

Undefined Academy

Bootcamp de JavaScript Full-stack

Semana 1, Clase 1:

Markdown & Terminal



Metodología de trabajo

Formato de las sesiones



(1h 30m)

- 10 min Revisión de ejercicio anterior
- 50 min Presentación del tema
- 20 min Ejercicios prácticos
- 10 min Preguntas y discusión

Herramientas del Bootcamp

- **Twitch:** <https://undf.sh/en-vivo>
- **Discord:** <https://undf.sh/discord>
- **Notion:** <https://undf.sh/base>

Agenda

- **El mundo del Markdown**
 - Sintaxis básica de Markdown
 - Sintaxis saborizada en GitHub
- **Uso básico de la Terminal**
 - Conociendo la Terminal
 - Comandos alias para la Terminal

El mundo del Markdown

Markdown

Markdown es un **lenguaje de marcado ligero** que se utiliza para formatear texto de manera sencilla y fácil de leer.

Creado por John Gruber en 2004¹.

1. <https://daringfireball.net/projects/markdown/>

Sintaxis básica de Markdown

Sintaxis básica de Markdown¹

Un encabezado de primer nivel

=====

Un encabezado de segundo nivel

Ahora es el momento de mostrar un párrafo ordinario, pero la idea de los párrafos es que ocupen bastante texto, así que, aquí vamos.

Este es otro párrafo ordinario.

Encabezado 3

> Esta es una cita en bloque.

>

> Este es el segundo párrafo de la cita en bloque..

>

> ## Este es un H2 en una cita en bloque

1. <https://daringfireball.net/projects/markdown/basics>

Sintaxis básica de Markdown¹

Markdown

Markdown es un ****lenguaje de marcado ligero**** que se utiliza para formatear texto de manera **sencilla** y **fácil** de leer.

Ventajas

1. Fácil de usar
2. Portátil
3. Legible
4. Versátil
5. Compatible con HTML
6. Rápido
7. Centrado en el contenido

Filosofía

> Markdown is intended to be as easy-to-read and easy-to-write as is feasible.

1. <https://daringfireball.net/projects/markdown/syntax>



Twitch



Search




PINNED



Stream Schedule



 <https://glrz.me/stream>

 Mondays to Thursdays

🌞 SUMMER

* 3pm (GTM-5) 🇷🇺 🇩🇪 🇮🇹

* 5pm (GTM-3) 🇦🇷

* 10pm (GTM+2) 🇸🇪 🇪🇸

❄️ WINTER

* 3pm (GTM-6) 🇮🇹

* 4pm (GTM-5) 🇷🇺 🇮🇹

* 6pm (GTM-3) 🇦🇷

* 10pm (GTM+1) 🇸🇪 🇪🇸

Undefined Shell

EL LOBBY

reglas

presentate

elige-rol

EL CLUB

LA CAFETERÍA

EL AUDITORIO

anuncios

off-topic

preguntas

glrodasz recargar la con...

enlaces

trabajos

memes

el-podium

el-rincón

EL CAMPUS

anuncios-bootca...

informacion-...

• informacion-bootcamp

February 3, 2023

Guillermo Rodas

02/03/2023 9:05 PM

⚠️

****IMPORTANTE****: para ver el resto de canales del bootcamp ve a **#elige-rol** y dale click al emoji 🗨️

📅

****¿Cuándo empieza?****

* El 1 de Marzo del 2023: <https://undf.sh/1ro-marzo>

💻

****¿En dónde veo las clases?****

* En Twitch, síguenos: <https://undf.sh/en-vivo>

📺

****¿Quedará grabado?****

* Sí, la grabación queda disponible en Twitch de inmediato. En YouTube la puedes ver acá: <https://undf.sh/en-diferido>

💻

****¿De qué son las clases?****

* Aquí puedes ver los contenidos: <https://undf.sh/contenidos>

🕒

****Horario****

* Lunes a Jueves a las 9 PM (GTM-5) 🥑 ****Latam****

escape to [cancel](#) • enter to [save](#)

🚀 54

❤️ 46

🇵🇪 21

🇦🇷 61

🎵 6

🇨🇷 3

🇨🇱 10

🇪🇨 41

🇲🇪 22

🇺🇷 4

🇪🇸 20

🇨🇴 15

🇺🇾 2

🇨🇱 3

🇨🇷 1

🇨🇱 1

🇪🇸 1

February 20, 2023

Guillermo Rodas

02/20/2023 2:48 PM

📖

Guía Pre-work <https://undf.sh/prework> (edited)

❤️ 76

👍 37

🗨️ 16

👏 11

🇨🇷 4

🇲🇪 6

🇦🇷 7

🇪🇨 5

🇺🇾 1

TODAY

🔒 Messages to yourself are end-to-end encrypted. No one outside of this chat, not even WhatsApp, can read or listen to them. [Click to learn more.](#)



Hola, _¿cómo estoy?_ qué ~~~extraño~~~
terapéutico es hablar con uno mismo. 🤪



TODAY

🔒 Messages to yourself are end-to-end encrypted. No one outside of this chat, not even WhatsApp, can read or listen to them. [Click to learn more.](#)

Hola, ¿cómo estoy? qué ~~extraño~~ terapéutico es hablar con uno mismo. 😬

15:02 ✓✓



| Type a message





glrodasz on Oct 4, 2022 • edited ▼

Owner



Background

We are having an issue when building `storybook-tools-helpers` package. This package has as dependency the `storybook-tools-styles` package, when we run the build is not transpiling the dependencies.

How to reproduce

1. Run the command

```
npx nx build storybook-tools-helpers --skip-nx-cache
```

2. Go to `dist/packages/storybook-tools-helpers`

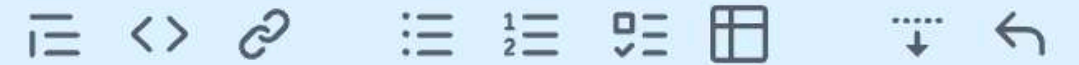
```
import { jsx } from 'react/jsx-runtime';
import { getClasses } from '@glrodasz/storybook-tools-styles';

var commonjsGlobal = typeof globalThis !== 'undefined' ? globalThis : typeof window !== 'undefined' ? window :

var check = function (it) {
  return it && it.Math == Math && it;
```


Write

Preview



Background

We are having an issue when building `storybook-tools-helpers` package. This package has as dependency the `storybook-tools-styles` package, when we run the build is not transpiling the dependencies.

How to reproduce

1. Run the command

...

```
npx nx build storybook-tools-helpers --skip-nx-cache
```

...

2. Go to `dist/packages/storybook-tools-helpers`

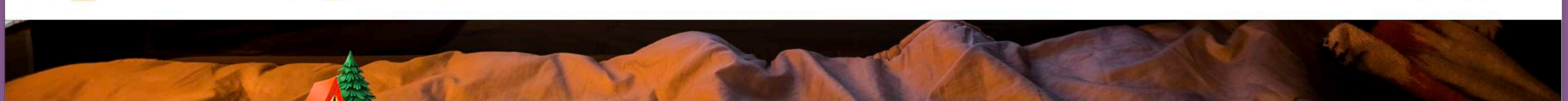
```
```js
```

```
import { jsx } from 'react/jsx-runtime';
```

```
import { getClasses } from '@glrodasz/storybook-tools-styles';
```

```
var commonjsGlobal = typeof globalThis !== 'undefined' ? globalThis : typeof window !== 'undefined' ? window : typeof global !== 'undefined' ? global : typeof self !== 'undefined' ? self : {};
```

```
var check = function (it) {
 return it && it.Math == Math && it;
```



# Knowledgebase

*Esta es la página de la **base de conocimientos**, aquí agregaremos todos los links y guías importantes. También incluiremos los enlaces de las clases con su respectiva grabación y diapositivas.*

## ¿Cómo acceder?

Puedes ver las clases [En vivo](#) o [En diferido](#).

[Ver clases en vivo](#)

**Horarios**

8pm GTM-6 (Lunes a Jueves) 🇲🇪

[Ver clases en diferido](#)

🇲🇪, 🇪🇨, 🇵🇪, 🇦🇷, 🇨🇱, 🇪🇸, 🇸🇪 aproximadamente una hora después de la transmisión queda

# Sintaxis sabORIZADA en GitHub





glrodasz on Oct 4, 2022 • edited ▼

Owner



## Background

We are having an issue when building `storybook-tools-helpers` package. This package has as dependency the `storybook-tools-styles` package, when we run the build is not transpiling the dependencies.

## How to reproduce

1. Run the command

```
npx nx build storybook-tools-helpers --skip-nx-cache
```

2. Go to `dist/packages/storybook-tools-helpers`

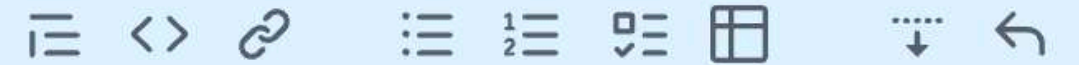
```
import { jsx } from 'react/jsx-runtime';
import { getClasses } from '@glrodasz/storybook-tools-styles';

var commonjsGlobal = typeof globalThis !== 'undefined' ? globalThis : typeof window !== 'undefined' ? window :

var check = function (it) {
 return it && it.Math == Math && it;
```

Write

Preview



## ## Background

We are having an issue when building `storybook-tools-helpers` package. This package has as dependency the `storybook-tools-styles` package, when we run the build is not transpiling the dependencies.

## ## How to reproduce

1. Run the command

...

```
npx nx build storybook-tools-helpers --skip-nx-cache
```

...

2. Go to `dist/packages/storybook-tools-helpers`

```
```js
```

```
import { jsx } from 'react/jsx-runtime';
```

```
import { getClasses } from '@glrodasz/storybook-tools-styles';
```

```
var commonjsGlobal = typeof globalThis !== 'undefined' ? globalThis : typeof window !== 'undefined' ? window : typeof global !== 'undefined' ? global : typeof self !== 'undefined' ? self : {};
```

```
var check = function (it) {  
  return it && it.Math == Math && it;
```

Sintaxis saborizada en GitHub¹

Tablas²

```
| Syntax      | Description |
| -----    | -----    |
| Header     | Title      |
| Paragraph   | Text       |
```

```
| Syntax | Description |
| --- | ----- |
| Header | Title |
| Paragraph | Text |
```

1. <https://github.github.com/gfm/>
2. <https://github.github.com/gfm/#tables-extension->

Sintaxis saborizada en GitHub¹

Listas de tareas²

- [x] Learn about Markdown
- [] Learn about the Terminal
- [] Learn about `alias` commands
- [] Graduate 🎉

1. <https://github.github.com/gfm/>

2. <https://github.github.com/gfm/#task-list-items-extension->

Extended Syntax¹

- Strikethrough (Tachado)
- Highlight (Subrayado)
- Footnotes (Notas al pie de página)
- Fenced Code Blocks (Bloques de código)
- Emojis 🏕️ `:tent:`

1. <https://www.markdownguide.org/extended-syntax/#strikethrough>

¿De verdad estamos
viendo en un Bootcamp
Full-stack?...

¿¡Markdown!?



Team 🔥 vs Team 🤡

THE

MARKDOWN GUIDE

MATT CONE



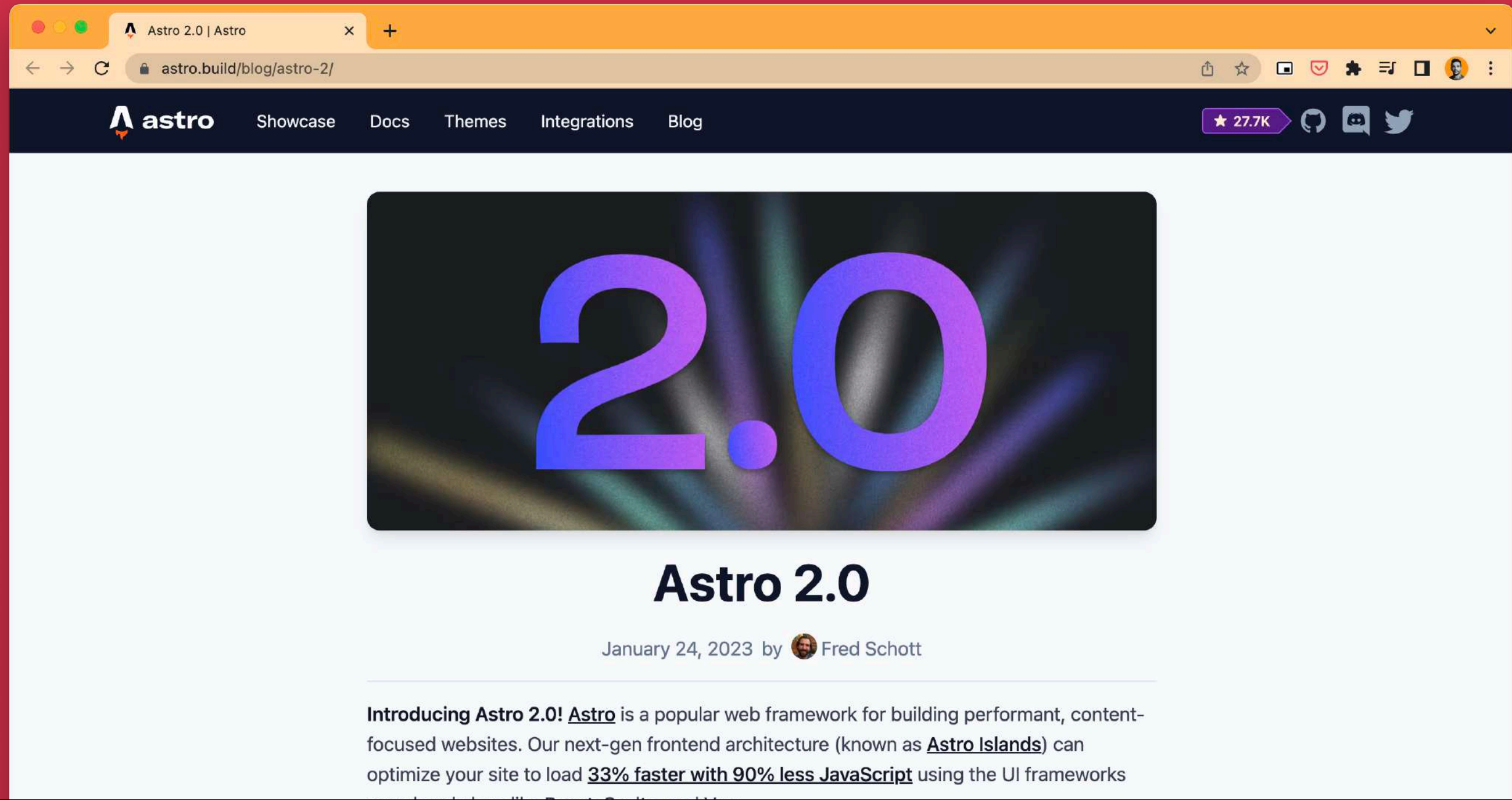
The Markdown Guide¹

By Matt Cone

Aprende Markdown en 60 páginas. El libro "La Guía de Markdown" incluye todo lo que necesitas para comenzar y dominar la sintaxis de Markdown.

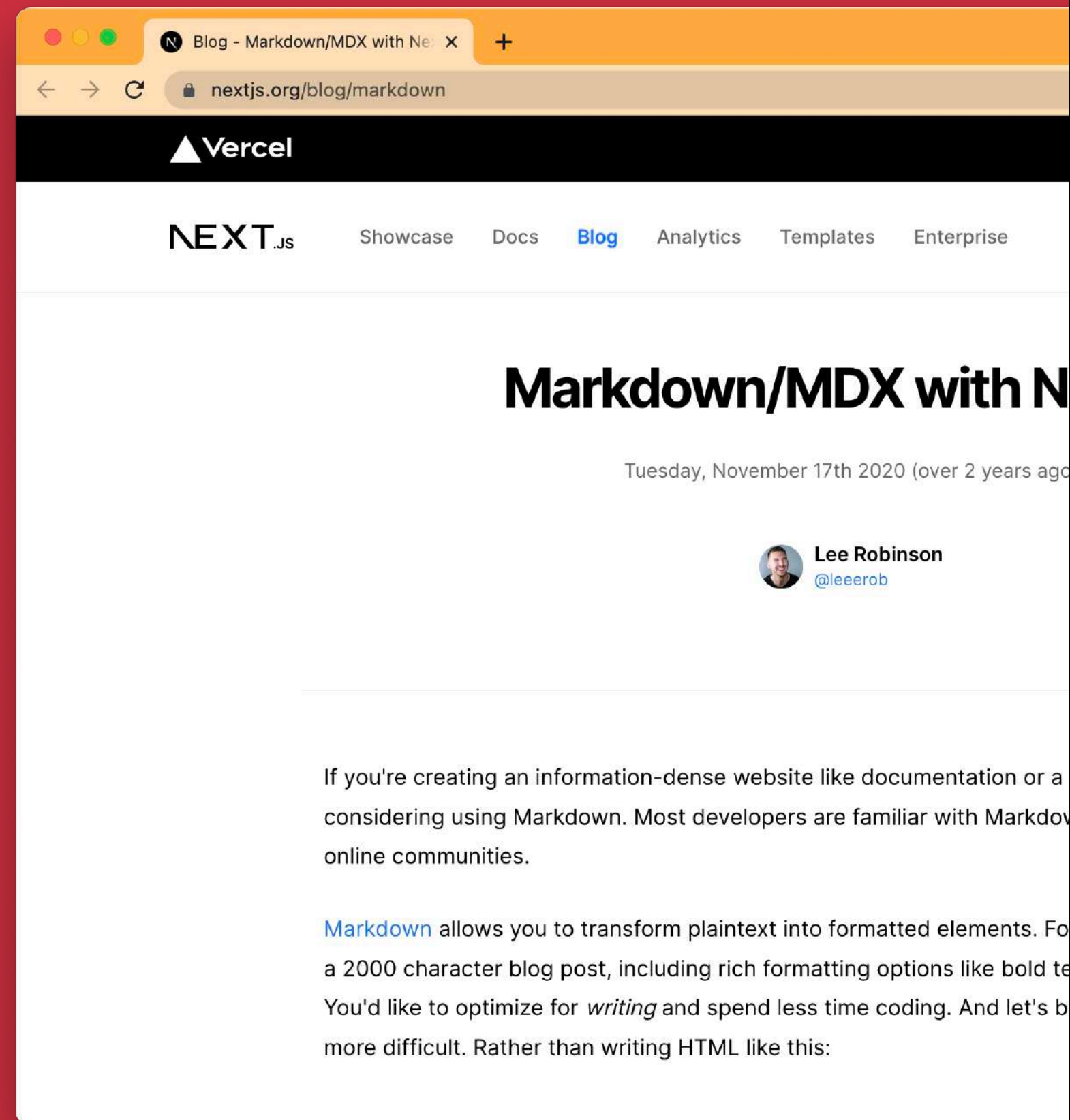
1. <https://www.markdownguide.org/book>

"Astro 2.0 is the first major web framework to deliver complete type-safety for Markdown and MDX."

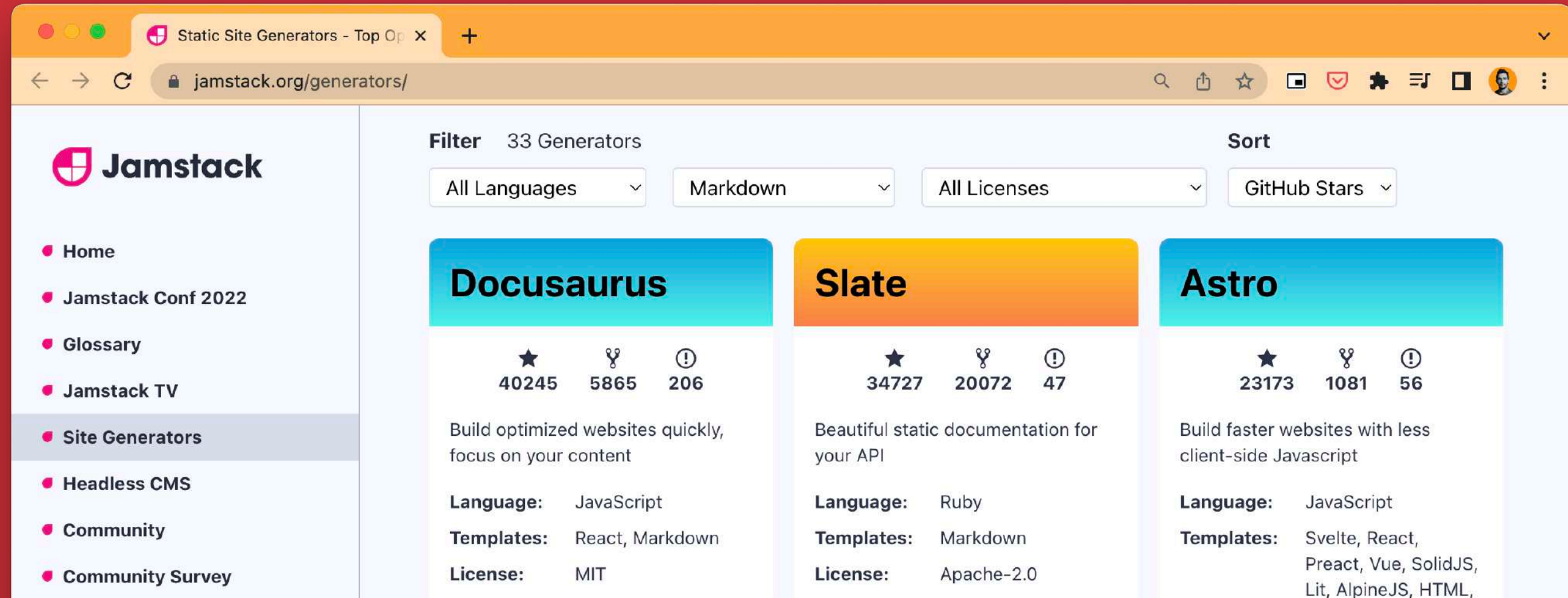


"If you're creating an information-dense website like documentation or a blog, you're probably considering using Markdown.

Most developers are familiar with Markdown from GitHub and other online communities."



Site Generators 12



1. <https://jamstack.org/generators/>
2. 33~ generadores de sitios estáticos usan Markdown



1

Undefined Academy
Bootcamp de JavaScript Full-stack

2

Semana 1, Clase 1:
Markdown & Terminal

3

Metodología de trabajo

4

Formato de las sesiones
(1h 30m)
- 10 min Revisión de ejercicio anterior
- 50 min Presentación del tema
- 20 min Ejercicios prácticos
- 10 min Preguntas y discusión

5

Herramientas del Bootcamp
- Twitch: <https://undf.sh/en-vivo>
- Discord: <https://undf.sh/discord>
- Notion: <https://undf.sh/base>

6

Agenda
- El mundo del Markdown
- Sintaxis básica de Markdown
- Sintaxis avanzada en GitHub
- Uso básico de la Terminal
- Conociendo la terminal
- Comandos útiles para la terminal

7

El mundo del Markdown

Bootcamp — Semana 1, Clase 1

TextVisualsDesign

1

Undefined Academy
Bootcamp de JavaScript Full-stack

2

Semana 1, Clase 1:
Markdown & Terminal

3

🏆 Metodología de trabajo

4

→ ## Formato de las sesiones 🕒 (1h 30m)
→ * `10 min` Revisión de ejercicio anterior
→ * `50 min` Presentación del tema
→ * `20 min` Ejercicios prácticos
→ * `10 min` Preguntas y discusión

5

Herramientas del Bootcamp 🔧
→ * ****Twitch****: [<https://undf.sh/en-vivo>] (<https://undf.sh/en-vivo>)
→ * ****Discord****: [<https://undf.sh/discord>] (<https://undf.sh/discord>)
→ * ****Notion****: [<https://undf.sh/base>] (<https://undf.sh/base>)

6

Agenda
→ * ****El mundo del Markdown****

00:05:18

Speech

Spoken Text

Just Type

1. Ordered List

⇧ ⌘ L

- Unordered List

⌘ L

- [] Task List

⇧ ⌘ L

> Blockquote

⌘ >

Text on Slide

Slide Break

⇧ ⌘ -

Body Text

→

Title

⌘ 1

Subtitle

⌘ 2

Heading

⌘ 3

Subheading

⌘ 4

</path/image.jpg>

</path/video.mp4>

Format

****Bold****

⌘ B

Italic

⌘ I

// Comment

⌘ /

~~Strikethrough~~

⇧ ⌘ U

==Highlight==

⇧ ⌘ U

[Footnote]:

⌘ K

[Link](http://...)

⌘ K

Recursos

- <https://daringfireball.net/projects/markdown/syntax>
- <https://commonmark.org/>
- <https://github.github.com/gfm/>
- <https://www.markdownguide.org/>
- <https://www.markdownguide.org/book>

Uso básico de la Terminal

Conociendo la Terminal

La Terminal




También conocida como **consola** o **línea de comandos** es una herramienta que nos permite interactuar con el sistema operativo mediante comandos de texto.

Ventajas de aprender a manejar la Terminal

1. Entender las bases de cómo funciona tu SO.
2. Realizar tareas con mayor velocidad.
3. Moverse con fluidez en un servidor.

¿Terminal, Shell, Prompt?




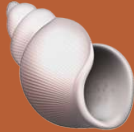

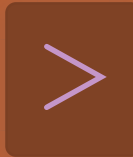

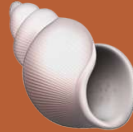






-  **Terminal** : es un programa que proporciona una **interfaz de línea de comandos**¹ para interactuar con el sistema operativo
-  **Shell**: es un programa que interpreta los comandos escritos en la Terminal y los ejecuta en el sistema operativo.
-  **Prompt**: es el mensaje que aparece en la línea de comandos de la Terminal y que indica que está listo para recibir un comando.

1. CLI: Command-line interface

Interfaz

Una interfaz es el punto de conexión o comunicación entre dos o más sistemas, dispositivos o elementos que necesitan interactuar entre sí para lograr un objetivo común.

Diferencias en los sistemas operativos¹

- **Windows:**  PowerShell,  PowerShell,  
- **Linux:**  Gnome Terminal,  Bash,  
- **macOS:**  Terminal,  Z Shell,  

1.  Terminal,  Shell,  Prompt

UNIX-like

Tanto **Linux** cómo **macOS** son sistemas operativos basados en UNIX, por lo tanto sus **Shells** son *prácticamente* iguales.

macOS Terminals

Warp

iTerm

```
..mos-MacBook-Pro:~/Code
~/Code (0.977s)
neofetch

      'c.
      ,xNMM.
      .OMMMMo
      OMMMO,
      .;loddo:' loolloddol;.
      cKMMMMMMMMMMNWMMMMMMMMMM0:
      .KMMMMMMMMMMMMMMMMMMMMMMWd.
      XMMMMMMMMMMMMMMMMMMMMMMX.
      ;MMMMMMMMMMMMMMMMMMMMMM:
      :MMMMMMMMMMMMMMMMMMMMMM:
      .MMMMMMMMMMMMMMMMMMMMMMX.
      kMMMMMMMMMMMMMMMMMMMMMMWd.
      .XMMMMMMMMMMMMMMMMMMMMMMk

glrodasz@Guillermos-MacBook-Pro
-----
OS: macOS 13.2.1 22D68 x86_64
Host: MacBookPro16,1
Kernel: 22.3.0
Uptime: 6 days, 1 hour, 41 mins
Packages: 117 (brew)
Shell: zsh 5.8.1
Resolution: 1792x1120@2x
DE: Aqua
WM: Quartz Compositor
WM Theme: Blue (Dark)
Terminal: WarpTerminal
CPU: Intel i9-9880H (16) @ 2.30GHz
GPU: Intel UHD Graphics 630, AMD Radeon Pro 5500 XT 8GB
Memory: 21486MiB / 32768MiB

~/Code
|
```

```
~/_Code
      'c.
      ,xNMM.
      .OMMMMo
      OMMMO,
      .;loddo:' loolloddol;.
      cKMMMMMMMMMMNWMMMMMMMMMM0:
      .KMMMMMMMMMMMMMMMMMMMMMMWd.
      XMMMMMMMMMMMMMMMMMMMMMMX.
      ;MMMMMMMMMMMMMMMMMMMMMM:
      :MMMMMMMMMMMMMMMMMMMMMM:
      .MMMMMMMMMMMMMMMMMMMMMMX.
      kMMMMMMMMMMMMMMMMMMMMMMWd.
      .XMMMMMMMMMMMMMMMMMMMMMMk
      kMMMMMMMMMMMMMMMMMMMMMMd
      ;KMMMMMMMWXXWMMMMMMMMMk.
      .cooc,. .,coo:.

glrodasz@Guillermos-MacBook-Pro.local
-----
OS: macOS 13.2.1 22D68 x86_64
Host: MacBookPro16,1
Kernel: 22.3.0
Uptime: 6 days, 1 hour, 41 mins
Packages: 117 (brew)
Shell: zsh 5.8.1
Resolution: 1792x1120@2x
DE: Aqua
WM: Quartz Compositor
WM Theme: Blue (Dark)
Terminal: iTerm2
Terminal Font: Hack-Regular 18
CPU: Intel i9-9880H (16) @ 2.30GHz
GPU: Intel UHD Graphics 630, AMD Radeon Pro 55
Memory: 21486MiB / 32768MiB

~/Code
> |
```

Windows Terminals

- PowerShell¹
- **Windows Terminal**

1. **PowerShell** esta disponible para todos los sistemas operativos.

Linux Terminals

- **GNOME Terminal**
- Alacritty¹

1. **Alacrity** está para todos los sistemas operativos y además es un Terminal Emulator, no un Terminal Application.

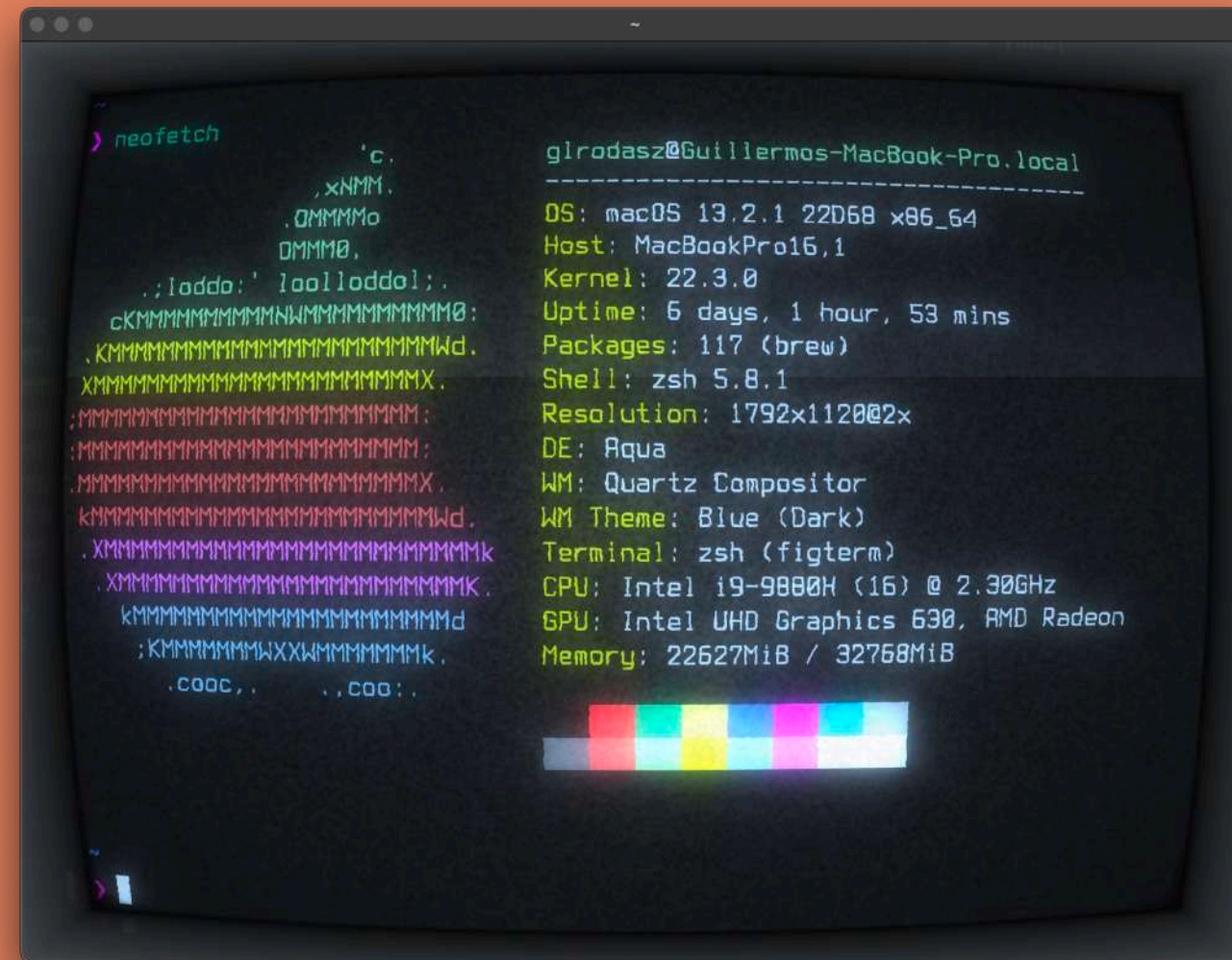
 *Buscar la diferencia entre "Terminal Application" y "Terminal Emulator"*

Hyper¹

Es una Terminal de código abierto basada en HTML5 disponible para todos los sistemas operativos.

1. <https://hyper.is/>

Cool Retro Terminal¹₂



1. <https://github.com/Swordfish90/cool-retro-term>
2. Solo para Linux y macOS.

Ubicación en la Terminal

Ubicación en la Terminal

En sistemas tipo UNIX¹ la ruta que vemos cuando abrimos la Terminal es el **HOME** del usuario.

Lo podemos comprobar con el comando `pwd`²

que nos mostrará algo como

```
/Users/<usuario>.
```

1. Linux y macOS.
2. Print current working directory.

Comandos básicos del sistema operativo

- **date**: Muestra la fecha y hora del sistema actual.
- **uptime**: Muestra el tiempo transcurrido desde que inició el equipo por última vez.
- **cal**: Muestra un calendario del mes actual.
- **df**: Muestra el espacio libre actual en el disco duro.
- **whoami**: Muestra el nombre del usuario actual.

Comandos para el manejo de carpetas

- **ls:**¹ Lista los archivos de la carpeta actual, o de la ruta dada. Ej. `ls Documents`.
- **cd:**² Nos permite navegar entre carpetas. Ej. `cd Documents` nos ubicaría en `/Users/<usuario>/Documents`.
- **pwd:**³ Nos muestra la ruta absoluta del directorio actual, así podemos saber en dónde estamos ubicados.

1. List directory contents.
2. Change current directory.
3. Print current working directory.

Comandos para el manejo de carpetas

- **mkdir**:¹ Crea una nueva carpeta dado un nombre para esa carpeta. Ej. `mkdir hola`.
- **cp -r**:² Nos permite copiar una carpeta dada una ruta inicial y una ruta final. Ej. `cp -r hola alo`.
- **rm -r**:³ Nos permite eliminar una carpeta dada una ruta. Ej. `rm -r hola`.

1. Make directory.
2. Copy.
3. Remove.

La opción `-r` significa que hará el proceso de manera recursiva.

Comandos para el manejo de archivos

- **touch:** Crea un archivo nuevo dada una ruta y/o nombre. Ej. `touch chao`.
- **cat:**¹ Imprime el contenido de un archivo dada su ruta. Ej. `cat chao`.
- **more:** Es muy similar a cat solo que nos permite navegar usando controles **tipo Vim**.

1. Concatenate and print files.

¿Cómo sale un usuario de Vim?
—Reiniciando la PC. 🤡

Comandos para el manejo de archivos

- **cp**:¹ Es el comando que nos permite copiar archivos dada una ruta inicial y una ruta final.
- **rm**: Nos permite eliminar un archivo dada una ruta.
- **mv**:² Nos permite mover un archivo o carpeta dada una ruta inicial y una ruta final.

Ej. `mv adios /Users/<usuario>/Downloads`

mv también es el comando que nos permite cambiar el nombre de un archivo o carpeta. Ej. `mv adios bye`.

1. Copiar
2. Mover.

 ¿Por qué `mv` (mover) se usa para renombrar archivos o carpetas?

Rutas relativas y absolutas

Las rutas se pueden especificar de manera relativa o absoluta.

Rutas relativas y absolutas

Cuando especificamos una ruta absoluta, debemos introducir toda la ruta, es decir, desde la raíz (slash). Ej. `cd`

`/Users/<usuario>/Documents`.

Mientras que las rutas relativas consideran el directorio actual del usuario como el punto de partida, si estuviésemos en el **HOME**. Ej. `cd Documents`.

Atajos de los comandos en la terminal

- **Ctrl + c:** Si queremos cancelar el comando actual o la terminal no responde podemos hacer uso de este atajo para detenerlo.
- **Ctrl + u:** Nos permite borrar la línea actual que estemos escribiendo en la terminal.
- **Ctrl + w:** Nos permite eliminar la última palabra de los comandos que estamos escribiendo en el momento.
- **Ctrl + a:** Nos permite movernos al inicio de la línea de comandos.
- **Ctrl + e:** Nos permite movernos al final de la línea de comandos.

Atajos de los comandos en la terminal

- **Flecha arriba:** Nos muestra el comando anterior de nuestro historial de comandos usados.
- **Flecha abajo:** Nos muestra el comando siguiente de nuestro historial de comandos usados.
- **Alt + b:**¹ Nos permite movernos entre las palabras de la línea de comandos actuales hacia atrás.
- **Alt + f:**² Nos permite movernos entre las palabras de la línea de comandos actuales hacia adelante.
- **Ctrl + r:** Nos permite hacer una búsqueda de comandos en el historial. Si encontramos un resultado que no es el deseado, podemos seguir presionando Ctrl + r para rotar la búsqueda.

1. En  macOS es Alt + flecha izquierda.

2. En  macOS es Alt + flecha derecha.

Comandos alias para la terminal

Comandos alias para la terminal

Un alias es una abreviatura o nombre alternativo que se utiliza para referirse a un comando en el sistema.

Los alias son útiles para simplificar y acelerar la entrada de comandos largos y complejos.

Comandos alias para la terminal

Supongamos que por alguna razón queremos filtrar las carpetas de nuestro directorio de usuario dada una palabra. Esto se podría hacer haciendo uso del comando **ls** y **grep**.

```
ls ~/ | grep <palabra>
```

Comandos alias para la terminal

Podríamos hacer un alias del comando anterior y este recibiría de manera dinámica el parámetro que le indiquemos.

```
alias findd="ls ~/ | grep $1"
```

Comandos alias para la terminal

A partir de la definición alias, podemos seguir haciendo uso del comando `findd`

```
findd Do
```

**Guardar los
comandos alias
de manera
permanente**


Guardar los comandos alias de manera permanente

Para guardarlos de manera permanente en **bash** debe hacerse uso del archivo

`.bashrc` mientras que para ~ **Shell** debe hacerse uso del archivo `.zshrc`.

 Para saber que tipo de Shell tienes, ejecuta `echo $SHELL`

- Sí tu resultado es `/bin/zsh` quiere decir que tienes **"Z Shell"**
- Sí tu resultado es `/bin/bash` quiere decir que tienes **"Bash"**

 **Si vas a editar tus archivos de configuración por primera vez, lo mejor es que uses VS Code.**

Según tu tipo de terminal puedes ejecutar:

```
code ~/.bashrc o code ~/.zshrc.
```

Sí no existen dichos archivos crea un nuevo, **VS Code** te ayudara creando dicho archivo en blanco solo debes guardar.

Comandos alias para la terminal

Ahora solo tienes que pegar tu alias en una sola linea dentro de tu archivo `~/ .bashrc` o `~/ .zshrc`.

```
# `~/ .bashrc` o `~/ .zshrc`  
alias findd="ls ~/ | grep $1"
```

**Comandos alias
recomendados**

Comandos alias recomendados

Los comandos start y dev se suelen usar mucho

```
alias npmd="npm run dev"  
alias npms="npm start"
```

Para evitar eliminar permanente los archivos¹

```
alias rm="trash"
```

El comando de clear se suele usar mucho

```
alias cl="clear"
```

Cuando tienes un día muy agitado, busca café.

```
alias cafe="cat /dev/urandom | hexdump | grep \"ca fe\""
```

1. <https://github.com/sindresorhus/trash#cli>

Recursos

- <https://git-tower.com/blog/command-line-cheat-sheet/>
- <https://www.shortcutfoo.com/app/dojos/command-line>
- <https://cheatography.com/davechild/cheat-sheets/linux-command-line/>

Practica

1. Vamos a crear una pequeña CV en Dillinger¹
2. Luego vamos a copiar dicho contenido y crearemos un nuevo archivo Markdown en nuestro **HOME**.
3. Finalmente haremos uso de `11ty` para generar el HTML de dicho Markdown.

1. <https://dillinger.io/>

Ejercicio

Crear un Github Profile README¹

1. Este debe tener una pequeña descripción tuya tipo CV.
2. Añade una tabla en Markdown (Obviamente) con al menos 5 comandos de la Terminal que hayas aprendido, de forma que tenga nombre del comando y descripción en columnas separadas.
3. Añade una sección con al menos 2 comandos alias que te parezcan útiles incluyendo columna de descripción.
4. (Opcional) Clona tu repositorio y ejecuta `llty` como en la practica.

1. <https://docs.github.com/en/account-and-profile/setting-up-and-managing-your-github-profile/customizing-your-profile/managing-your-profile-readme>

¿Preguntas?