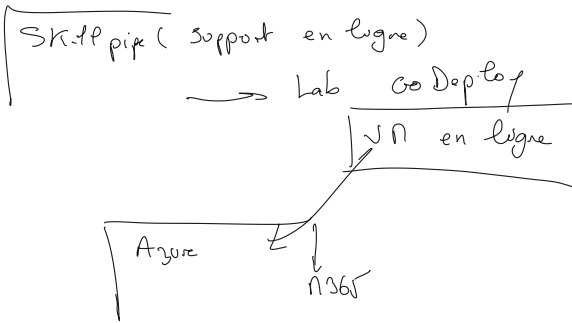
Alexandre
Rodrigue
Kenny
Nicolas

[redacted]

SkillPipe Reader



Lien vers les labs du cours :
https://github.com/MicrosoftLearning/AZ-040T00-Automating-Administration-with-PowerShell

Labs en ligne : https://leaflearning365.com/skillpipe/

http://lms.godeploy.it Code sur ReDeem : NNBSOV (activable 4 fois)

Reference PowerShell : How to use the PowerShell documentation - PowerShell | Microsoft Docs
Powershell dev avec VS Code : Using Visual Studio Code for PowerShell Development - PowerShell | Microsoft Docs

Trouver les commandes par module :
Get-command -Module ActiveDirectory

2002 : première version du .Net

2005 : v1 du powershell (qui fonctionne sur le .Net Framework (Windows uniquement)

2016 : v1 du .Net "core" : qui permet l'execution de code sous windows / linux, avec des différences. (est un sous ensemble du .Net Framework)

core v1

dernière version du .Net framework : 4.8..

v3 (dernière version core)

v6 (2021)

v7 (2022)

Get-Service | Get-Member

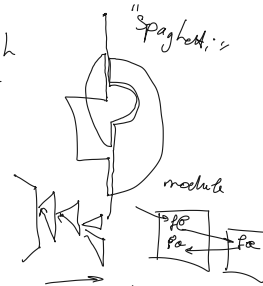
TypeName : System.ServiceProcess.**ServiceController**

Name MemberType Definition

Niveaux fonctionnels de Programmation:

- Assembleur
- Prog. naturelle → BASIC, bash cmd
- Prog. structurée → C, pascal, Php
- Prog. orientée Objet
  → C++, ptp objet, Delphi, Java, C#, powershell

(Prog. fonctionnelle
↳ passer des fonctions à des fonctions)



⇒ Objets = ??

"Concept de la voiture" → Caractéristiques conceptuelles (définitions de membres)
↳ la classe en prog.

"La voiture sur le parking" → Instances de la classe Voiture
→ Valeurs aux membres

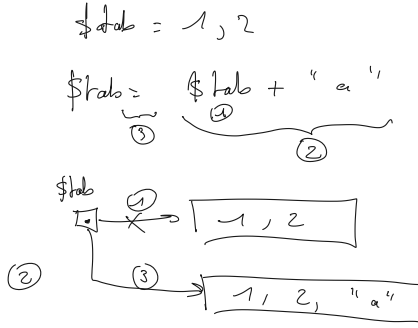
Type, Classe
↓
Membres
↙ ↘
Propriétés → Méthodes, Opérations, Verbes
↙ ↘
get get set

 # 3

Get-ADUser -Filter * -Properties whenCreated | Sort-Object -Property whenCreated -Descending | Select-Object -Property Name,@{n='Account age (days)';e={(New-TimeSpan -Start $PSItem.whenCreated).Days}}

WMI Explorer :
Releases · vinaypamnani/wmie2 · GitHub

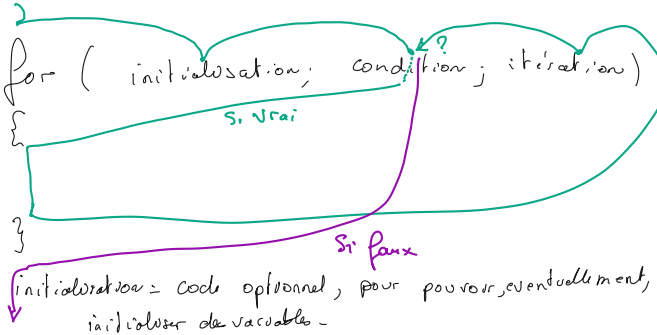
lowercasing
UPPERCASING
Pascal Casing
camel Casing



Galerie Powershell :
PowerShell Gallery | Home

Apprendre à utiliser Git :
Learn Git Branching

PKIs pour les scripts



Trouver les couleurs :
write-host "bonjour" -ForegroundColor ([System.ConsoleColor]:: <tab...>

Git

2

Poste de dev → fichiers

→ Besoin de Git en ligne de Commande

/ c:\

PF
Windows
data / SRC

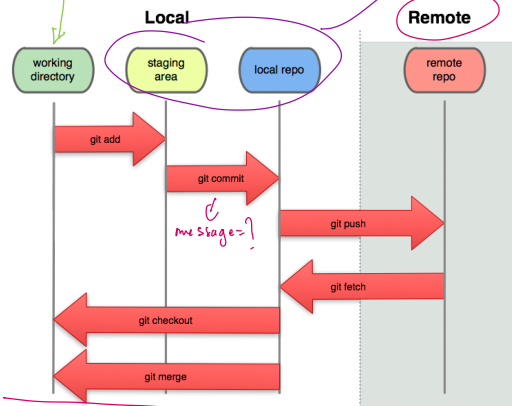
[/] racine git

① git init

•git

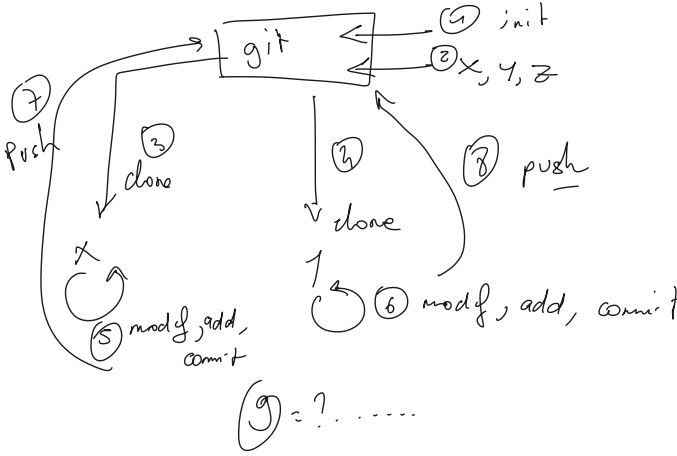
gitlab
hub
etc...

"Vue demandée des fichiers"

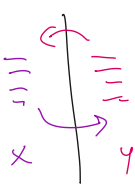


git clone → init

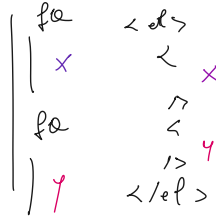
→ git push



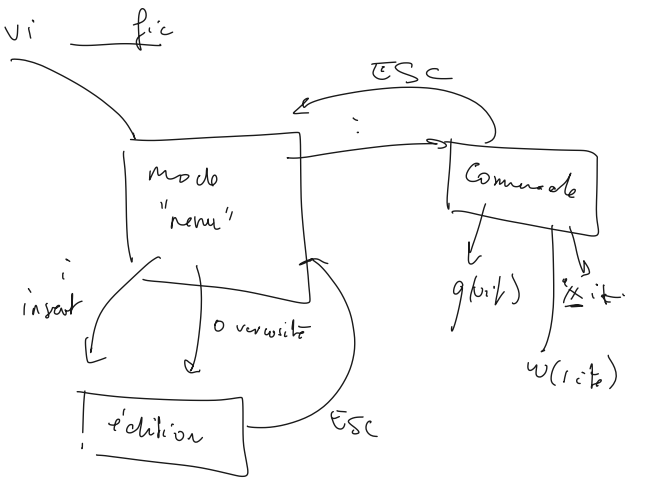
Risques de fusion :



code, JSON, XML



[Git - Downloading Package (git-scm.com)]



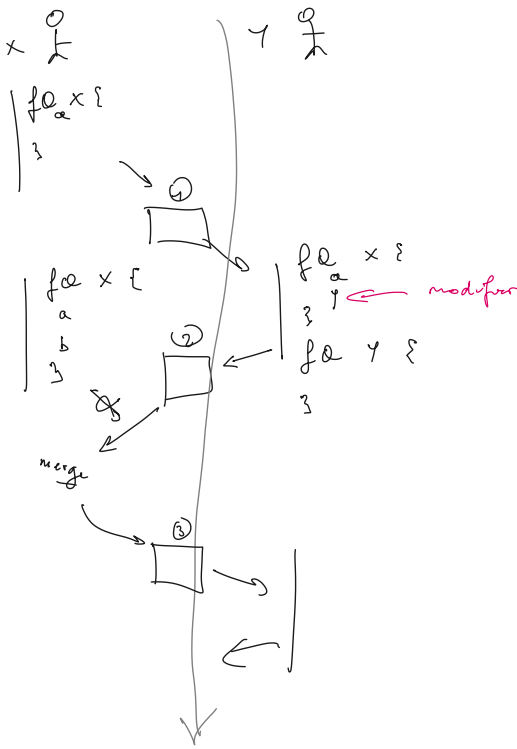
Création de repo local :
git init
Pousser sur GitLab :
git push --set-upstream git@gitlab.com:philippe59710fr/$(git rev-parse --show-toplevel | xargs basename).git $(git rev-parse --abbrev-ref HEAD)

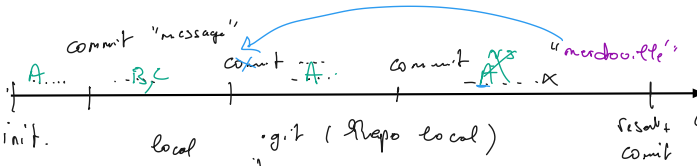
git clone https://gitlab.com/philippe59710fr/stagepwsh.git

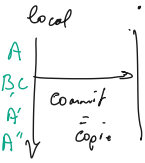
git add .
(SET EDITOR=code) SET EDITOR=
git commit :
git config global.....

git commit
-> i (insertion) -> ESC -> :x (sous vi)

git push (envoie en ligne)

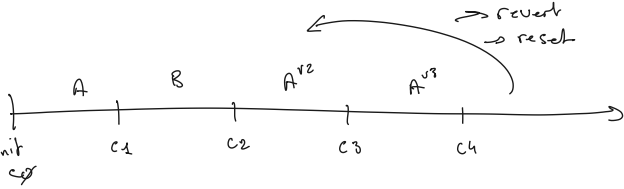

x
y
for x {
}
1
for x {
a
b
}
for x {
a
y
}
for y {
}
modifier
2
merge
3





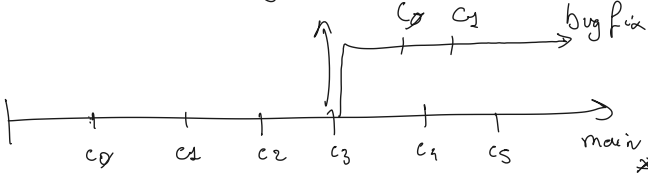
→ git revert (annule 1 commit)

→ git resett (retour en arrière complet)

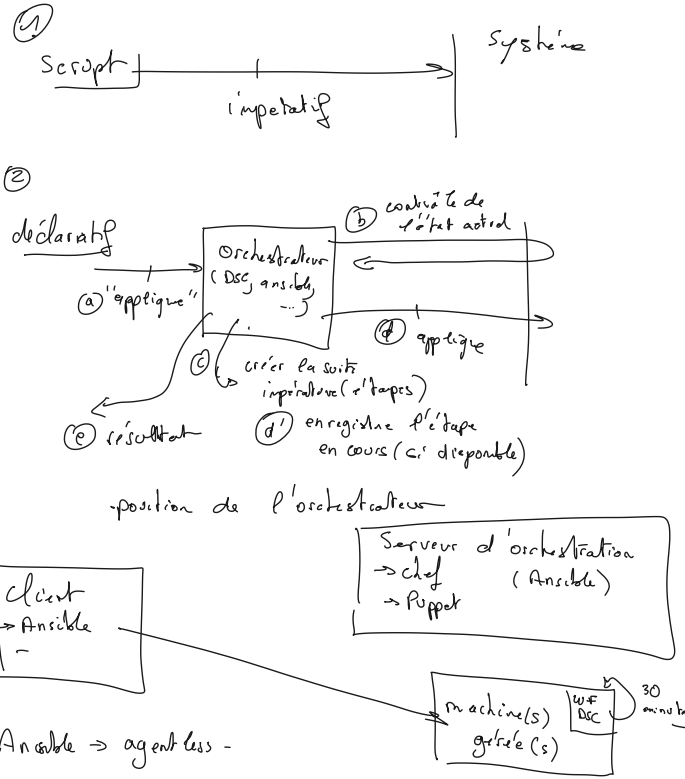


branche

git branch <nom>



git checkout <nom de branch>



https://www.metricsthatmatter.com/MTMStudent/m2ilille

Learning and Development Services (microsoft.com)