

# Conception Et Programmation Web

**LIFWEB** - <http://lifweb.pages.univ-lyon1.fr/>

**Semestre printemps 2024-2025**

**L3 - UCBL**

**Aurélien Tabard** - <https://tabard.fr/>, basé sur les cours de Romuald Thion

# **Outillage et bonnes pratiques**

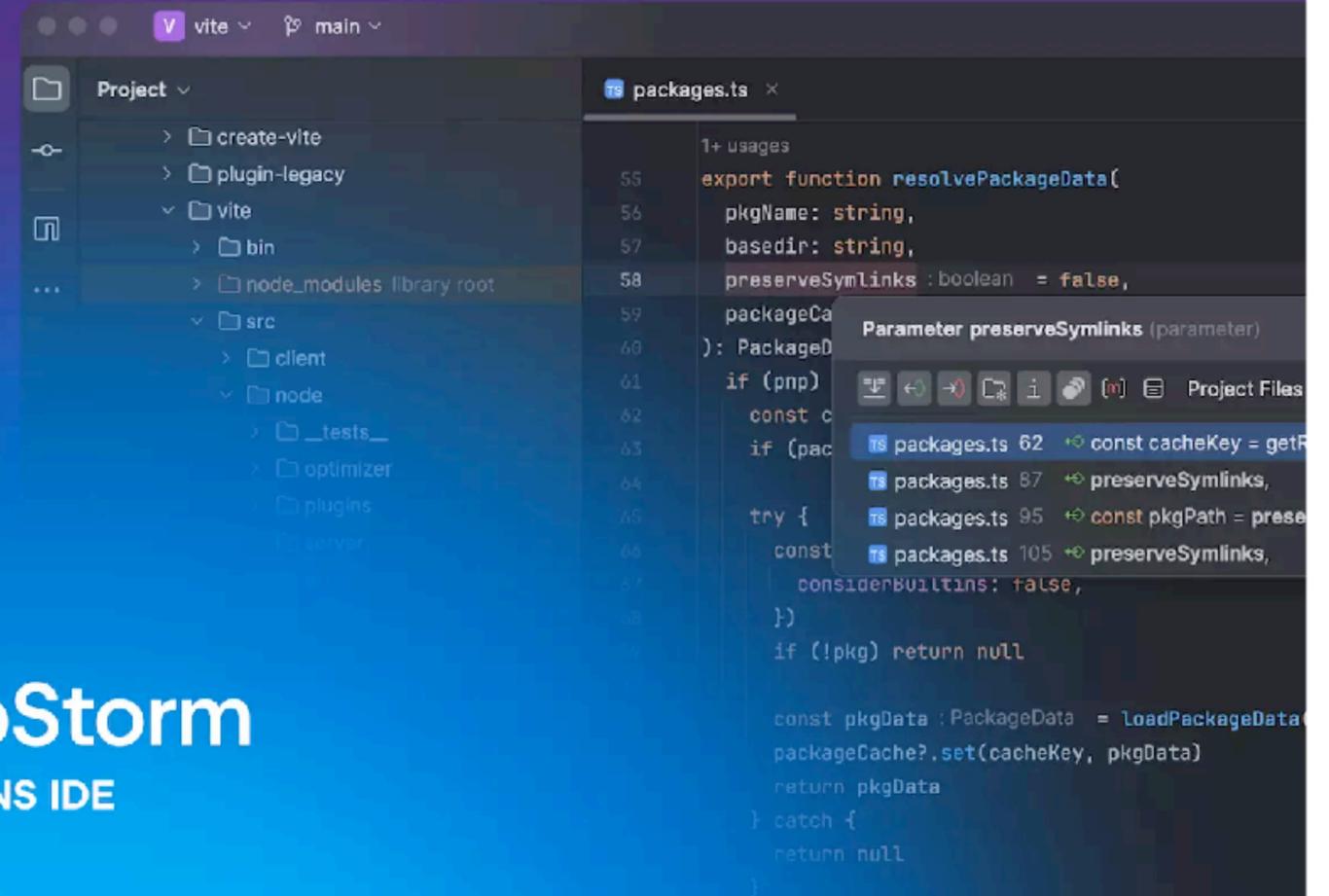
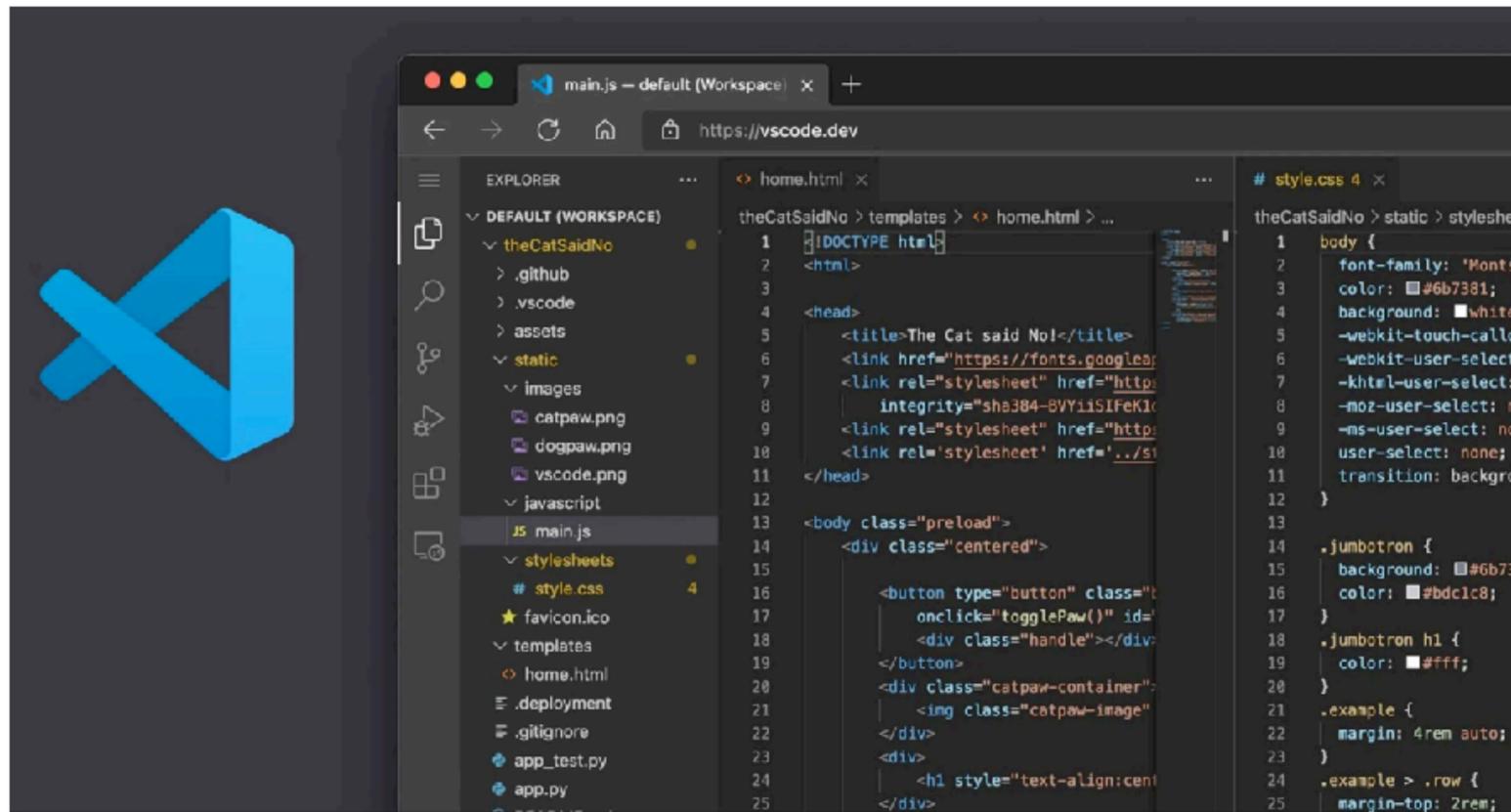
**Aurélien Tabard - 2024-2025**

# Outillage et bonnes pratiques

## Plan

1. Environnement de développement
2. Gérer votre code et collaborer
3. Automatiser, déployer, et surveiller
4. Gérer vos dépendances de projet
5. Packager vos projets

# Integrated Development Environment (IDE)



 **WebStorm**  
JETBRAINS IDE

# Linters

Détection des erreurs de syntaxe dans le code

Des règles pour le projet entier

Améliorer la qualité du code

<https://eslint.org/>

<https://stylelint.io/>

# Formatters

Réorganiser le code pour le rendre plus lisible et plus consistant

Gagner du temps sur le formattage

Produire du code uniforme et cohérent

<https://prettier.io/>

<https://biomejs.dev/>

# Typescript

[www.typescriptlang.org](http://www.typescriptlang.org)

Objectif : améliorer le typage de JavaScript

```
const Name: string = "Toto" // ✓
```

```
const Age: number = 12 // ✓
```

```
const Surname: string = false // ✗
```

```
const YearOfBirth: number = "1998" // ✗
```

# JsDoc

[jsdoc.app](http://jsdoc.app)

 Langage de balisage utilisé pour documenter du code source JavaScript

```
/**
 * Function to add a book to the list.
 *
 * @constructor
 *
 * @param {string} title - The title of the book.
 * @param {string} author - The author of the book.
 */
function addBook(title, author) {
  blablabla...
}
```

Pour générer la doc:  
> jsdoc book.js

# Extensions d'IDE

## Live Server

- Serveur web minimaliste avec un système de HMR

## HTML CSS Support

- Extension VSCode qui permet la complétion des attributs dans le code

## IntelliSense

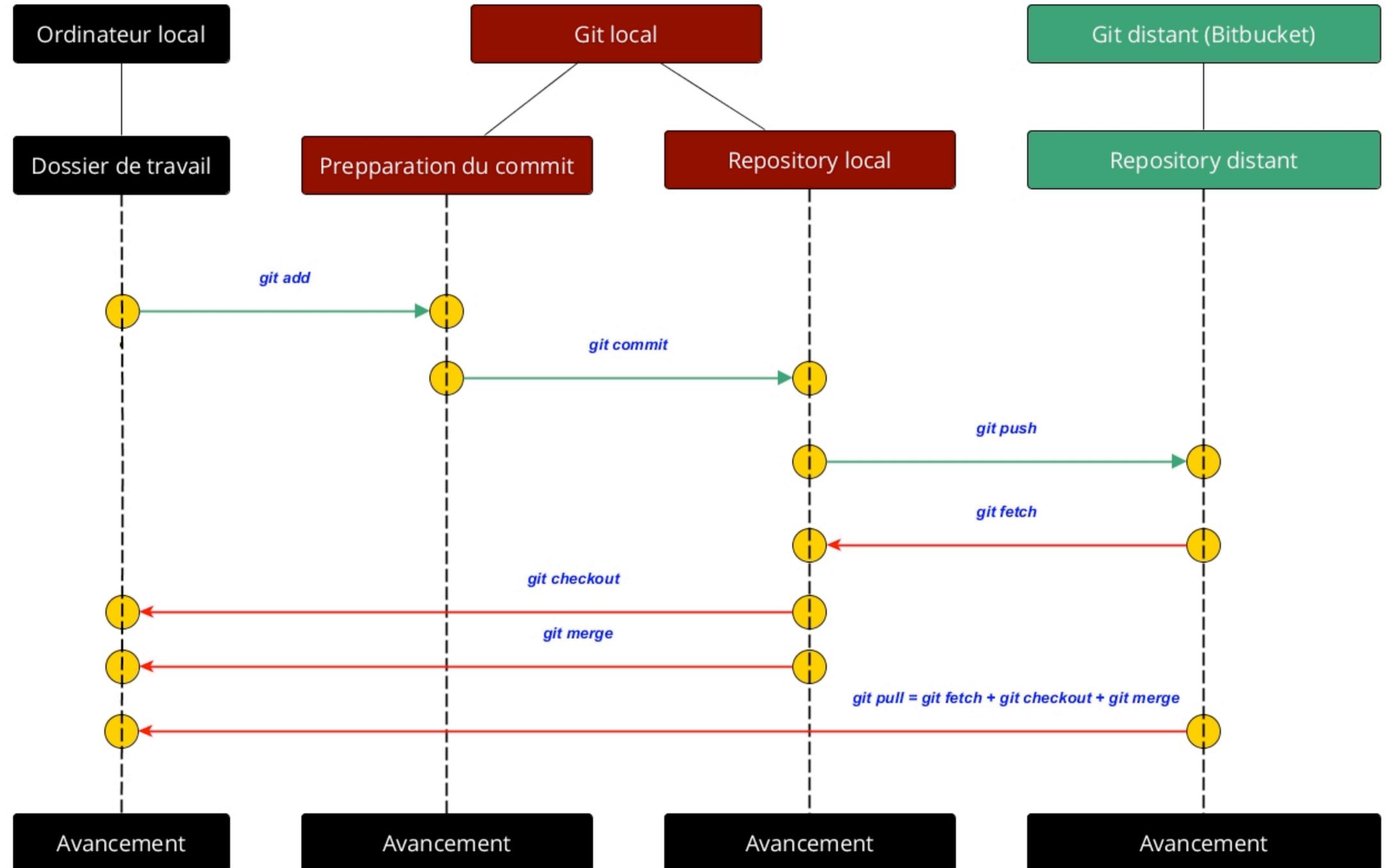
- Fonction d'édition de code, utilisée pour la complétion de code, les informations sur les paramètres...

# Outillage et bonnes pratiques

## Plan

1. Environnement de développement
2. Gérer votre code et collaborer
3. Automatiser, déployer, et surveiller
4. Gérer vos dépendances de projet
5. Packager vos projets

# Git

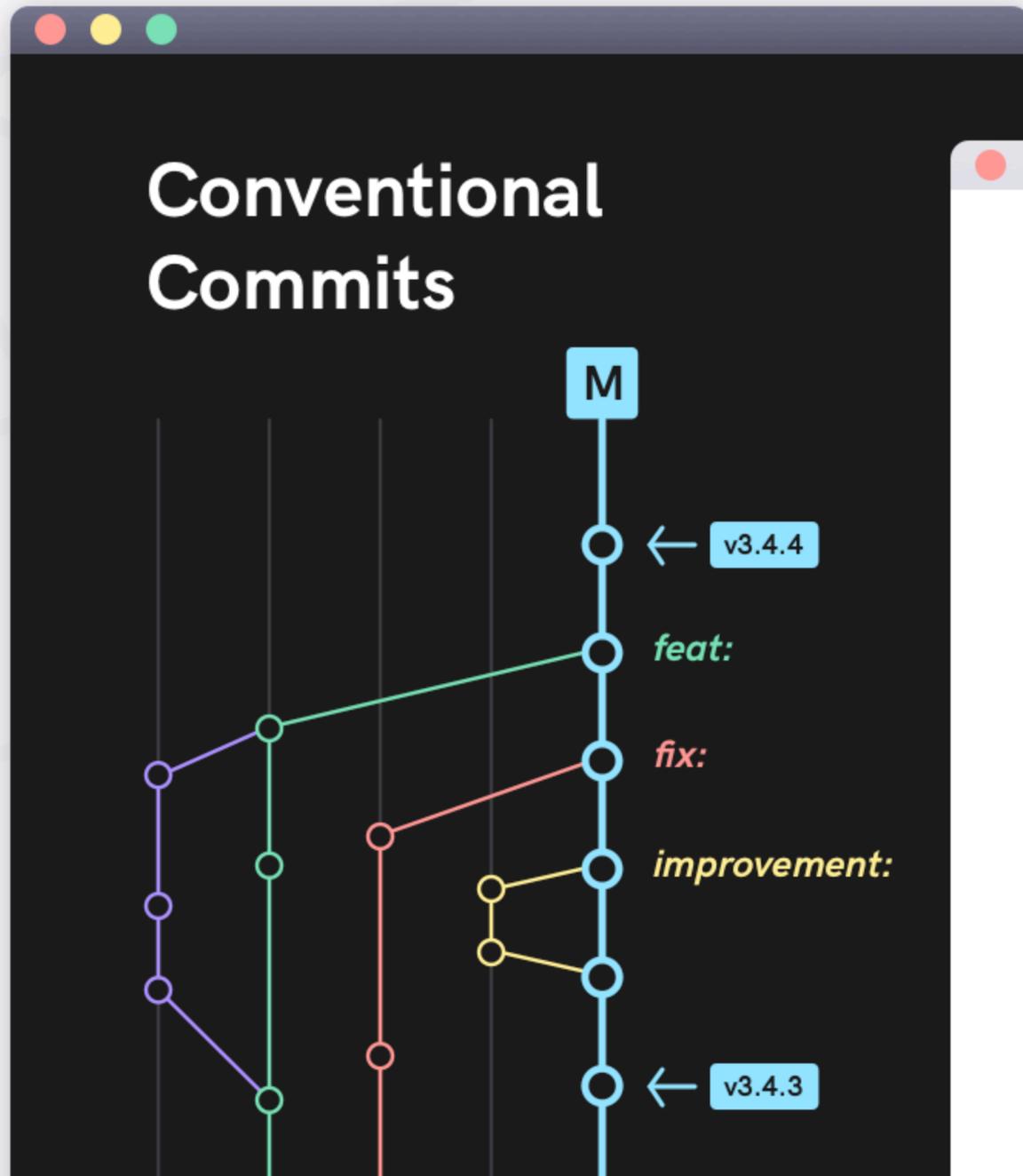


# Erreurs classiques

Des noms commits inadaptés

Des fichiers qui ne devraient pas être là !

Aucune merge request / pull request



# Changelog

**v3.4.4** 12/06/2019

- feat:* Image gallery upload
- fix:* Fixed MacOS 10.13.1. installation bugs
- improvement:* Added links to management icons

**v3.4.3** 10/06/2019

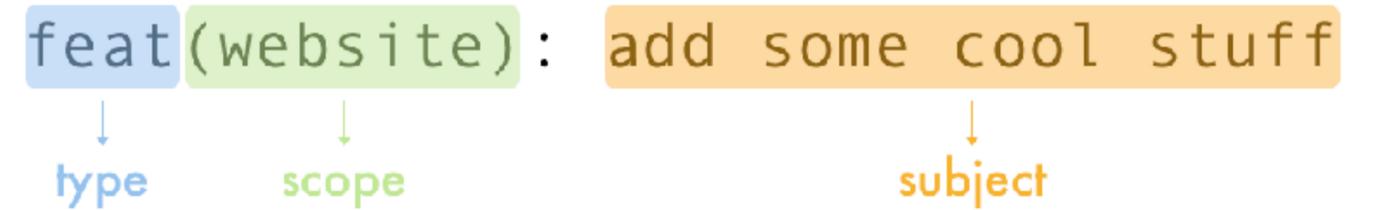
- improvement:* editing and deleting photos on mobile
- fix:* Fixed broken links on footer

**v3.4.2** 10/06/2019

## Conventional Commits

# Conventional commits

## Structure des messages :



```
<type>[optional scope]: <description>
```

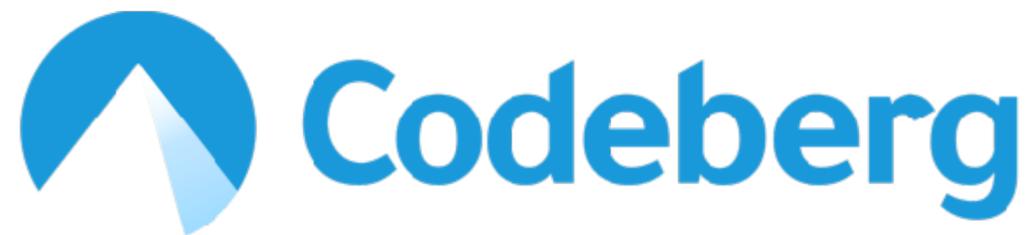
```
[optional body]
```

```
[optional footer(s)]
```

### Quelques types :

- **feat**: (new feature for the user, not a new feature for build script)
- **fix**: (bug fix for the user, not a fix to a build script)
- **docs**: (changes to the documentation)
- **style**: (formatting, missing semi colons, etc; no production code change)
- **refactor**: (refactoring production code, eg. renaming a variable)
- **test**: (adding missing tests, refactoring tests; no production code change)
- **chore**: (updating grunt tasks etc; no production code change)

# Quelles plateformes ?



# Pourquoi utiliser une forge ?

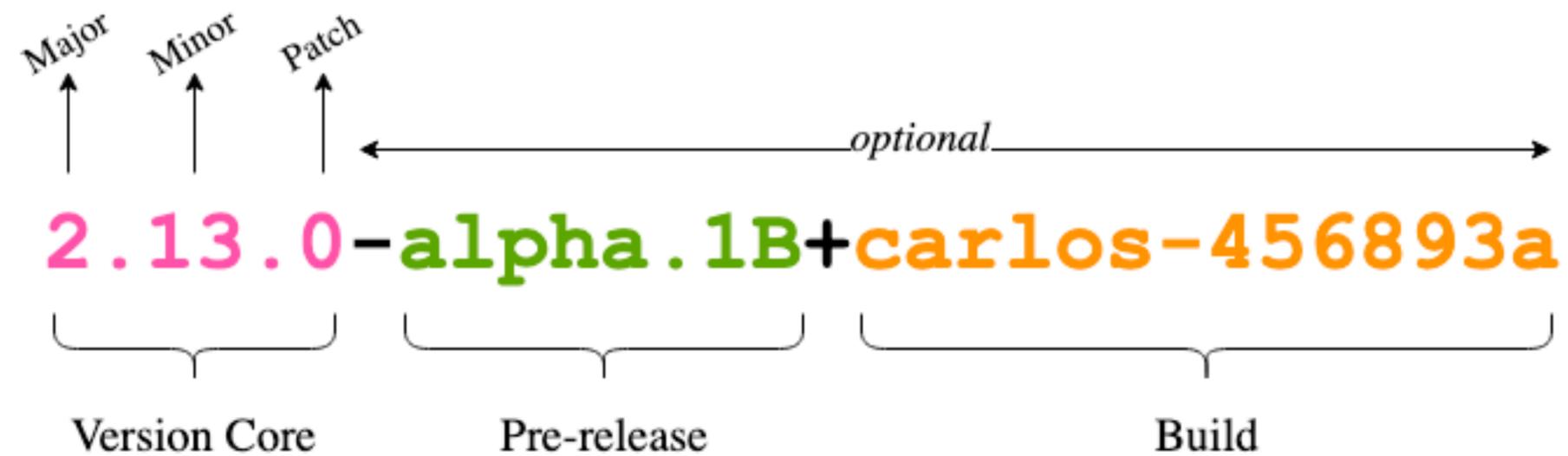
- Interface de visualisation de projets Git
- Utilisation des merge requests / pull requests
- Gestion de projets avec les issues et boards
- Continuous Integration / Delivery (CI/CD)
  - Exécution automatique de tests, du packaging, du déploiement, etc.
- Déploiement sur GitLab / Github Pages
- Espace de documentation (Wiki...)

# Intro à gitlab pour la gestion de projet

The screenshot shows a GitLab issue page for the project 'M1IF10 Projet Transversal 2020'. The issue title is 'Vidéo d'intro à GitLab'. The issue is currently 'Open' and was created 2 minutes ago by 'TABARD AURELIEN'. The issue description contains a video placeholder with the text 'Drop or upload designs to attach'. The right sidebar shows the issue's metadata, including the assignee 'TABARD AURELIEN', milestone 'Lancement', time tracking (estimated 2h), due date 'Sep 26, 2020', labels 'Gestion de projet' and 'developpement', confidentiality 'Not confidential', and lock status 'Unlocked'. The left sidebar shows the project navigation menu with 'Issues' selected. The bottom of the page shows a comment input field with a rich text editor and an 'Attach a file' button.

# Nommage des versions

## Semantic versioning



<https://semver.org/>

# Outillage et bonnes pratiques

## Plan

1. Environnement de développement
2. Gérer votre code et collaborer
3. Automatiser, déployer, et surveiller
4. Gérer vos dépendances de projet
5. Packager vos projets

# Continuous Integration (CI)

Détecter rapidement les erreurs

Automatiser les bonnes pratiques

Fusionner régulièrement le code pour s'assurer de sa qualité !

[Husky](#)

[Lint Staged](#)

[Gitlab CI/CD](#)

# Continuous Delivery (CD)

Automatiser le déploiement de l'application

Livrer rapidement et avec sérénité 🧘

Mettre en place des tests automatisés

[Gitlab CI/CD](#)

# Déploiement

Github/Gitlab Pages - <https://pages.github.com/> - demo

Vercel - <https://vercel.com> - Permet d'héberger votre app en quelques clics.  
Propose aussi des métriques.

Netlify - <https://www.netlify.com/>

Heroku - <https://www.heroku.com/>

Clevercloud - <https://www.clever-cloud.com>

OVH - <https://www.ovhcloud.com> - mutualisé ou VM dédiée

# Monitoring

## Comment savoir ce qu'il se passe sur son serveur

Permet de surveiller et analyser les données de votre application avec des graphes.

<https://grafana.com/>

Observe vos applications et vos machines et propose beaucoup de fonctionnalités.

<https://newrelic.com>



# Suivi de performance

Fournir des insights pour optimiser les perfs et garantir une expérience utilisateur optimale

<https://www.datadoghq.com/>

Tests de perfs: Evaluer les performances des applications web en simulant un grand nombre d'utilisateurs concurrents.

<https://jmeter.apache.org/>

# Notifications

## Mail

### [PIXEL] Bug Report job : 172720758

#### Operation : vod

- Job Status : **Failed**
- Step Name : Transcode
- Sources :
  - [REDACTED]\_ur-un-air-de-Noel\_c12565627\_Sur-un-air-de-No.mp4
- Targets :
  - name : web - clip id : Sur-un-air-de-Noel\_c12565627\_Sur-un-air-de-No
  - name : proxy - clip id : Sur-un-air-de-Noel\_c12565627\_Sur-un-air-de-No
  - name : iphone - clip id : Sur-un-air-de-Noel\_c12565627\_Sur-un-air-de-No
  - name : ipad - clip id : Sur-un-air-de-Noel\_c12565627\_Sur-un-air-de-No

## Slack

Customer rtlbe

Error on clip [13052474](#) during workflow [rtlbe-13052474-d9cc4e87-ffb3-4702-8409-c95dadad3c0c](#) during step [pixel-transcode-workflow-resumer](#) => check on [videotools](#)

#### Sources

#### Error

```
Error: The clipping start timecode [13:16:52:01] was not found in input number [1]
```

# Outillage et bonnes pratiques

## Plan

1. Environnement de développement
2. Gérer votre code et collaborer
3. Automatiser, déployer, et surveiller
4. Gérer vos dépendances de projet
5. Packager vos projets

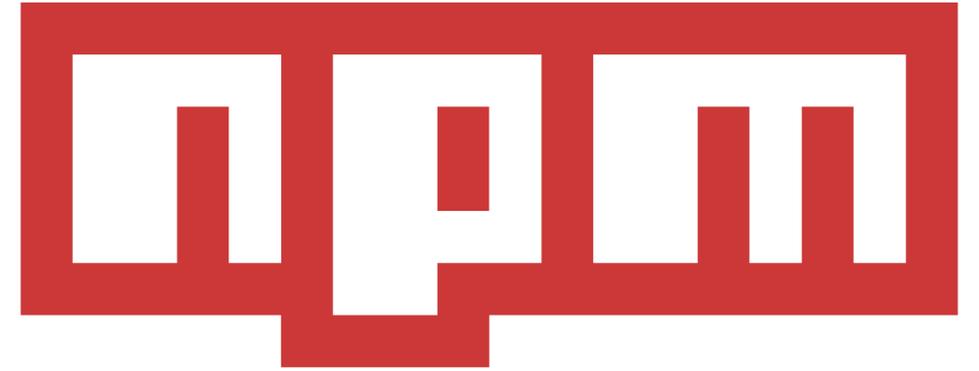
# NPM

`npm` : Node Packet Manager

- Gestionnaire de packages
- Installation de dépendances
- Scripts personnalisés

Alternatives / successeurs:

- `pNpM`
- `Yarn`



# NPM

*npm* est à Node.js ce que *pip* est à Python, avec un support natif des environnements type [venv](#) .

```
npm --version
# 8.16.0
npm init
# question interactives
npm install slugify
# added 1 package in 2s
cat main.js
# import slugify from "slugify"
# console.log(slugify("C'est un test ♥ !"));
node main.js
# C'est-un-test-love-!
```

# Décrire son environnement

## package.json

```
{
  "name": "cm5_exemples",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "dev": "DEBUG=app nodemon server.js"
  },
  "dependencies": {
    "dotenv": "^10.0.0"
  },
  "devDependencies": {
    "eslint": "^7.32.0",
    "nodemon": "^2.0.12"
  }
}
```

La définition du projet et dépendances, voir [docs.npmjs.com](https://docs.npmjs.com) dépendances.

## Avec le fichier package.json précédent :

```
npm install
# added 296 packages in 2s

npm run dev
# > cm4_exemples@1.0.0 dev
# > cross-env DEBUG=app nodemon server.mjs

# [nodemon] 2.0.19
# [nodemon] to restart at any time, enter `rs`
# [nodemon] watching path(s): *.*
# [nodemon] watching extensions: js,mjs,json
# [nodemon] starting `node server.mjs`
#   app Server listening at http://127.0.0.1:5000/ +0ms
```

# npm the package runner

<https://www.npmjs.com/package/npm>

npm permet d'exécuter des commandes locales au dossier (dans `node_modules/`) sans les installer globalement

Executes *command* either from a local `node_modules/.bin`, or from a central cache, installing any packages needed in order for *command* to run.

npm est implicitement utilisé par les scripts du `package.json`.

# Exemple

```
/tmp/npx> npx cowsay "Hello world"  
Need to install the following packages:  
  cowsay@1.5.0  
Ok to proceed? (y) y
```

```
< Hello world >
```

```
-----  
 \      ^  ^  
  \    (oo)\_____  
   (__) \       )\\/\  
        ||----w |  
        ||     ||
```

```
/tmp/npx> ll
```

```
/tmp/npx>
```

# Systemes de modules

## Plusieurs co-existent

### Common JS (CJS)

- historique
- `const maLib = require('laLib');`
- `module.exports = { ... };`
- par défaut pour `.js`

### EcmaScript (ESM) :

- standard JS
- `import maLib from 'laLib';`
- `export default { ... };`
- par défaut pour `.mjs`

💡 Préférer les modules ESM 💡

# Outillage et bonnes pratiques

## Plan

1. Environnement de développement
2. Gérer votre code et collaborer
3. Automatiser, déployer, et surveiller
4. Gérer vos dépendances de projet
5. Packager vos projets

# Bundlers

Beaucoup de fichiers (html, css, js, images, ...)

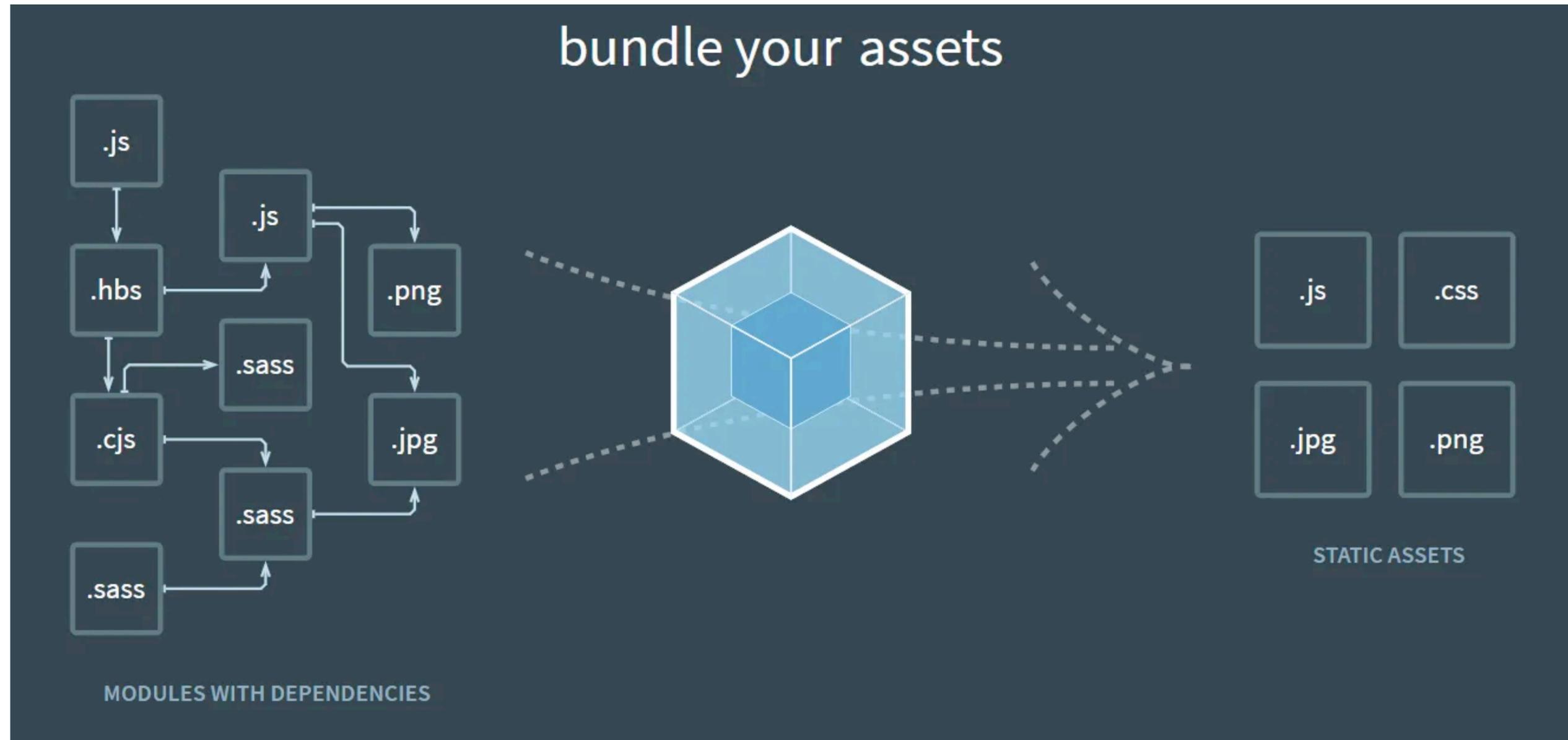
Il faut réduire le poids de nos applications !

On compresse tout grâce aux bundlers 🦹

[Webpack](#)

[Vite](#)

# Principe des bundlers



# Outillage et bonnes pratiques

## Plan

1. Environnement de développement
2. Gérer votre code et collaborer
3. Automatiser, déployer, et surveiller
4. Gérer vos dépendances de projet
5. Packager vos projets

# Bonnes pratiques

<https://12factor.net/>

## Configuration

- Fichier .env et cross-env (GitHub)

## Logging

- Exemple <https://getpino.io/>

## Reverse-proxy (a.k.a. front)

- <https://www.nginx.com/>

## Sécurité

- HTTPS, compte non privilégié