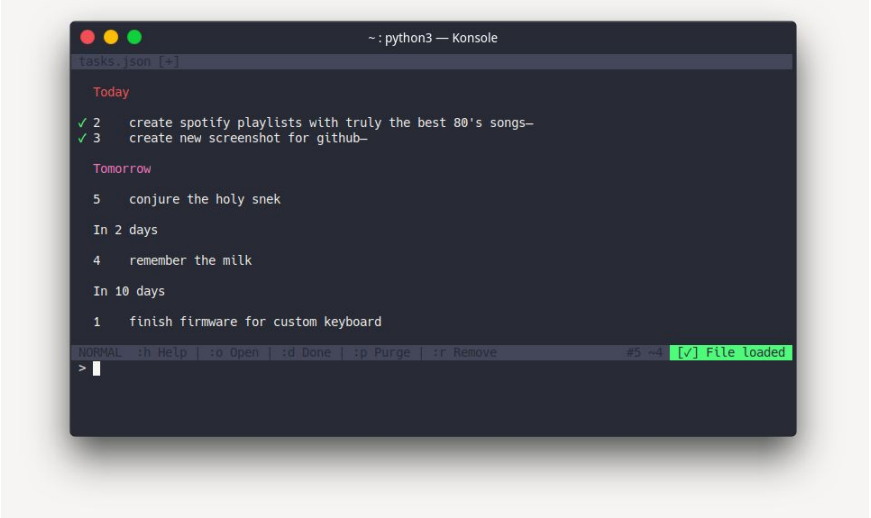
RustoDo – Gestionnaire de tâches CLI en Rust

Découverte des bases de rust

https://docs.google.com/presentation/d/1f7TnA_kq1s_8R2WV5B_pT-PXm3RFQBPE2wDWYxM5fUw/edit?usp=drive_link

Fonctionnalités principales

- add 'Faire les courses' → ajoute une tâche
- list → affiche toutes les tâches
- done 3 → marque une tâche comme terminée
- remove 3 → supprime une tâche
- Sauvegarde dans un fichier local (JSON, CSV, ou texte simple)



Phase 1 : Base du CLI

- Parsing manuel des arguments avec std::env::args
- Gestion en mémoire via un Vec<String>
- Squelette d'application CLI minimal

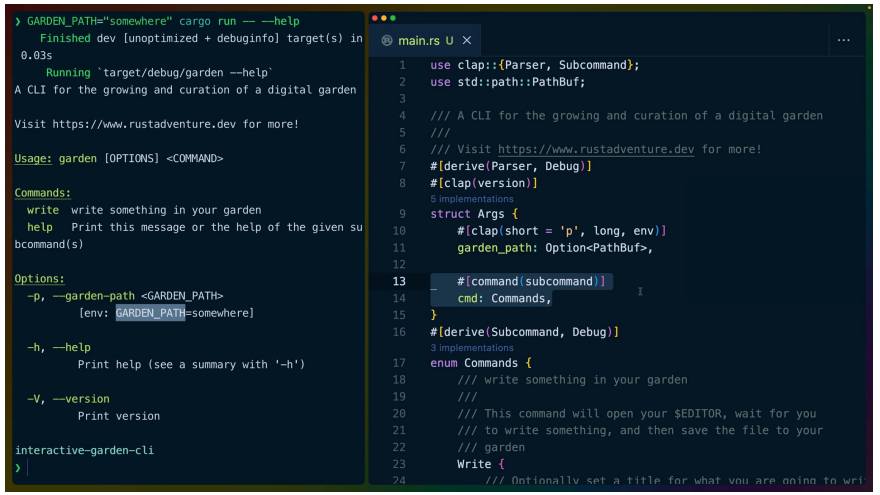
Phase 2 : Structuration & persistance

- Struct Task { id, title, done }
- Sauvegarde dans un fichier JSON via serde_json
- Transition vers une gestion persistante

```
MyType { a: 1, b: 2, c: 3 }

MyType { a: 1, b: 2,
         ..Default::default() }

mytype!(a: 1, b: 2)

MyTypeBuilder::default()
    .a(1).b(2).build()

new_mytype(1, 2)
```

Phase 3 : Ergonomie & robustesse

- Utilisation de clap pour une interface ergonomique
- Gestion d'erreurs avec Result, thiserror
- Tests unitaires avec cargo test



Phase 4 : Fonctions avancées

- Dates d'échéance avec chrono
- Sous-commandes : todo add, todo list...
- Colorisation de la sortie avec colored
- Export CSV / Markdown (bonus)

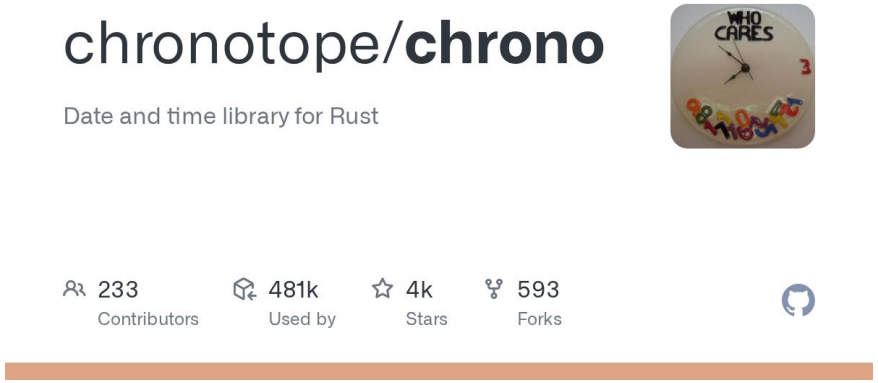


Image Source

Concepts Rust abordés

- Ownership & borrowing
- Structs et enums
- Gestion des erreurs (Result, Option)
- Modules, crates, I/O, fichiers
- Outils : cargo build/run/test

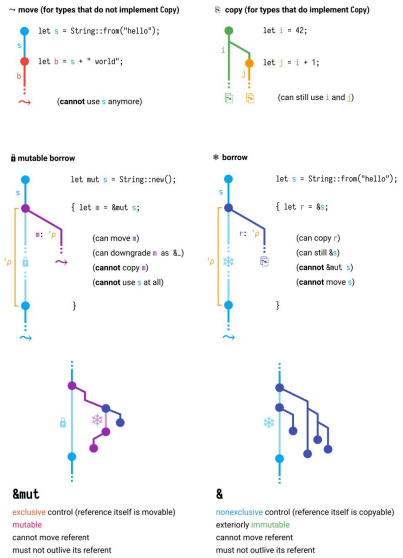


Image Source