELOPERS



Sublime Text



VS Code

A color scheme for focusing

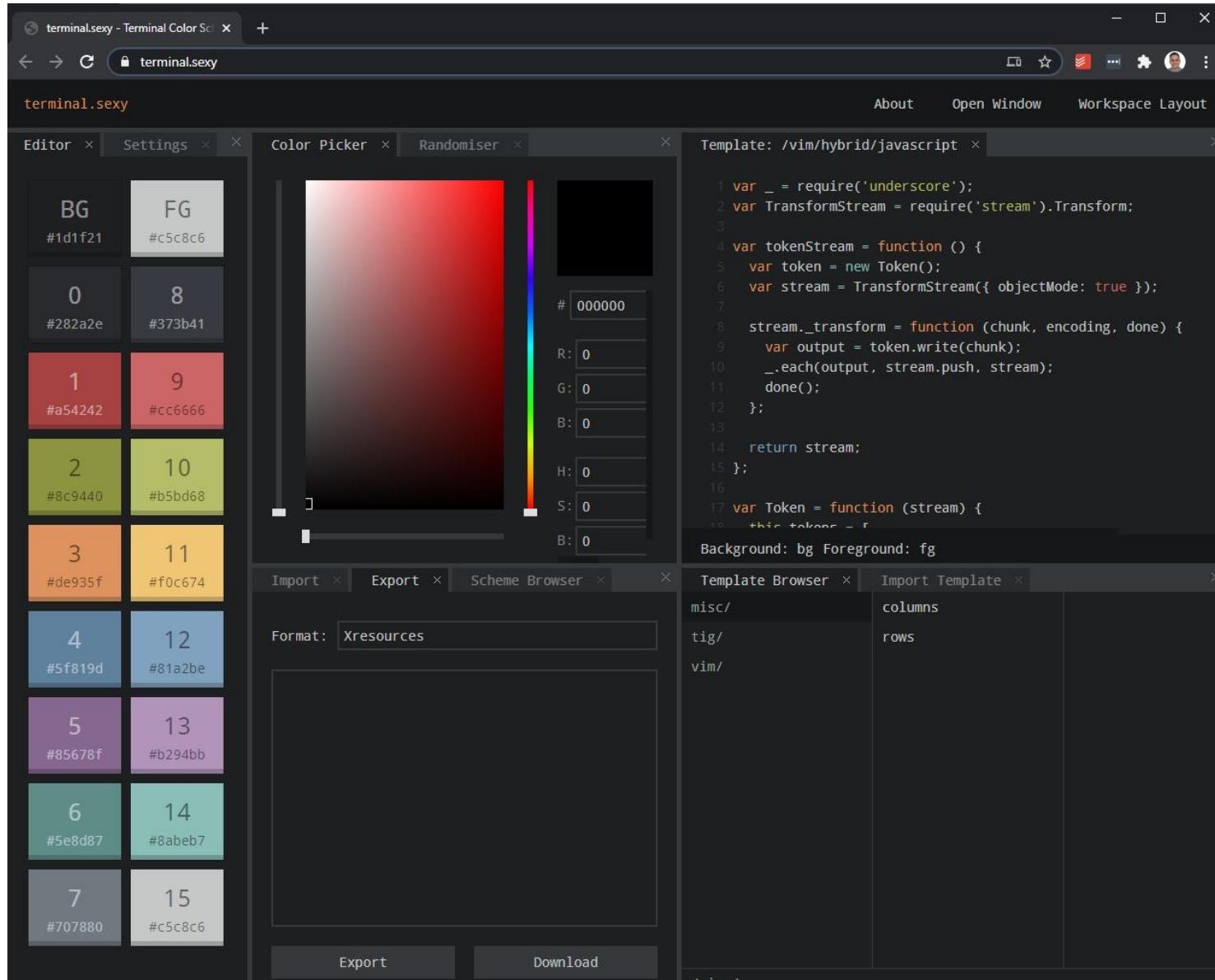
S 46 Best color themes for text edit X +

← → C 🔒 slant.co/topics/358/~best-color-themes-for-text-editors

BEST COLOR THEMES FOR TEXT EDITORS	PRICE	LAST UPDATED
96  Tomorrow Theme	-	Jul 14, 2020
93  Gruvbox	FREE	Jun 18, 2020
90  Dracula	-	Jul 7, 2020
89  Base16 - Eighties Dark	-	Jul 8, 2020
87  Molokai	-	Jun 19, 2020

SEE FULL LIST

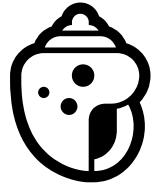
<https://www.slant.co/topics/358/~best-color-themes-for-text-editors>



# terminal.sexey



# Shell\* features



\* ZSH (Way more than a command line interpreter)

It sounds boring! Do I really need to know these?

Definitely, yes! Why? Let me explain

- Are the base on plugins and tools (the tuning)
- We tend to pay much attention to the fancy stuff, overlooking the value of the simple/base things.

# Let me get this straight!



~ 1989

A screenshot of the IntelliJ IDEA IDE. The left side shows a project structure with several packages like 'RegExSupport', 'resources', 'spellchecker', and 'testSrc'. The right side shows the code editor with Java code for 'CreateHtmlFileAction.java'. The code handles creating HTML files, including support for different template types like HTML 5, HTML 4, and XHTML.

```
intellij-community [~/IDEAProjects/intelliж-community] - .../xml/impl/src/com/intelliж/ide/actions/CreateHtmlFileAction.java [intelliж.xml.impl]
Project CreateHtmlFileAction.java CoverageSuitesBundle.java
30
31     @NotNull private static final String DEFAULT_HTML_TEMPLATE_PROPERTY = <!-->
32     "DefaultHtmlTemplate";
33
34     public CreateHtmlFileAction() {
35         super("HTML File", "Creates new HTML file", StdFileTypes.HTML.getIcon());
36     }
37
38     @Override
39     protected String getDefaultTemplateProperty() { return <!-->
40         DEFAULT_HTML_TEMPLATE_PROPERTY; }
41
42     @Override
43     protected void buildDialog(Project project, PsiDirectory directory, <!-->
44         CreateFileDialogFromTemplateDialog.Builder builder) {
45         builder
46             .setTitle("New HTML File")
47             .addKind(Kind: "HTML 5 file", StdFileTypes.HTML.getIcon(), <!-->
48                 FileTemplateManager.INTERNAL_HTML5_TEMPLATE_NAME)
49             .addKind(Kind: "HTML 4 file", StdFileTypes.HTML.getIcon(), <!-->
50                 FileTemplateManager.INTERNAL_HTML_TEMPLATE_NAME)
51             .addKind(Kind: "XHTML file", StdFileTypes.XHTML.getIcon(), <!-->
52                 FileTemplateManager.INTERNAL_XHTML_TEMPLATE_NAME);
53
54
55     @Override
56     protected String getActionName(PsiDirectory directory, String newName, <!-->
57         String templateName) {
58         return "HTML File";
59     }
60
61     @Override
62     public void actionPerformed(ActionEvent e) {
63         CreateHtmlFileAction.super.getDefaultTemplateProperty();
64     }
65 }
```

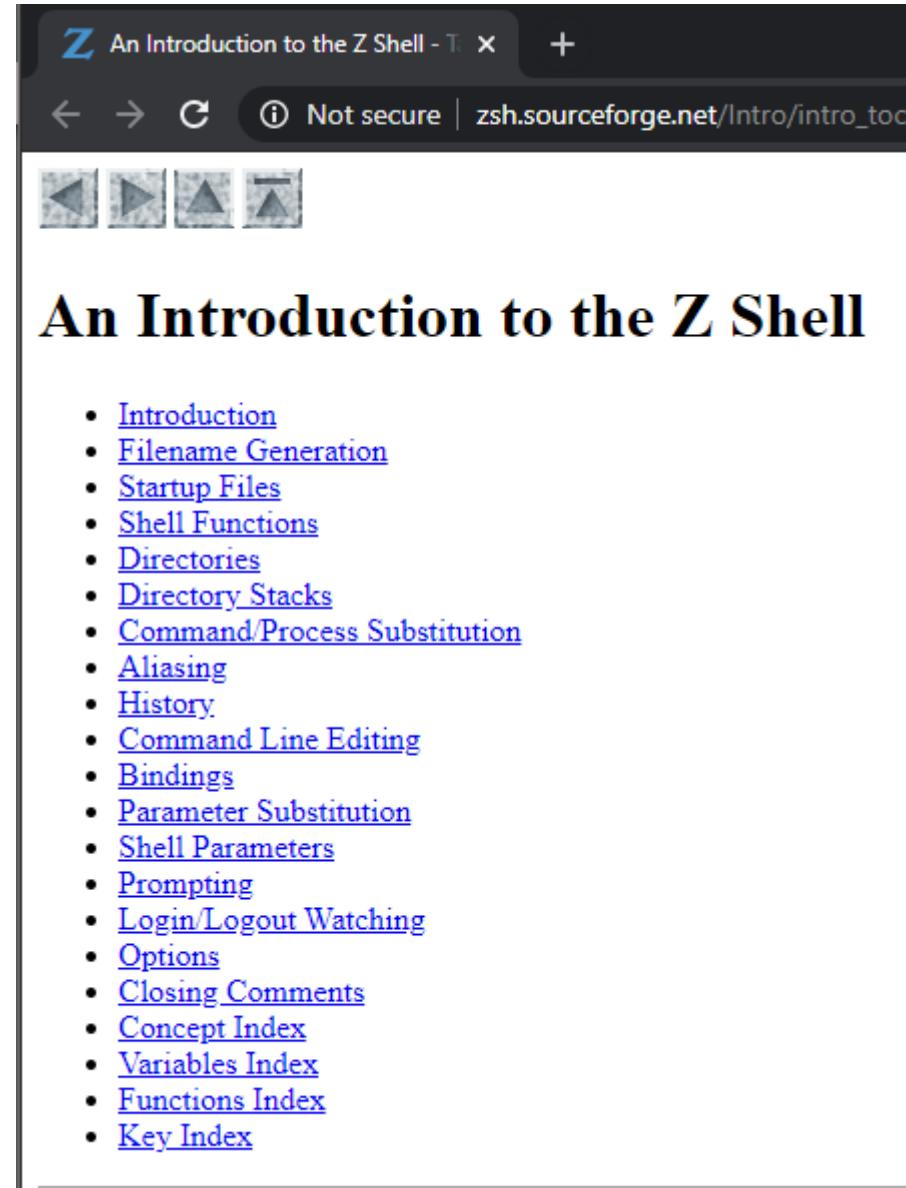


~ 1970

A screenshot of a terminal window titled 'Terminal' on a Linux system. The prompt shows the user is at 'me@linuxbox:~\$'. The terminal is currently empty, showing only the title bar and a few icons.

- Ok, I'm ready! Hit me with all I'll need to learn.

- Are you sure? 😊 ok, look ➡



The screenshot shows a web browser window with the title "An Introduction to the Z Shell - T". The address bar indicates the page is "Not secure" and the URL is "zsh.sourceforge.net/Intro/intro\_toc.html". Below the address bar are four small navigation icons. The main content area features a large, bold heading "An Introduction to the Z Shell". Below it is a bulleted list of topics:

- [Introduction](#)
- [Filename Generation](#)
- [Startup Files](#)
- [Shell Functions](#)
- [Directories](#)
- [Directory Stacks](#)
- [Command/Process Substitution](#)
- [Aliasing](#)
- [History](#)
- [Command Line Editing](#)
- [Bindings](#)
- [Parameter Substitution](#)
- [Shell Parameters](#)
- [Prompting](#)
- [Login/Logout Watching](#)
- [Options](#)
- [Closing Comments](#)
- [Concept Index](#)
- [Variables Index](#)
- [Functions Index](#)
- [Key Index](#)

[http://zsh.sourceforge.net/Intro/intro\\_toc.html#SEC3](http://zsh.sourceforge.net/Intro/intro_toc.html#SEC3)

- No way, that's too much! Tell me where should I focus my attention first...

- Ok  

- Startup Files
- Shell Functions
- Aliasing
- History
- Command Line Editing
- Bindings
- Shell Parameters
- Prompting
- Completions

# Why do I need to care about: Startup Files? 😐

Go to the [first](#), [previous](#), [next](#), [last](#) section, [table of contents](#).

## Startup Files

There are five startup files that zsh will read commands from:

```
$ZDOTDIR/.zshenv  
$ZDOTDIR/.zprofile  
$ZDOTDIR/.zshrc  
$ZDOTDIR/.zlogin  
$ZDOTDIR/.zlogout
```

[http://zsh.sourceforge.net/Intro/intro\\_3.html#SEC3](http://zsh.sourceforge.net/Intro/intro_3.html#SEC3)

Not Secure | catb.org/jargon/html/R/rc-file.html

rc file

Prev ? R ? Next

rc file: /RRC fi:l, n.

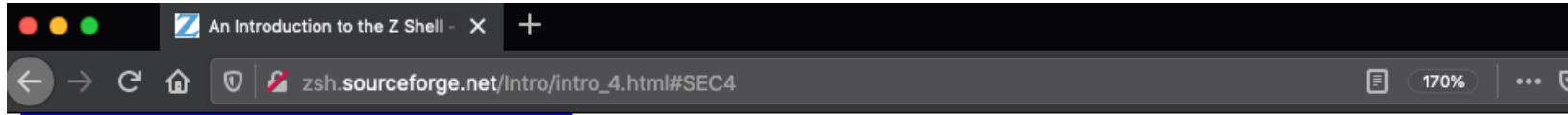
[Unix: from runcom files on the [CTSS](#) system 1962-63, via the startup script `/etc/rc`] Script file containing startup instructions for an application program (or an entire operating system), usually a text file containing commands of the sort that might have been invoked manually once the system was running but are to be executed automatically each time the system starts up. See also [dot file](#), [profile](#) (sense 1).

Prev ? Up ? Next  
RBL ? Home ? RE

<http://www.catb.org/jargon/html/R/rc-file.html>

Because: It's where all steroids live

# Why do I need to care about: Shell Functions? 😐



zsh also allows you to create your own commands by defining shell functions. For example:

```
% yp () {  
>     ypmatch $1 passwdbyname  
> }  
% yp pfalstad  
pfalstad:*:3564:35:Paul John Falstad:/u/pfalstad:/usr/princeton/bin/zsh
```

[http://zsh.sourceforge.net/Intro/intro\\_4.html#SEC4](http://zsh.sourceforge.net/Intro/intro_4.html#SEC4)

Because: That's how we get to the doing more while typing less or doing more valuable tasks that we have never even imagined 😊

# Why do I need to care about: Aliasing?



Go to the [first](#), [previous](#), [next](#), [last](#) section, [table of contents](#).

---

## Aliasing

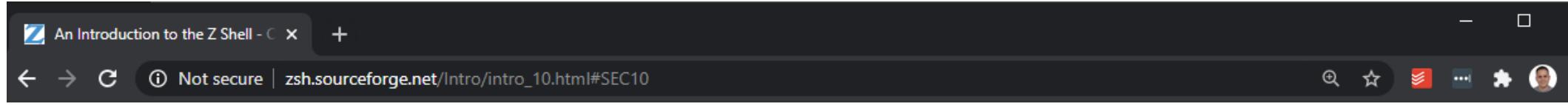
Often-used commands can be abbreviated with an alias:

```
% alias uc=uncompress  
% ls  
hanoi.Z  
% uc hanoi  
% ls  
hanoi
```

[http://zsh.sourceforge.net/Intro/intro\\_8.html#SEC8](http://zsh.sourceforge.net/Intro/intro_8.html#SEC8)

Because: No one should enjoy typing the same repeatedly 

# Why do I need to care about: Command Line Editing? 😎



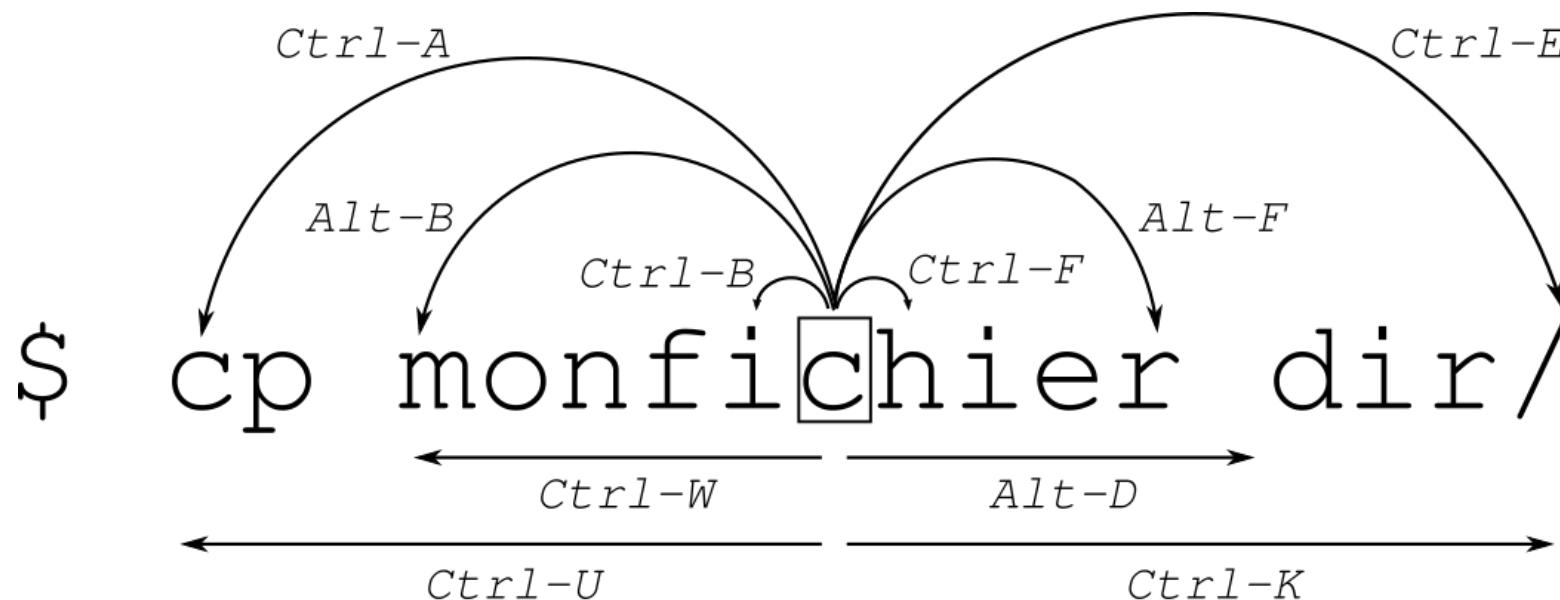
zsh's command line editor, **ZLE**, is quite powerful. It is designed to emulate either emacs or vi; the default is emacs. To set the bindings for vi mode, type `bindkey -v`.

In addition to basic editing, the shell allows you to recall previous lines in the history. In emacs mode, this is done with `^P` (control-P):

[http://zsh.sourceforge.net/Intro/intro\\_10.html#SEC10](http://zsh.sourceforge.net/Intro/intro_10.html#SEC10)

Because: It's how you get to edit fast 💨

# GNU Readline



} Moving  
}  
} Erasing

[https://clementc.github.io/blog/2018/01/25/moving\\_cli/](https://clementc.github.io/blog/2018/01/25/moving_cli/)

# GNU Readline (emacs)



Bash Emacs Editing Mode (readline) x +

scribd.com/document/985254/Bash-Emacs-Editing-Mode-readline-Cheat-Sheet

SCRIBD Search Q Upload EN ▾ Sign In Read Free For 14 Days

Learn more about Scribd Membership >

Download Now

1 of 2 Q Q

Search document

Home Saved Bestsellers

Readline Emacs Editing Mode Cheat Sheet  
Default Keyboard Shortcuts for Bash

<https://www.scribd.com/document/985254/Bash-Emacs-Editing-Mode-readline-Cheat-Sheet>

# Why do I need to care about: Shell Parameters (env vars)?



An Introduction to the Z Shell - S X +  
← → C ⓘ Not secure | zsh.sourceforge.net/Intro/intro\_13.html#SEC13

Go to the [first](#), [previous](#), [next](#), [last](#) section, [table of contents](#).

---

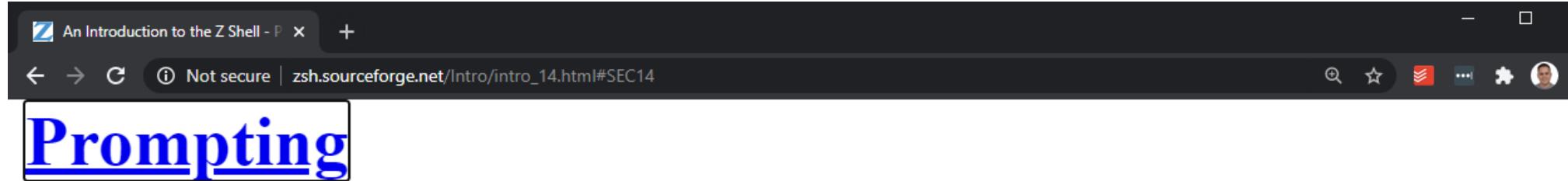
## Shell Parameters

The shell has many predefined parameters that may be accessed. Here are some examples:

[http://zsh.sourceforge.net/Intro/intro\\_13.html#SEC13](http://zsh.sourceforge.net/Intro/intro_13.html#SEC13)

Because: it's where the real tuning of your tools happens

# Why do I need to care about: Prompting? 😎



The default prompt for zsh is:

```
phoenix% echo $PROMPT  
%m%#
```

The %m stands for the short form of the current hostname, and the %# stands for a % or a #, depending on whether the shell is running as root or not. zsh supports many other control sequences in the PROMPT variable.

[http://zsh.sourceforge.net/Intro/intro\\_14.html#SEC14](http://zsh.sourceforge.net/Intro/intro_14.html#SEC14)

Because: It's the feature that will make you look sexy, I mean, like a sexy hacker and be happy 😎

 denysdovhan /  
spaceship-prompt

 Sponsor



Watch ▾

90



Star

12.5k

README.md



 Spaceship ZSH

 prompt for Astronauts.

npm v3.11.2

build passing

zsh >=v5.2

twitter

@SpaceshipZSH

support patreon

  
Now part of 

README.md

# starship

CROSS-SHELL PROMPT

[workflow](#) [failing](#) [crates.io v0.44.0](#) [in repositories 11](#)[discord](#) [56 online](#) [twitter](#) [@StarshipPrompt](#)[Website](#) · [Installation](#) · [Configuration](#)

The minimal, blazing-fast, and infinitely customizable prompt for any shell!

- Fast: it's fast – *really really* fast! 🚶
- Customizable: configure every aspect of your prompt.
- Universal: works on any shell, on any operating system.
- Intelligent: shows relevant information at a glance.
- Feature rich: support for all your favorite tools.
- Easy: quick to install – start using it in minutes.

```
fish /Users/matan/dev

~/dev
> false

~/dev
> true

~/dev
> cd rocket
```

[Explore the Starship docs](#) ➔

 sindresorhus / pure

 Sponsor

 Watch ▼

79

 Star

9.1k

readme.md

## Pure

Pretty, minimal and fast ZSH prompt

```
● ● ●  
~  
> exit 1  
  
~  
> cd pure  
~/dev/pure  
  
~/dev/pure master ↓  
> touch dirty  
  
~/dev/pure master* ↓  
> echo much pure
```



romkatv / powerlevel10k

Watch ▾

86

Star

9.1k

README.md

## Powerlevel10k

chat on gitter

Powerlevel10k is a theme for Zsh. It emphasizes [speed](#), [flexibility](#) and [out-of-the-box experience](#).

The screenshot shows a terminal window with three distinct sections, each demonstrating a different style of the Powerlevel10k theme:

- Lean Style:** The first section shows a minimalist interface with a black background and white text. It displays the command `~/src master >` and a timestamp `took 5s at 16:23:42`. A small red box highlights the timestamp.
- Classic Style:** The second section shows a more traditional style with a dark background and light text. It displays the command `~/src > master` and a timestamp `took 5s at 16:23:42`. A small red box highlights the timestamp.
- Rainbow Style:** The third section shows a vibrant, multi-colored style with a dark background. It displays the command `~/src master` and a timestamp `took 5s at 16:23:42`. A small yellow box highlights the timestamp.

# Why do I need to care about: Completions? 🤔

---

## Chapter 6: Completion, old and new

Completion of command arguments is something zsh is particularly good at. The simplest case is that you hit `<TAB>`, and the shell guesses what has to go there and fills it in for you:

```
% ls  
myfile theirfile yourfile  
% cat t<TAB>
```

expands the command line to

```
% cat theirfile
```

and you only had to type the initial letter, then TAB.

<http://zsh.sourceforge.net/Guide/zshguide06.html#l144>

Because: It'd alleviate a lot the memorization effort and because you'll love it 😍

[zsh-users/zsh-completions](#)

Watch 97 Star 3.7k Fork 538

Code Issues 20 Pull requests 4 Actions Security ...

master Go to file Add file Code

**syohex Merge pull request #754 from ka...** 6 days ago 1,349

- .github #591: move CONTRIBUTING to root 2 years ago
- src incorporated code review comm... 6 days ago
- .editorconfig Add EditorConfig file denoting ind... 8 years ago
- .gitignore git ignore some IDE files 2 years ago
- CONTRIBUTIN... Fix typos 4 months ago
- LICENSE Change default license to the Zsh l... 3 years ago
- README.md change os name: Mac OS to macO... 6 months ago
- zsh-completio... replace broken link to linux-mag ... 3 months ago
- zsh-completio... Use modifier expansion instead of ... 3 years ago

**README.md**

## zsh-completions

release v0.31.0

contributors 203 IRC #zsh-completions chat on gitter

Additional completion definitions for Zsh.

This project aims at gathering/developing new completion scripts that are not available in Zsh yet. The scripts may be contributed to the Zsh project when stable enough.

About

Additional completion definitions for Zsh.

zsh zsh-completions shell completion

Readme View license

Releases 36

0.31.0 Latest on Sep 20, 2019 + 35 releases

Contributors 213

+ 202 contributors Languages

Shell 100.0%

0 • 1 zsh

```
> git commit --all
--all
--allow-empty
--allow-empty-message
--amend
--author
--branch
--cleanup
--date
--dry-run
--edit
--file
--gpg-sign
--include
--interactive
--message
--no-ahead-behind
--no-edit
--no-gpg-sign
--no-post-rewrite
--no-status
--no-verify
--null
--only
--patch
```

-- stage all modified and deleted paths  
-- allow recording an empty commit  
-- allow recording a commit with an empty message  
-- amend the tip of the current branch  
-- override the author name used in the commit  
-- show branch information  
-- specify how the commit message should be cleaned up  
-- override the author date used in the commit  
-- only show list of paths that are to be committed or not, and any untracked  
-- edit the commit message before committing  
-- read commit message from given file  
-- GPG-sign the commit  
-- update the given files and commit the whole index  
-- interactively update paths in the index file  
-- use the given message as the commit message  
-- don't display detailed ahead/behind counts relative to upstream branch  
-- do not edit the commit message before committing  
-- don't GPG-sign the commit  
-- bypass the post-rewrite hook  
-- do not include the output of git status in the commit message template  
-- bypass pre-commit and commit-msg hooks  
-- dry run with NULL-separated output format  
-- commit only the given files  
-- use the interactive patch selection interface to chose which changes to co

0 ↑ 1m 1 zsh

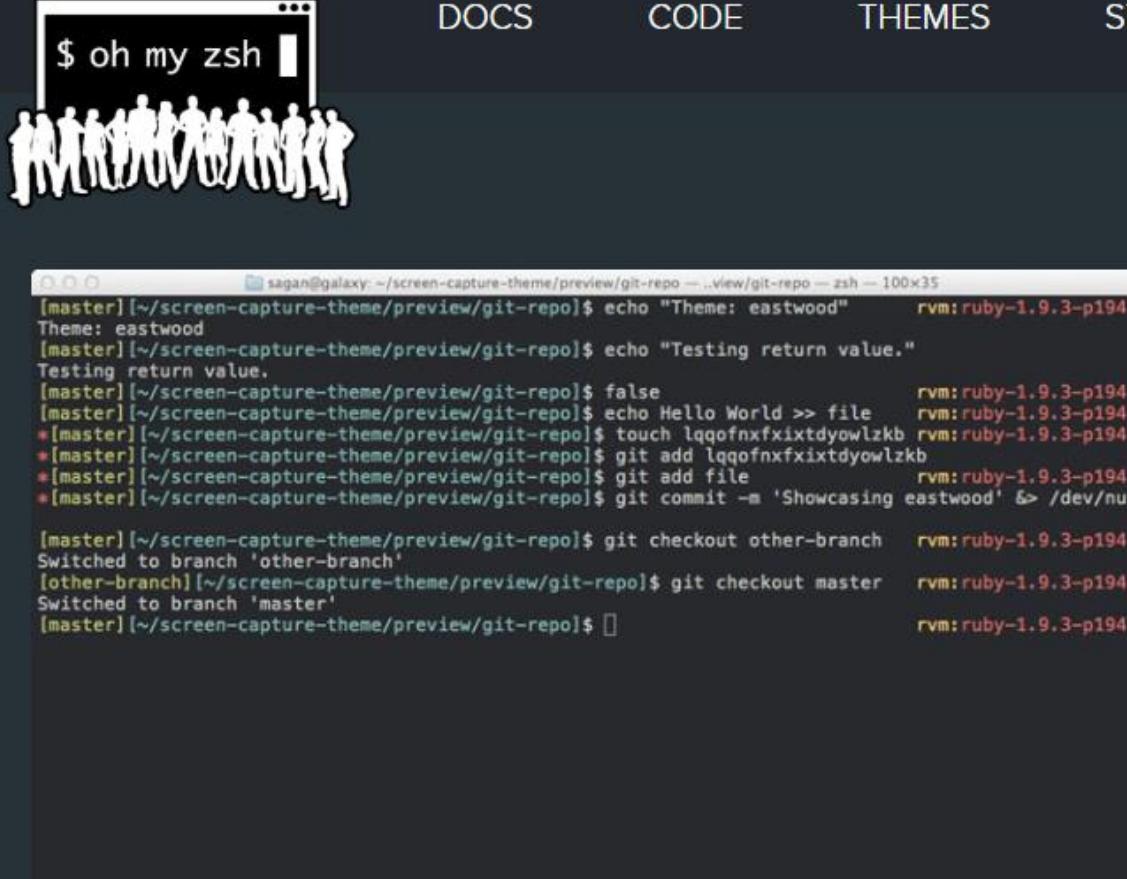
# Plugins



# Demystifying plugins

- All are mere zsh/bash scripts sourced/loaded at the startup, which define things like:
  - Aliases
  - Functions
  - Key bindings
  - Completions
  - ....
  - Enhancements (wrappers)  to awesome CLI tools (programs/apps)

# Oh-my-zsh



```
[master] ~/screen-capture-theme/preview/git-repo$ echo "Theme: eastwood"      rvm:ruby-1.9.3-p194
Theme: eastwood
[master] ~/screen-capture-theme/preview/git-repo$ echo "Testing return value."
Testing return value.
[master] ~/screen-capture-theme/preview/git-repo$ false                         rvm:ruby-1.9.3-p194
[master] ~/screen-capture-theme/preview/git-repo$ echo Hello World >> file    rvm:ruby-1.9.3-p194
*[master] ~/screen-capture-theme/preview/git-repo$ touch lqqofnxfixtdyowlzkb rvm:ruby-1.9.3-p194
*[master] ~/screen-capture-theme/preview/git-repo$ git add lqqofnxfixtdyowlzkb
*[master] ~/screen-capture-theme/preview/git-repo$ git add file                  rvm:ruby-1.9.3-p194
*[master] ~/screen-capture-theme/preview/git-repo$ git commit -m 'Showcasing eastwood' &> /dev/null

[master] ~/screen-capture-theme/preview/git-repo$ git checkout other-branch     rvm:ruby-1.9.3-p194
Switched to branch 'other-branch'
[other-branch] ~/screen-capture-theme/preview/git-repo$ git checkout master      rvm:ruby-1.9.3-p194
Switched to branch 'master'
[master] ~/screen-capture-theme/preview/git-repo$ 
```

THEMES

STORE

COMMUNITY

Fork me on GitHub

Your terminal never  
felt *this* good before.

Oh My Zsh is a delightful, open source, community-driven framework for managing your Zsh configuration. It comes bundled with thousands of helpful functions, helpers, plugins, themes, and a few things that make you shout...

"Oh My ZSH!"

Install oh-my-zsh

# Oh-my-zsh

- 2 things at the same time:
  - Plugin and config manager
  - Plugin library
- 280+ plugins
- But...



**Robby Russell**  
robbyrussell

VP Engineering/Partner  
[@planetargon](#). Old-timer Ruby on Rails developer. Creator of [@ohmyzsh](#). Host of Maintainable Software Podcast  
he/him/his

Follow

Sponsor

...

3.6k followers · 33 following · 105

# Antigen



Antigen v2

The plugin manager for zsh.

→ C [github.com/zsh-users/antigen#motivation](https://github.com/zsh-users/antigen#motivation)

## Motivation

If you use zsh and [oh-my-zsh](#), you know that having many different plugins that are developed by many different authors in a single (sub)repo is not very easy to maintain. There are some really fantastic plugins and utilities in oh-my-zsh, but having them all in a single repo doesn't really scale well. And I admire robbyrussell's efforts for reviewing and merging the gigantic number of pull requests the project gets. We need a better way of plugin management.

This was discussed on [a few issues](#), but it doesn't look like there was any progress made. So, I'm trying to start this off with Antigen, hoping to better this situation. Please note that I'm by no means a zsh or any shell script expert (far from it).

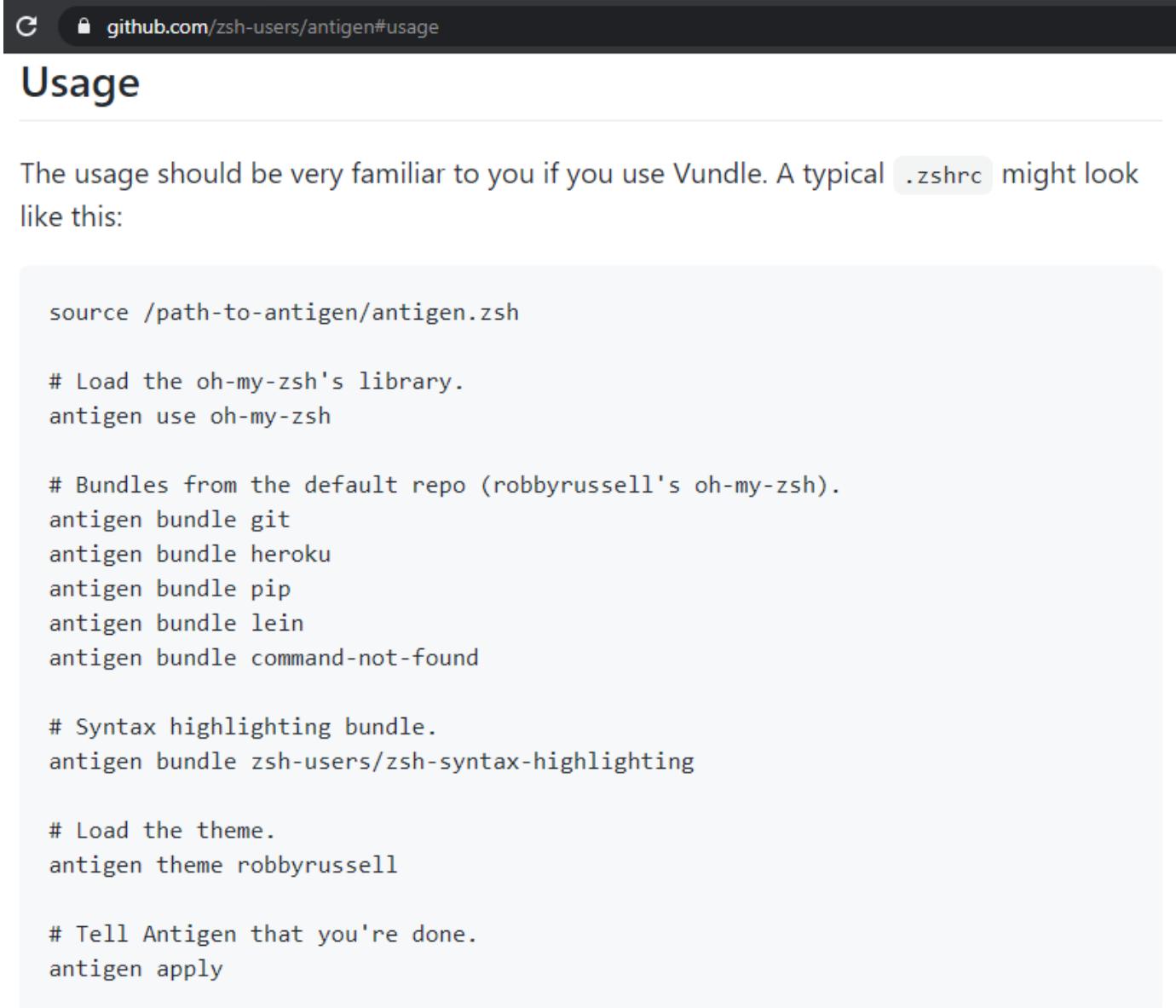
Inspired by vundle, Antigen can pull oh-my-zsh style plugins from various github repositories. You are not limited to use plugins from the oh-my-zsh repository only and you don't need to maintain your own fork and pull from upstream every now and then. I actually encourage you to grab plugins and scripts from various sources, straight from the authors, before they even submit it to oh-my-zsh as a pull request.

Antigen also lets you switch the prompt theme with one command, just like that

<https://github.com/zsh-users/antigen#motivation>

# Antigen

- Only 1 thing:  
Plugin and  
config manager



The usage should be very familiar to you if you use Vundle. A typical `.zshrc` might look like this:

```
source /path-to-antigen/antigen.zsh

# Load the oh-my-zsh's library.
antigen use oh-my-zsh

# Bundles from the default repo (robbryussell's oh-my-zsh).
antigen bundle git
antigen bundle heroku
antigen bundle pip
antigen bundle lein
antigen bundle command-not-found

# Syntax highlighting bundle.
antigen bundle zsh-users/zsh-syntax-highlighting

# Load the theme.
antigen theme robbryussell

# Tell Antigen that you're done.
antigen apply
```

# Prezto

sorin-inescu / prezto

Watch 223 Star 11.4k

Code Issues Pull requests Actions Projects Security Insights

master 34 branches 0 tags Go to file Add file Code

oohlaf Add check for empty WORKON\_HOME (#1861) e919723 7 hours ago 1,889 commits

.github Adjust GitHub templates with commented lines 3 years ago

modules Add check for empty WORKON\_HOME (#1861) 7 hours ago

runcoms Make .zcomp\* location configurable (#1842) 2 months ago

.editorconfig Small editorconfig file cleanup 3 years ago

.gitignore Allow modules to be loaded from multiple places (#1458) 3 years ago

.gitmodules Use shallow clone for powerlevel9k and powerlevel10k submodules (#1... 4 months ago

CONTRIBUTING.md Fix spelling error in CONTRIBUTING.md (#1590) 2 years ago

LICENSE Bump license years and tidy up readme (#1273) 3 years ago

README.md README: fix small grammatical error (#1857) 7 days ago

init.zsh Allow module functions directory to be a symlink 6 months ago

README.md

## Prezto — Instantly Awesome Zsh

Prezto is the configuration framework for [Zsh](#); it enriches the command line interface environment with sane defaults, aliases, functions, auto completion, and prompt themes.

About

The configuration fram

Readme

MIT License

Releases

No releases published

Contributors 297

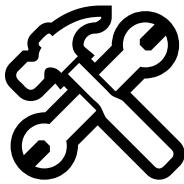
+ 286 contributors

Languages

Shell 100.0%

This image shows a screenshot of the GitHub repository page for 'prezto' by 'sorin-inescu'. The repository has 223 watchers and 11.4k stars. The 'Code' tab is selected, showing a list of recent commits. The commits are from 'oohlaf' and include changes like 'Add check for empty WORKON\_HOME', 'Adjust GitHub templates', and 'Make .zcomp\* location configurable'. The repository also contains files like '.github', 'modules', 'runcoms', '.editorconfig', '.gitignore', '.gitmodules', 'CONTRIBUTING.md', 'LICENSE', 'README.md', and 'init.zsh'. On the right side, there are sections for 'About', 'Releases', 'Contributors', and 'Languages', with 'Shell' being the primary language at 100.0%.

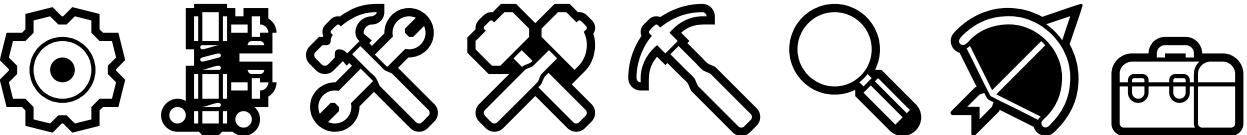
# Tools



Plugins, frameworks, tools, tricks that let us get the most  
of a given shell



# CLI Tools

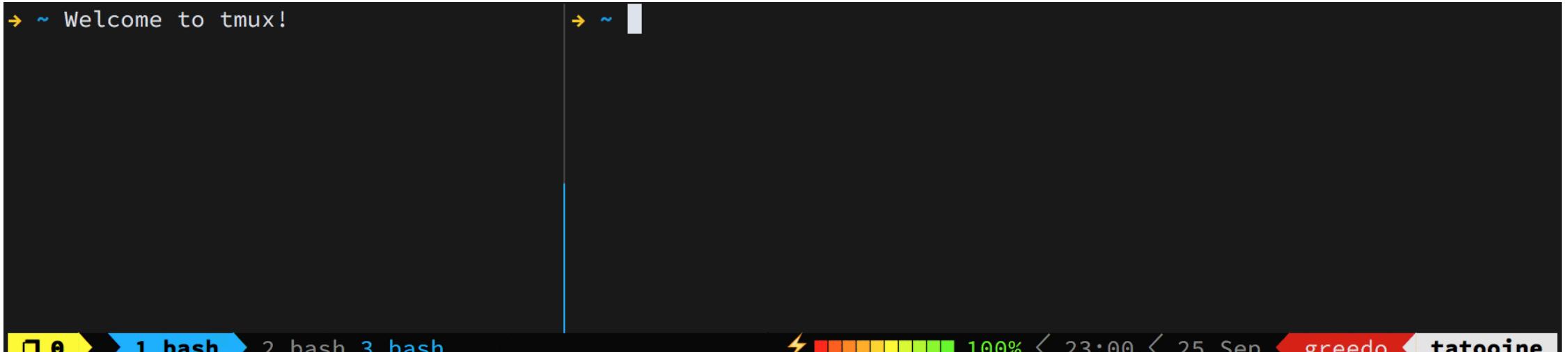


- There are thousands of awesome 😎 Unix CLI tools
- But, it's important that you differentiate the two 🤝 sources of power ⚡ in the CLI and how each other relate or complement 💥:
- CLI programs vs. Tools build with shell features

# My Favorites



# Tmux: A terminal multiplexer



→ ~ Welcome to tmux!

→ ~ [ ]

□ e ➤ 1 bash ➤ 2 bash 3 bash

⚡ [██████████] 100% < 23:00 < 25 Sep ➤ greedo ➤ tatooine

<https://github.com/gpakosz/.tmux>



<https://github.com/tmux/tmux/>



# Homebrew

The Missing Package Manager for macOS (or Linux)

Search Homebrew

English ▾

## Install Homebrew

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
```

Paste that in a macOS Terminal or Linux shell prompt.

The script explains what it will do and then pauses before it does it. Read about other [installation options](#).



fzf is a general-purpose command-line fuzzy finder.

```

src/util/eventbox_test.go
src/curses/curses_test.go
src/util/atomicbool.go
src/util/util_test.go
src/tokenizer_test.go
src/chunklist_test.go
src/algo/algo_test.go
src/util/eventbox.go
src/curses/curses.go
src/pattern_test.go
src/options_test.go
src/history_test.go
src/reader_test.go
src/merger_test.go
src/cache_test.go
src/util/util.go
src/tokenizer.go
src/item_test.go
src/constants.go
src/chunklist.go
src/ansi_test.go
src/algo/algo.go
src/terminal.go
src/fzf/main.go
src/pattern.go
> src/options.go
src/merger.go
src/history.go
src/reader.go
src/merger.go
src/cache.go
src/item.go
src/core.go
src/ansi.go
35/63
> .go$||

package fzf

import (
    "fmt"
    "os"
    "regexp"
    "strconv"
    "strings"
    "unicode/utf8"
    "github.com/junegunn/fzf/src/curses"
    "github.com/junegunn/go-shellwords"
)

const usage = `usage: fzf [options]

Search
  --extended      Extended-search mode
                  (enabled by default; +x or --no-extended to disable)
  --exact         Enable Exact-match
  --i             Case-insensitive match (default: smart-case match)
  --n, --nth=N[...]
                  Comma-separated list of field index expressions
                  for limiting search scope. Each can be a non-zero
                  integer or a range expression ([BEGIN]..[END]).
  --with-nth=N[...]
                  Transform the presentation of each line using
                  field index expressions
  --d, --delimiter=STR Field delimiter regex (default: AWK-style)
  --no-sort       Do not sort the result
  --tac           Reverse the order of the input
  --tiebreak=CRI[...]
                  Comma-separated list of sort criteria to apply
                  when the scores are tied [length|begin|end|index]
                  (default: length)`
```

It's an interactive Unix filter for command-line that can be used with any list; files, command history, processes, hostnames, bookmarks, git commits, etc.

## Feature comparison of ack, ag, git-grep, GNU grep and ripgrep

There are many greplike tools available to choose from. ripgrep is currently the fastest, which folks seem to focus on, but they have different features. Here's a chart to help you see which tools can do what.

If you find this chart useful, consider making one for the projects you run, even if it includes "competing" projects. [The best project for someone might not be your project, and that's OK.](#)

If you have updates to the chart, please [submit as a GitHub issue](#).

	ack	ag	git-grep	GNU grep	rg
Name	ack	The Silver Searcher	git-grep	GNU grep	ripgrep
Project page	<a href="#">Link</a>	<a href="#">Link</a>	<a href="#">Link</a>	<a href="#">Link</a>	<a href="#">Link</a>
Language	Perl	C	C	C	Rust
License	Artistic License 2.0	Apache License 2.0	GPL 2	GPL 3	MIT/Unlicense
Regular expressions	Native Perl	Perl-compatible	Basic, extended, or Perl-compatible	Basic, extended, or Perl-compatible (experimental)	Fast non-backtracking, Perl-compatible, or automatic selection
Feature	ack	ag	git-grep	GNU grep	rg

[Code](#)[Issues 131](#)[Pull requests 36](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)[Insights](#)[master](#)[1 branch](#)[53 tags](#)[Go to file](#)[Add file](#)[Code](#)

wting Merge branch 'lilydjwg\_move\_temp\_file\_to\_same\_device'

06e082c on Sep 9, 2018

839 commits



Bump to 22.5.3.

2 years ago



Bump to 22.5.3.

2 years ago



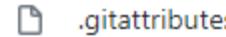
Add match consecutive tests.

4 years ago



(Running pre-commit) Use Yelp-style indentation.

3 years ago



(Running pre-commit) Run end of file fixer.

3 years ago



.lanore .nvtest cache/

2 years ago

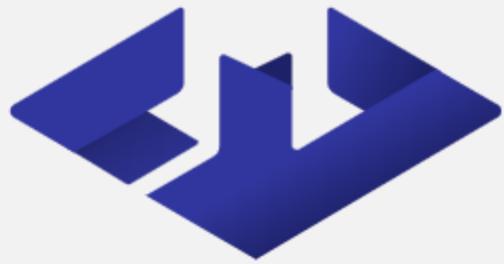
## About

A cd command that learns - easily navigate directories from the command line

autojump python filesystem  
command-line-tool

[Readme](#)[View license](#)

README.md



B A T  
A cat clone with wings

[build](#) passing [build](#) passing [license](#) MIT/Apache-2.0 [crates.io](#) v0.15.4

A *cat(1)* clone with syntax highlighting and Git integration.

[Key Features](#) • [How To Use](#) • [Installation](#) • [Customization](#) • [Project goals, alternatives](#)

[\[中文\]](#) [\[日本語\]](#) [\[한국어\]](#) [\[Русский\]](#)

## Syntax highlighting

`bat` supports syntax highlighting for a large number of programming and markup languages:



jq is a lightweight and flexible command-line JSON processor.

[Download jq 1.6 ▾](#)[Try online at \[jqplay.org\]\(#\)!](#)

jq is like `sed` for JSON data - you can use it to slice and filter and map and transform structured data with the same ease that `sed`, `awk`, `grep` and friends let you play with text.

jq is written in portable C, and it has zero runtime dependencies. You can download a single binary, `scp` it to a far away machine of the same type, and expect it to work.

jq can mangle the data format that you have into the one that you want with very little effort, and the program to do so is often shorter and simpler than you'd expect.

Go read the [tutorial](#) for more, or the [manual](#) for way more.

Ask questions on [StackOverflow](#) using the `jq` tag, or on the `#jq` channel on [Freenode](#).



# tldr-pages

COLLABORATIVE CHEATSHEETS FOR CONSOLE COMMANDS

---

[build](#) passing [chat](#) on [gitter](#) [merged PRs](#) 3.7k [contributors](#) 399 [license](#) CC BY 4.0

## What is tldr-pages?

The tldr-pages project is a collection of community-maintained help pages for command-line tools, that aims to be a simpler, more approachable complement to traditional [man pages](#).



write down a command-line to see the help text that matches each argument

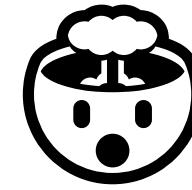
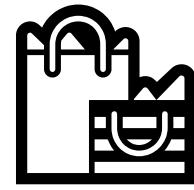
try [showthedocs](#) for explaining other languages

[EXPLAIN](#)

## examples

- :(){ :|:& };:
- for user in \$(cut -f1 -d: /etc/passwd); do crontab -u \$user -l 2>/dev/null; done
- file=\$(echo `basename "\$file"`)
- true && { echo success; } || { echo failed; }
- cut -d ' ' -f 1 /var/log/apache2/access\_logs | uniq -c | sort -n
- tar zcf - some-dir | ssh some-server "cd /; tar xvzf -"
- tar xzvf archive.tar.gz
- find . -type f -print0
- ssh -i keyfile -f -N -L 1234:www.google.com:80 host
- git log --graph --abbrev-commit --pretty=oneline origin..mybranch

# Inspiration



I know... you could get addicted 😊

[Code](#)[Issues 15](#)[Pull requests 15](#)[Actions](#)[Security](#)[Insights](#)

readme.md

[My open source work is supported by the community](#)[What is an awesome list?](#) [Contribution guide](#) [Creating a list](#) [Twitter](#) [Stickers & t-shirts](#)Follow the [Awesome Twitter account](#) for updates on new list additions.Just type [awesome.re](#) to go here. Check out my [blog](#) and follow me on [Twitter](#).

#

# awesome-list

An awesome list is a list of awesome things curated by the community. There are awesome lists about everything from [CLI applications](#) to [fantasy books](#). The [main repository](#) serves as a curated list of awesome lists.



Here are 3,140 public repositories matching this topic...

[Language: All ▾](#)[Sort: Best match ▾](#)

Find Open Source By Searching, | X +  
← → C 🔒 awesomeopensource.com



## Awesome Open Source

**Find Open Source By Searching, Browsing and Combining 7,000 Topics Across 59 Categories And 325,977 Projects**

Search, browse and combine topics

[Code](#)[Issues 5](#)[Pull requests 2](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)[Insights](#)[master](#)[2 branches](#)[0 tags](#)[Go to file](#)[Add file](#)[Code](#)

unixorn Merge pull request #975 from unixorn/2020-July-updates ...

✓ 65426ee 8 hours ago 2,814 commits

<a href="#">.circleci</a>	<a href="https://youtube-dl.org">https://youtube-dl.org</a> occasionally 503s, whitelist it from checks	2 months ago
<a href="#">.github</a>	Add framework option to PR template	8 days ago
<a href="#">zsh-plugin-assessor</a>	ZPA: Dash-compatibility; -q/--quiet options (no output except stderr)	2 years ago
<a href="#">.codeclimate.yml</a>	Disable markdownlint's ordered list item prefix style check	3 years ago
<a href="#">CODE_OF_CONDUCT.md</a>	Update Code of Conduct	last month
<a href="#">Contributing.md</a>	Update Contributions guideline and PR Template	12 months ago
<a href="#">LICENSE</a>	Initial commit	6 years ago
<a href="#">README.md</a>	Add Rust example	2 days ago
<a href="#">Writing_Plugins.md</a>	Add advise about fpath to Writing_Plugins.md	8 months ago

[README.md](#)

## awesome-zsh-plugins

[awesome](#) ←

### About

A collection of ZSH frameworks, plugins, themes and tutorials.

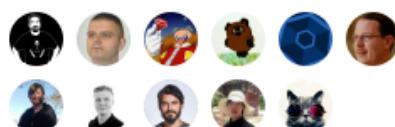
collection awesome-list zsh-plugins  
zsh-frameworks zsh-theme  
zsh-completions awesome list  
zsh-configuration zgen oh-my-zsh  
antigen zsh-plugin

[Readme](#)[BSD-3-Clause License](#)

### Releases

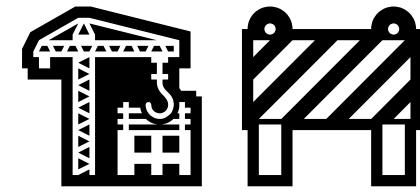
No releases published

### Contributors 122



+ 111 contributors

# Environment setup





Your unofficial guide to dotfiles on GitHub.

[Home](#)

[Tutorials](#)

[General-purpose utilities](#)

[Tool-specific frameworks](#)

[Bootstrap repositories](#)

[Inspiration](#)

[Tips and tricks](#)

[FAQ](#)



## GitHub ❤️ ~/

Why would I want my dotfiles on GitHub?

- *Backup, restore, and sync* the prefs and settings for your toolbox. Your dotfiles might be the most important files on your machine.
- *Learn* from the community. Discover new tools for your toolbox and new tricks for the ones you already use.
- *Share* what you've learned with the rest of us.

## Navigating this site

If you're just starting out, before you go symlinking everything in `~/\*`, you might want to check out some [tutorials](#) that discuss how you can organize your dotfiles. Next, you could look through the [general-purpose dotfiles utilities](#) to find a system you can use to manage your dotfiles. Some [tool-specific frameworks](#) could save you time in customizing specific programs like your shell.

Prefer to start from a popular base configuration? Check out dotfiles [bootstrap repositories](#). Or just looking for inspiration, to see how others are doing things? Check out [other GitHub users' dotfiles](#).



*Your unofficial guide to dotfiles on GitHub.*

[Home](#)

[Tutorials](#)

[General-purpose utilities](#)

[Tool-specific frameworks](#)

[Bootstrap repositories](#)

[Inspiration](#)

[Tips and tricks](#)

[FAQ](#)

## Bootstrap repositories

Rather than starting out with an empty repository, some people like to base their dotfiles on a preexisting base configuration. Below are some popular repositories that you can use as a starting point. If you're looking for other people's dotfiles to see how their setup works, check out the [inspiration](#) page.

- [thoughtbot/dotfiles](#) (6138 stars). A base set of configs, plugins, scripts, and aliases for vim, tmux, git, ruby, and zsh.
- [dotphiles](#) (594 stars). A community driven framework of dotfiles, for the usual terminal apps and shells, designed to work across multiple platforms and degrade for older versions of software or OS, allowing you to use the same settings on all your machines.

## README.md

# Dotbot



Dotbot makes installing your dotfiles as easy as `git clone $url && cd dotfiles && ./install`, even on a freshly installed system!

- [Rationale](#)
- [Getting Started](#)
- [Configuration](#)
- [Directives \(Link, Create, Shell, Clean, Defaults\)](#)
- [Plugins](#)
- [Command-line Arguments](#)
- [Wiki](#)

## Rationale

Dotbot is a tool that bootstraps your dotfiles (it's a [Dot]files [bo]o[t]strapper, get it?). It does *less* than you think, because version control systems do more than you think.

Demo



garodriguezlp / zsh-bootstrap

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags Go to file Add file Code

	garodriguezlp Add intial version of all scripts and tools	5d6354f 41 minutes ago 3 commits
bundles	Add intial version of all scripts and tools	41 minutes ago
demo	Add intial version of all scripts and tools	41 minutes ago
dotbot @ 1d4f434	Add dotbot	10 days ago
dotfiles	Add intial version of all scripts and tools	41 minutes ago
scripts	Add intial version of all scripts and tools	41 minutes ago
.gitmodules	Add dotbot	10 days ago
README.md	Add intial version of all scripts and tools	41 minutes ago
install	Add intial version of all scripts and tools	41 minutes ago
install.conf.yaml	Add intial version of all scripts and tools	41 minutes ago

README.md

## zsh-bootstrap

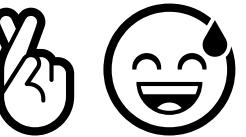
Powered by dotbot and made with ❤ for my friends @pls

### Installation

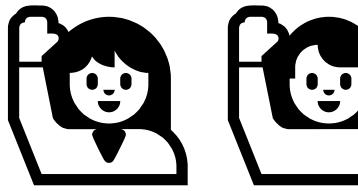
The following steps are the only ones required to start your Zsh journey

Try it: `git clone https://github.com/garodriguezlp/zsh-bootstrap && bash ./zsh-bootstrap/install`

# Demo time



Q&A



Once again, thanks!

