

# INTRODUCTION À



# SOMMAIRE

1. Qui suis-je ?
2. Les objectifs du cours
3. git
4. Pourquoi ?
5. Quelques commandes
6. Différents outils
7. Exercice pratique
8. Mes recommandations
9. Questions & Remerciements
10. Sources

# Qui suis-je ?

```
1  {  
2    "FullName": "Loïc POCCARD",  
3    "Age": "25",  
4    "Job": {  
5      "Role": "Développeur Front-End",  
6      "JobType": "Freelance",  
7      "Since": 4  
8    },  
9    "Favorite language": "JavaScript",  
10   "Hobbies": ["Automobile", "Horlogerie"]  
11 }
```

# Les objectifs du cours

```
12  {  
13      "Goals": [  
14          "Découvrir git",  
15          "Comprendre son utilité",  
16          "Apprendre ses premières commande git",  
17          "Découvrir gitlab & github",  
18          "Exercice pratique"  
19      ]  
20  }
```

# git

- Logiciel libre
- Logiciel de gestion des versions (versionning)
- Linus Torvalds
- Pair à pair (peer to peer)
- Plusieurs versions du code à différents endroits

# Pourquoi ?

- Versionner son code
- Travailler à plusieurs sur un même projet
- Revenir en arrière sans tout réécrire
- Faciliter la résolution des erreurs

# Quelques commandes

These are common Git commands used in various situations:

start a working area (see also: `git help tutorial`)

<code>clone</code>	Clone a repository into a new directory
<code>init</code>	Create an empty Git repository or reinitialize an existing one

work on the current change (see also: `git help everyday`)

<code>add</code>	Add file contents to the index
<code>mv</code>	Move or rename a file, a directory, or a symlink
<code>restore</code>	Restore working tree files
<code>rm</code>	Remove files from the working tree and from the index
<code>sparse-checkout</code>	Initialize and modify the sparse-checkout

examine the history and state (see also: `git help revisions`)

<code>bisect</code>	Use binary search to find the commit that introduced a bug
<code>diff</code>	Show changes between commits, commit and working tree, etc
<code>grep</code>	Print lines matching a pattern
<code>log</code>	Show commit logs
<code>show</code>	Show various types of objects
<code>status</code>	Show the working tree status

grow, mark and tweak your common history

<code>branch</code>	List, create, or delete branches
<code>commit</code>	Record changes to the repository
<code>merge</code>	Join two or more development histories together
<code>rebase</code>	Reapply commits on top of another base tip
<code>reset</code>	Reset current HEAD to the specified state
<code>switch</code>	Switch branches
<code>tag</code>	Create, list, delete or verify a tag object signed with GPG

collaborate (see also: `git help workflows`)

<code>fetch</code>	Download objects and refs from another repository
<code>pull</code>	Fetch from and integrate with another repository or a local branch
<code>push</code>	Update remote refs along with associated objects

# Différents outils





# Exercice pratique

```
21 {  
22     "Run": "Se rendre sur la page https://github.com/Poccard1/evogue-practice"  
23 }
```

# Mes recommandations

- Pratiquer un maximum
- Privilégier l'utilisation de git en ligne de commande plutôt qu'avec des outils graphiques



# Questions & Remerciements

EVOGUE



# Sources

- [fr.wikipedia.org/wiki/Git](https://fr.wikipedia.org/wiki/Git)