LP{

# INTRODUCTION À

git

LP{

git

# SOMMAIRE

# Qui suis-je ?

```
{
    "FullName": "Loïc POCCARD",
    "Age": "25",
    "Job": {
        "Role": "Développeur Front-End",
        "JobType": "Freelance",
        "Since": 4
    },
    "Favorite language": "JavaScript",
    "Hobbies": ["Automobile", "Horlogerie"]
}
```

LP{

git

# Les objectifs du cours

```
{
    "Goals": [
        "Découvrir git",
        "Comprendre son utilité",
        "Apprendre ses premières commande git",
        "Découvrir gitlab & github",
        "Exercice pratique"
    ]
}
```

LP{

git

# git

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

LP{

git

# Pourquoi ?

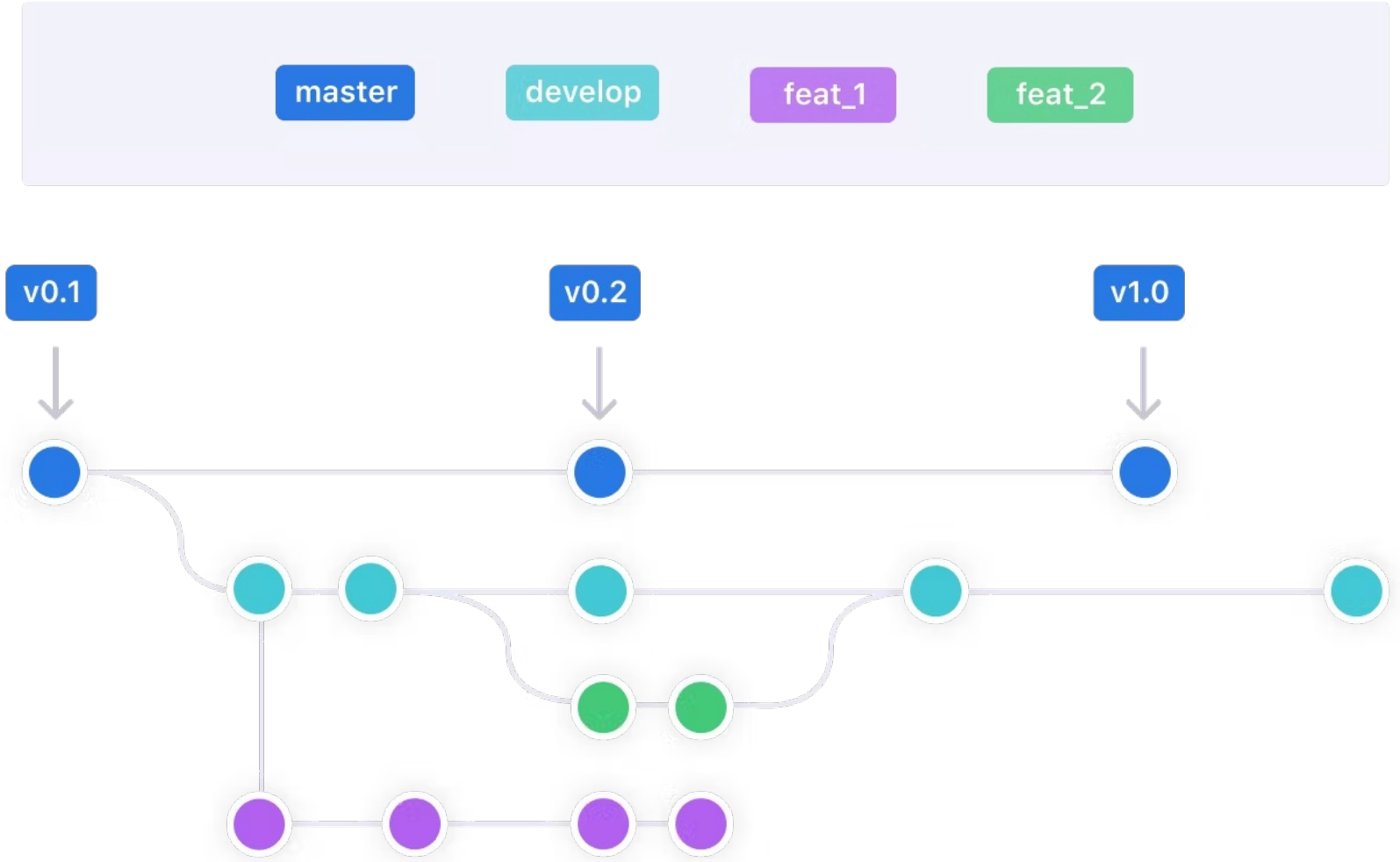
- Versionner son code
- Travailler à plusieurs sur un même projet
- Revenir en arrière sans tout réécrire
- Faciliter la résolution des erreurs

LP{

git

# Son fonctionnement

- Plusieurs branchs
- Des commits
- Des collaborateurs
- Un code source
- Plusieurs versions


master
develop
feat_1
feat_2
v0.1
v0.2
v1.0

# Quelques commandes

```
These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
   clone             Clone a repository into a new directory
   init              Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
   add               Add file contents to the index
   mv                Move or rename a file, a directory, or a symlink
   restore           Restore working tree files
   rm                Remove files from the working tree and from the index
   sparse-checkout   Initialize and modify the sparse-checkout

examine the history and state (see also: git help revisions)
   bisect            Use binary search to find the commit that introduced a bug
   diff              Show changes between commits, commit and working tree, etc
   grep              Print lines matching a pattern
   log               Show commit logs
   show              Show various types of objects
   status            Show the working tree status

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

collaborate (see also: git help workflows)
   fetch             Download objects and refs from another repository
   pull              Fetch from and integrate with another repository or a local branch
   push              Update remote refs along with associated objects
```

LP{

git

# Différents outils


LP{

git

# Exercice pratique

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

LP{

git

# Mes recommandations

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

LP{

# Questions & Remerciements

EVOGUE

# Sources

- fr.wikipedia.org/wiki/Git