

Git and Github quick guide training

Module one

Part 1

- Installation of Windows Terminal (Powershell)
- Installation of Git for Windows
- Customization of Powershell and Git Bash (Born again shell) using oh-my-posh

Install Git-for-Windows



The screenshot shows the Git website homepage. At the top left is the Git logo (a red diamond with a white branching diagram) followed by the text "git --everything-is-local". To the right is a search bar with a magnifying glass icon and the text "Search entire site...". On the left side, there is a navigation menu with links: "About", "Documentation", "Downloads" (highlighted in red), "GUI Clients", "Logos", and "Community". Below the "Downloads" link, there is a section titled "The entire Pro Git book" with the text "written by Scott Chacon and Ben Straub is available to read". The main content area is titled "Downloads" and features three operating system options: "macOS" (with an Apple logo), "Windows" (with a Windows logo), and "Linux/Unix" (with a Tux logo). Below these options, it says "Older releases are available and the Git source repository is on GitHub." On the right side, there is a large monitor displaying the "Latest source Release 2.41.0" with a link to "Release Notes (2023-06-01)" and a prominent button that says "Download for Windows".

git --everything-is-local

Search entire site...

About
Documentation
Downloads
GUI Clients
Logos
Community

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to read

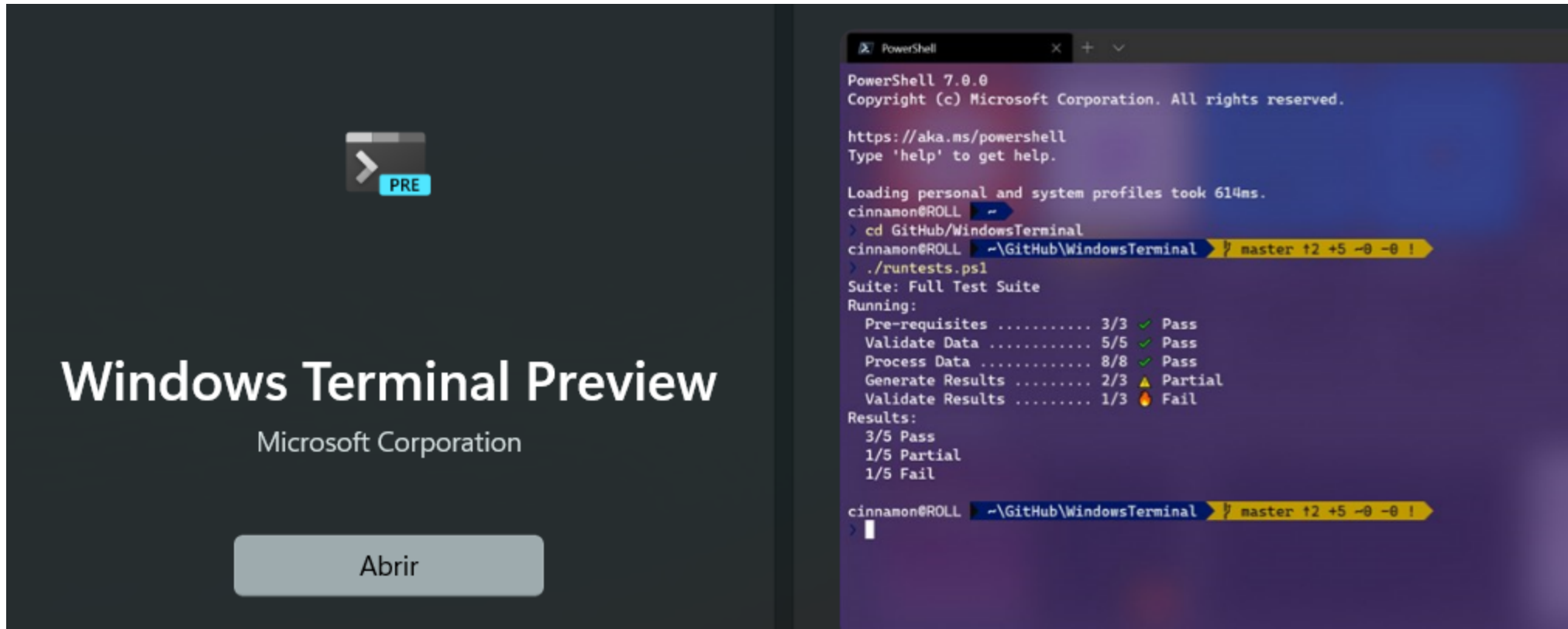
Downloads

macOS Windows Linux/Unix

Older releases are available and the Git source repository is on GitHub.

Latest source Release
2.41.0
[Release Notes \(2023-06-01\)](#)
Download for Windows

Install Windows Terminal



Install Powershell

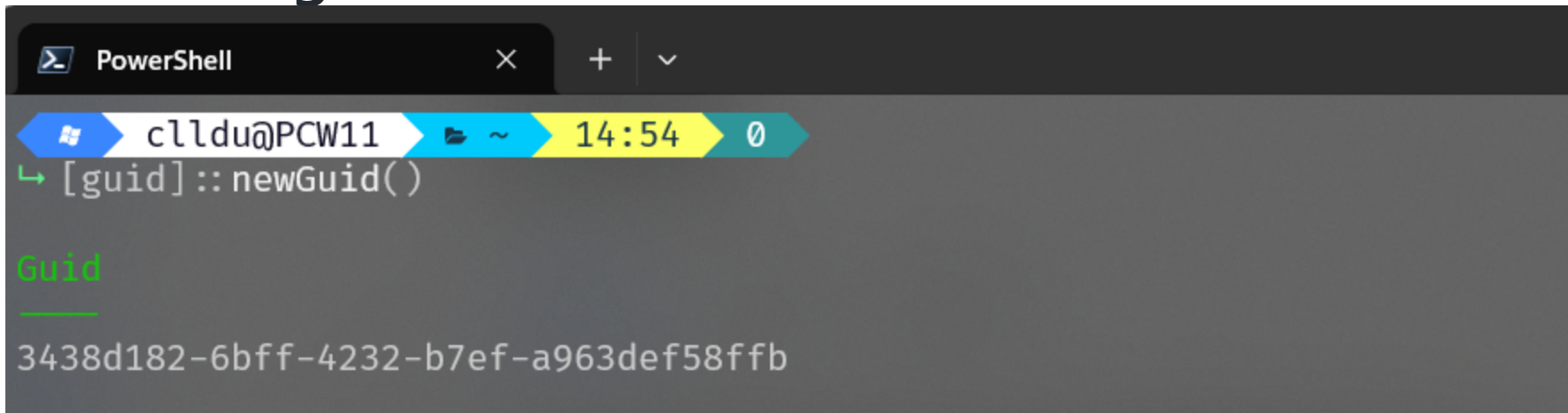
<https://github.com/PowerShell/powershell/releases>

v7.3.6 Release of PowerShell

PowerShell-7.3.6-win-x64.msi

758E130DC13708A72CEE29DB0B8EFD987376A4A4A22114C3B00D50CF7295D35E

Create a 'guid'



A screenshot of a PowerShell terminal window. The title bar shows 'PowerShell' with standard window controls. The terminal prompt is 'c\l\du@PCW11 ~ 14:54 0'. The command '[guid]::newGuid()' has been entered and executed. The output is a GUID: '3438d182-6bff-4232-b7ef-a963def58ffb'. The word 'Guid' is printed in green above the GUID.

```
PowerShell
c\l\du@PCW11 ~ 14:54 0
[guid]::newGuid()

Guid
-----
3438d182-6bff-4232-b7ef-a963def58ffb
```

Abrir o arquivo json

```
    "face": "Hack Nerd Font",  
  },  
  "guid": "{5ba085dd-7153-4102-804b-b7d253925678}",  
  "hidden": false,  
  "icon": "%PROGRAMFILES%\\git\\mingw64\\share\\git\\git-for-windows.ico",  
  "name": "Git Bash",  
  "opacity": 56,  
  "startingDirectory": "C:/Users/clldu",  
  "useAcrylic": true
```

Install Oh-my-Posh

Oh My Posh







A prompt theme engine for any shell.

Get Started

A preview of the Oh My Posh prompt theme. It features a dark blue background. The prompt is composed of several colored segments: a yellow segment with 'jan', an orange segment with a folder icon and 'oh-my-posh', a green segment with a git icon and 'main', and a blue segment with a checkmark icon. To the right of these segments is the text 'hello world!' followed by a cursor. Further to the right, the text 'in zsh at 13:50:27' is displayed in a light blue color.

jan oh-my-posh main ✓ hello world! in zsh at 13:50:27

Winget to install Oh-my-posh

-  Get started
-  Installation
-  Windows
-  macOS
-  Linux
-  Fonts

winget scoop manual

Open a PowerShell prompt and run the following command:

```
winget install JanDeDobbeleer.OhMyPosh -s winget
```


Install Nerd Fonts



Create a '.bashrc' file



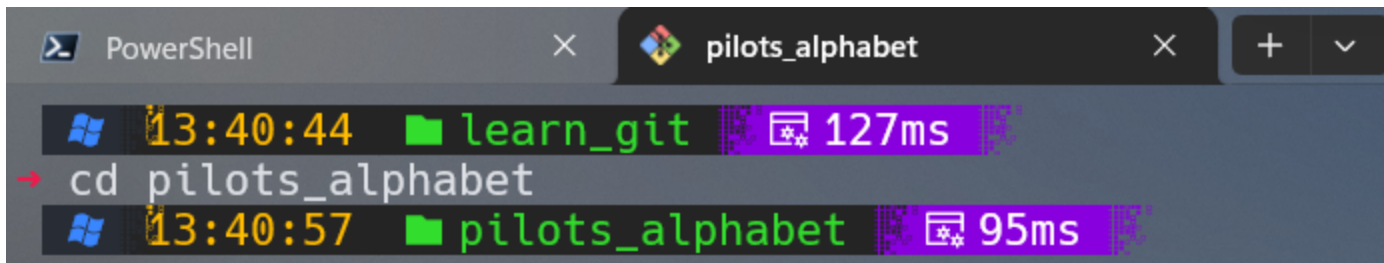
Open ".bashrc"

```
eval "$ (oh-my-posh init bash)"
```

```
eval "$ (oh-my-posh init bash --config "C:/Users/yourname/craver.omp.json")"
```

```
C:/Users/yourname/appData/Local/Programs/oh-my-posh/themes/craver.omp.json
```

Git bash terminal



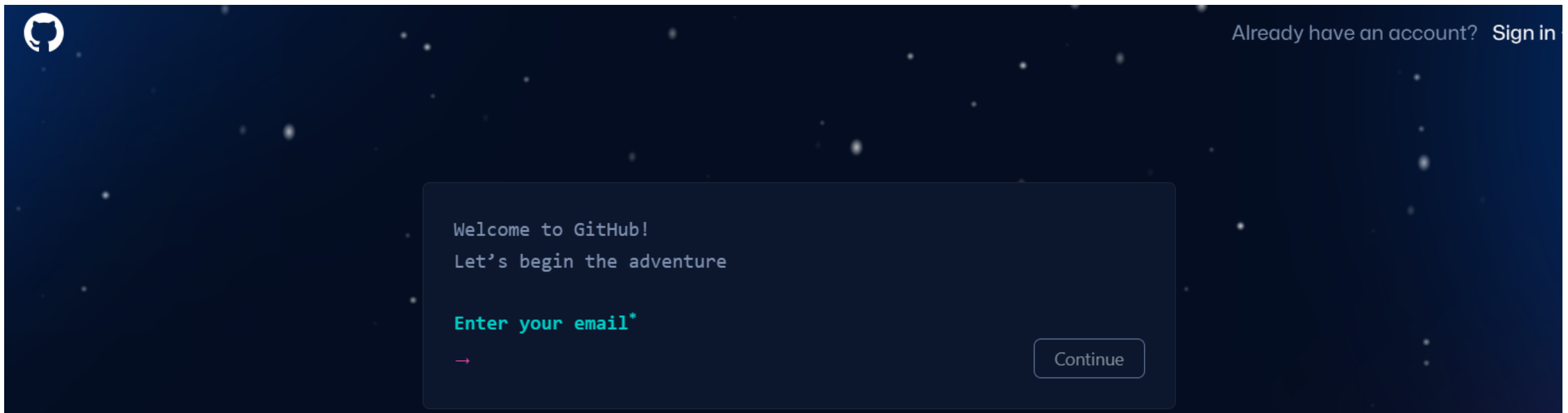
The screenshot shows a Windows PowerShell terminal window with two tabs: 'PowerShell' and 'pilots_alphabet'. The 'pilots_alphabet' tab is active. The terminal displays two command-line prompts. The first prompt is at 13:40:44, showing the current directory as 'learn_git' and a system icon with '127ms'. The second prompt is at 13:40:57, showing the current directory as 'pilots_alphabet' and a system icon with '95ms'. A red arrow points to the 'cd pilots_alphabet' command.


```
PowerShell × pilots_alphabet × + v  
13:40:44 learn_git 127ms  
→ cd pilots_alphabet  
13:40:57 pilots_alphabet 95ms
```

Part 2

- *Setting up a Github account*
- *Git configuration*
- *Connecting Git to Github*

Sign up at Github

A screenshot of the GitHub sign-up page. The background is dark blue with a starry space pattern. In the top left corner is the GitHub logo. In the top right corner, the text "Already have an account? Sign in" is visible. In the center, there is a dark blue rounded rectangle containing the following text: "Welcome to GitHub!" followed by "Let's begin the adventure" on the next line. Below this, the text "Enter your email*" is displayed in a light blue color. Underneath the email prompt is a small red arrow pointing to the right. To the right of the email input area is a rounded rectangular button with the text "Continue".

 Already have an account? [Sign in](#)

Welcome to GitHub!
Let's begin the adventure

Enter your email*

→

Continue

Git configuration

1. Set your identity:

```
git config --global user.name "Your Name"  
git config --global user.email "your.email@example.com"
```

2. Handle end-of-line character differences:

```
git config --global core.autocrlf true
```

3. Prevent conversion warning messages:

```
git config --global core.safecrlf false
```

4. Set Notepad as the default editor (replace "notepad" with the command for your preferred editor):

```
git config --global core.editor "notepad"
```

To list all your current Git configuration settings, you can use:

```
git config --list
```

Remember that these commands will affect your global Git configuration. If you want to configure settings for a specific repository, you can omit the `--global` flag and run these commands within the repository directory.

What's the best way to make your connection with GitHub safe?

To ensure a safe connection with GitHub, follow these best practices:

1. **Enable Two-Factor Authentication (2FA):** Use 2FA to add an extra layer of security to your GitHub account.
2. **Use Strong Passwords:** Create complex passwords that include a mix of letters, numbers, and symbols. Avoid using easily guessable information.

3. **SSH Keys:** Use SSH keys for authentication instead of passwords. This provides stronger security and is less susceptible to phishing attacks.
4. **Regularly Update:** Keep your operating system, browser, and any development tools up to date with the latest security patches.

5. **Review App Permissions:** Review and revoke unnecessary third-party app permissions that have access to your GitHub account.
6. **Beware of Phishing:** Be cautious of emails or messages asking for your GitHub credentials. Always verify the source before providing any sensitive information.

7. **Monitor Account Activity:** Regularly review your GitHub account activity for any suspicious login attempts.
8. **Use Verified Tools:** Only use reputable Git clients and development tools.

9. **Private Repositories:** If you're working on sensitive projects, consider using private repositories to limit access.
10. **GitHub Security Settings:** Explore GitHub's security settings and features, such as security alerts, dependency insights, and code scanning.

By implementing these practices, you can significantly enhance the security of your connection with GitHub.

Connecting Git to GitHub

The process is made up of the following steps:

- *SSH Keys generation (private and public)*
- *Private key registration with SSH*
- *Public key registration on GitHub*

Ed25519

Ed25519 was introduced in OpenSSH 6.5 of January 2014: "Ed25519 is an elliptic curve signature scheme that offers better security than ECDSA and DSA and good performance". Its main strengths are its speed, its constant-time run time (and resistance against side-channel attacks), and its lack of nebulous hard-coded constants.

```
ssh-keygen -t ed25519 -C "unique name to identify this key"
```



```

15:54:10 .ssh 108ms
→ ssh-keygen -t ed25519 -C "clldscotti@gmail.com" -f /c/users/clldu/OneDrive/Documentos/ssh_keys/.ssh/id_ed25519
Generating public/private ed25519 key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/users/clldu/OneDrive/Documentos/ssh_keys/.ssh/id_ed25519
Your public key has been saved in /c/users/clldu/OneDrive/Documentos/ssh_keys/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:gADvYhUdzLMXnKwaDR4Qe/oW1MNGf6CQhMCJuRcMZAU clldscotti@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|*E*=*+000+0      |
|++00=** ++.      |
|..+0*=*.. .      |
|..0 +00+. .      |
|.0.. .0 S        |
|.. ...           |
|   0             |
|   .             |
+-----+
+----[SHA256]-----+
16:19:42 .ssh 23.163s
→ ls
id_ed25519 id_ed25519.pub
16:28:47 .ssh 102ms
→ eval "$(ssh-agent)"
Agent pid 924
16:31:32 .ssh 114ms
→ ssh-add /c/users/clldu/onedrive/Documentos/ssh_keys/.ssh/id_ed25519
Enter passphrase for /c/users/clldu/onedrive/Documentos/ssh_keys/.ssh/id_ed25519:
Identity added: /c/users/clldu/onedrive/Documentos/ssh_keys/.ssh/id_ed25519 (clldscotti@gmail.com)
16:33:33 .ssh 11.796s
→ ssh -T git@github.com
The authenticity of host 'github.com (140.82.112.3)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.

```

Part 3

Hosting your project on Github

- *The phonetic website project*
- *Hosting your project*
- *Clonning the repository*
- *Adding files to the project*
- *Adding files to the index(staging area)*
- *Commiting changes to the repository*
- *Pushing a new version to Github*

Links

Git (and/or GitHub), for Dummies

<https://blog.stackademic.com/git-and-or-github-for-dummies-6a363d86e1b5>

Top Git & GitHub Tips & Tricks for Smooth Development

<https://blog.stackademic.com/top-git-github-tips-tricks-for-smooth-development-a3b66746ca9c>

Links

Setting Up SSH Access for Private GitHub Repositories

<https://medium.com/@ak4634/setting-up-ssh-access-for-private-github-repositories-2e6ec8ff551b>

How Do I Delete a Git Branch Locally and Remotely?

<https://medium.com/git-happy/how-do-i-delete-a-git-branch-locally-and-remotely-97bc772520a7>