C03 XML Prompting & Ancres Sémantiques

Conception Advanced

Structurer les prompts complexes pour des résultats reproductibles

Pourquoi XML fonctionne

Claude a été entraîné sur des corpus massifs qui incluent du XML. Quand vous structurez un prompt avec des balises, vous créez des délimiteurs sans ambiguïté qui séparent l'instruction, le contexte, le code de référence et les contraintes. La structure peut clarifier ces rôles, mais ne garantit ni le suivi des instructions ni la prévention des injections de prompt.

La structure XML est pertinente pour les tâches complexes ou réutilisables. Pour une correction d'une ligne, c'est du bruit inutile.

Structure de base

```
<instruction>
Extraire la logique de validation vers un module séparé
</instruction>

<context>
Service Express, Node.js 20, pas de dépendances externes
</context>

<code_example>
// Pattern existant à conserver
function validateEmail(email: string): boolean {
  return /^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(email)
}
</code_example>

<constraints>
- Ne pas modifier les signatures de fonctions publiques
- Garder la compatibilité avec les tests existants
</constraints>

<output>
Nouveau fichier src/validation/index.ts avec les exports
</output>
```

Catalog de tags réutilisables

Tag	Usage
`<instruction>`	La tâche principale
`<context>`	Stack, état du système, historique
`<code_example>`	Pattern de référence à suivre
`<constraints>`	Ce qu'il ne faut pas modifier
`<output>`	Format et contenu attendus
`<state>`	État actuel (bug, diff, logs)
`<task>` + `<subtask>`	Décomposition hiérarchique

Ancres sémantiques

Une ancre sémantique nomme une méthode ou un motif établi. Elle donne à Claude une référence concise, mais la tâche exige encore un contexte de projet et un contrôle.

```
Refactorise l'intégration de paiement avec Ports & Adapters.
Garde le code du prestataire hors du domaine.
Vérifie que les tests de paiement existants passent toujours.
```

« Ports & Adapters » nomme l'approche. Les deux phrases suivantes précisent son application à cette tâche et le contrôle attendu. Si Claude applique mal le terme, décrivez la frontière voulue plutôt que de répéter son nom. Voir la section du guide sur les ancres sémantiques.

Les commentaires tels que `// ANCHOR: auth-flow` sont des marqueurs de position dans le code. Ils peuvent désigner une zone du fichier, mais le mot « ancre » a ici un autre sens.

Quand utiliser XML

Cas	Utiliser XML ?
Requête simple, une ligne	Non
Bug à corriger dans une fonction	Non
Implémentation multi-fichiers	Oui
Prompt réutilisable dans une command	Oui
Review avec critères précis	Oui

Intégration avec CLAUDE.md

Vous pouvez définir les conventions XML du projet dans `CLAUDE.md` pour standardiser les prompts de l'équipe :

```
## Conventions de prompt XML
Tags projet : <api_design>, <accessibility>, <perf_budget>
Pour les features : toujours inclure <context> et <constraints>
```

Tous les membres de l'équipe utilisent ensuite le même vocabulaire de tags, ce qui rend les prompts partagés (dans `.claude/skills/`) cohérents et compréhensibles.