

Construction de l'application Marvel

Mettre en oeuvre les différents concepts manipulés au cours du développement d'une application web moderne.

Objectifs

Travailler de manière professionnelle

- Travailler en équipe: agilité, versionning, historique, branches, tags, issues, pull requests...
- Utiliser les bons outils: IDE, Source Control, linters, formatters...
- Respecter les bonnes pratiques et normes : qualité, tests, CI/CD...
- Documenter son travail: README, Documentation...

Concepts manipulés

- **agilité** : méthode de développement
- **développement web** : html, css, javascript, typescript, react
- **aide au développement**: vscode, github copilot, eslint, prettier
- **travail collaboratif**: git, github
- **tests unitaires, tests d'intégration**: jest, cypress, playwright
- **intégration continue**: github actions
- **déploiement continu**: github actions
- **documentation**: markdown, mkdocs
- **qualimétrie**: sonarqube

Initialisation du projet marvel-app

- Créer un dépôt sur GitHub avec le nom `marvel-app` . Cocher la case `Initialize this repository with a README`
 - Le fichier README.md permet de décrire le projet, son objectif, son fonctionnement, les prérequis, les étapes d'installation, de configuration et de lancement. Nous y reviendrons plus tard.
- Cloner le dépôt sur votre machine locale avec la commande:
 - `git clone https://github.com/<votre-login>/marvel-app.git`
 - Ne pas utiliser le lecteur réseau **Z:** pour travailler sur le projet. Utiliser un répertoire local sur votre machine pour éviter les problèmes.

Version 0.1.0 - Objectif

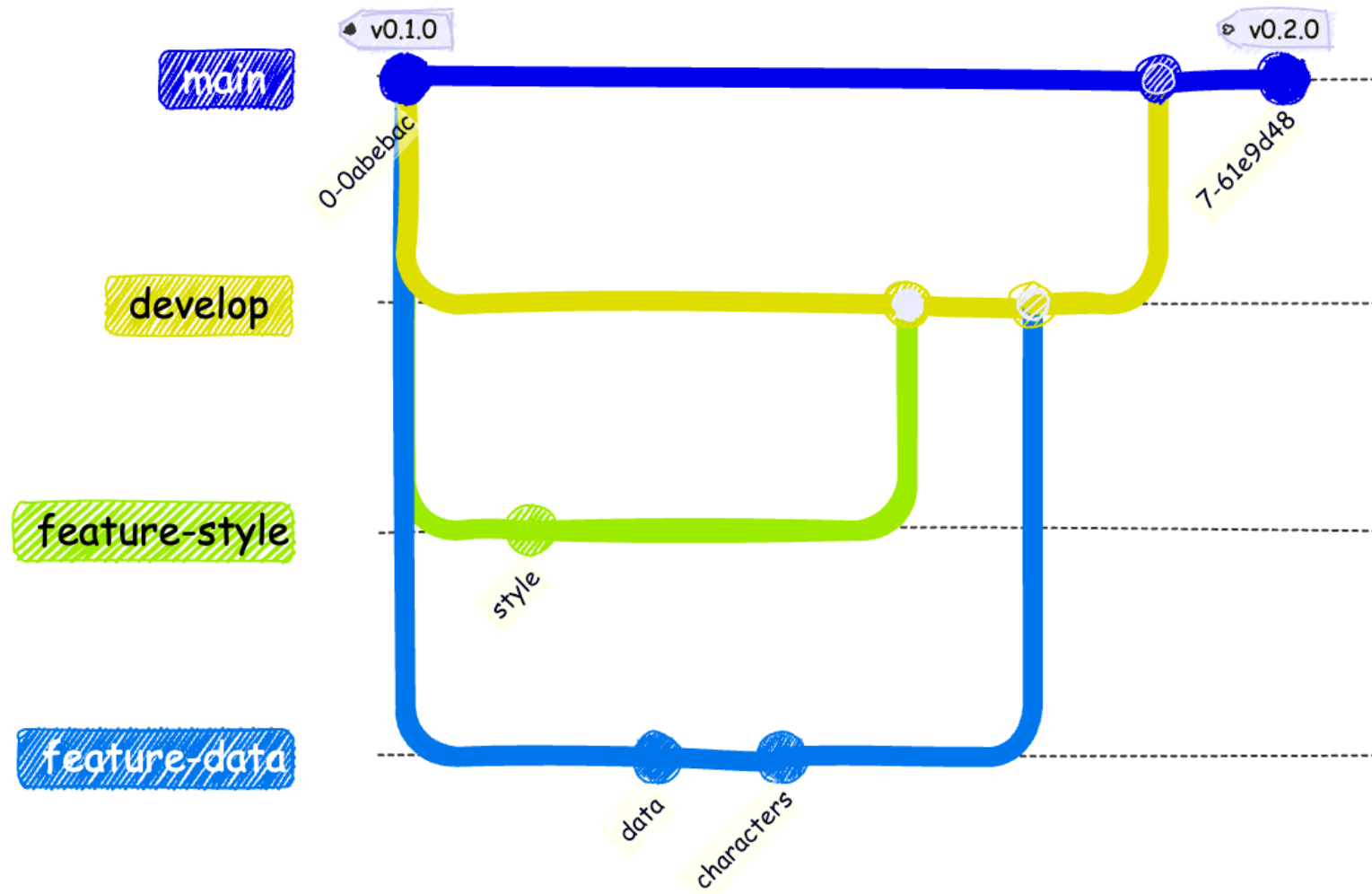


- Gestion de version - **semver**
- Préparation du projet
 - Utilisation de **npm** pour gérer les dépendances du projet
 - Utilisation de **browser-sync** pour créer un serveur web local
- Implémentation d'une page web
 - Contenu statique de la page **HTML** (HyperText Markup Language)
 - Application du style avec **CSS** (Cascading Style Sheets)

Version 0.2.0 - Objectif

- Utilisation de **git** pour travailler sur plusieurs fonctionnalités en même temps
 - Gestion des branches
 - git flow
- Ajout de contenu **JavaScript** pour récupérer des données depuis un fichier **JSON**
- Ajout de style **CSS** pour améliorer l'affichage

Version 0.2.0 - Objectif (suite)



Version 0.3.0 - Objectif

- Transformation de la page statique en application web dynamique
 - Utilisation de **React** pour créer des composants réutilisables
- Détection d'un bug dans la version 0.3.0
 - Création d'une branche de type correctif

Version 0.3.1 - Objectif (suite)



Version 0.4.0 - Objectif

- Création de plusieurs pages (Home, About, Contact)
- Utilisation de react-router pour gérer la navigation entre les pages
- Mise en oeuvre d'un layout pour les pages (header, footer, etc...)

Version 0.4.0 - Objectif (suite)



Version 1.0.0 - Objectif

- Création de la page de détail d'un personnage
- Sécurisation de l'application
 - Création d'un workflow de CI/CD avec GitHub Actions
 - Protection des branches `main` et `develop`
- Création d'une release 1.0.0 et déploiement de l'application