

Cahier du Participant

Claude Code 2.1.45 + Claude-Craft 7.13.0 — Exercices et notes

The Bearded Bear

Février 2026



Table des matières

Cahier du Participant - Formation Claude Code 2.1.45 + Claude-Craft 7.13.0	2
Informations	2
Objectifs de la formation	2
Jour 1 : Fondamentaux & Nouveaux Projets	2
Module 1 : Introduction à Claude Code 2.1.45	2
Module 2 : Le Framework Claude-Craft 7.13.0	3
Module 3 : Workflow de Développement	4
Module 4 : Nouveau Projet Symfony	5
Jour 2 : Projets Existants & Pratique Avancée	6
Module 5 : Intégrer sur un Projet Existant	6
Module 6 : Qualité et Sécurité	6
Module 7 : Agents Spécialisés, BMAD et Docker	7
Module 8 : Outils Avancés	8
Module 9 : Atelier Pratique	10
Auto-évaluation finale	10
Niveau de confiance (1 = Pas confiant, 5 = Très confiant)	10
Mes points forts	11
Mes axes d'amélioration	11
Plan d'action personnel	11
Semaine 1 (après la formation)	11
Mois 1	11
Questions à poser au formateur (suivi)	11
Ressources à garder	12
Liens utiles	12
Commandes essentielles	12
Contacts	13
Notes libres	13
Feedback formation	14
Ce que j'ai le plus apprécié	14
Ce qui pourrait être amélioré	14
Ma note globale (1-10) : ____	14

Cahier du Participant - Formation Claude Code 2.1.45 + Claude-Craft

7.13.0

Informations

Nom : _____

Date de formation : _____

Formateur : _____

Objectifs de la formation

À la fin de cette formation, vous serez capable de :

- ☐ Installer et configurer Claude Code 2.1.45
- ☐ Utiliser les nouvelles fonctionnalités (Plan Mode, Hooks, Background Tasks)
- ☐ Installer Claude-Craft 7.13.0 via npx (structure TCL)
- ☐ Suivre les workflows de développement (Quick, Standard, Enterprise)
- ☐ Charger et utiliser les skills et agents
- ☐ Auditer un projet existant et corriger les issues
- ☐ Appliquer les bonnes pratiques (TDD, SOLID, OWASP)
- ☐ Connaître BMAD v6, Ralph Wiggum et le QA Recette
- ☐ Comprendre Extended Thinking et MCP

Jour 1 : Fondamentaux & Nouveaux Projets

Module 1 : Introduction à Claude Code 2.1.45

Notes personnelles

Commandes de base

Commande	Description
/help	
/model	
/cost	
/exit	

Nouvelles commandes (2.1.45)

Commande	Description
/plan	
/tasks	
/hooks	
/keybindings	
/skills	
/compact	Compacter le contexte
/doctor	Diagnostiquer les problèmes
/mcp	Gérer les serveurs MCP
/config	Configuration

Exercice 1 : [OK] Completed / [KO] Needs review Difficultés rencontrées : _____

Module 2 : Le Framework Claude-Craft 7.13.0**TCL (Tiered Context Loading)**

Niveau	Fichiers	Tokens	Chargement
ALWAYS			
ON-DEMAND			
REFERENCE			

Économie de tokens : _____%

Structure TCL

```
.claude/  
├── ----- # Config minimale (~700 bytes)  
├── ----- # Liens rapides  
├── ----- # Triggers automatiques  
├── ----- # Documentation complète  
├── ----- # Best practices on-demand  
├── ----- # Agents IA  
└── ----- # Commandes slash
```

Installation npx

Commande interactive

npx -----

Commande directe

npx ----- --tech=----- --lang=-----

Exercice 2 : [OK] Completed / [KO] Needs review **Difficultés rencontrées :** _____

Module 3 : Workflow de Développement

Les 3 tracks

Track	Durée	Quand l'utiliser
_____	< 30 min	
_____	1-3 jours	
_____	> 1 semaine	

Commandes workflow

Commande	Ce qu'elle fait
/workflow:init	
/workflow:analyze	
/workflow:plan	
/workflow:implement	

Plan Mode + Workflow

1. Activer: /plan
2. Analyser: /workflow:analyze
3. Planifier: /workflow:plan
4. Désactiver: /plan off
5. Exécuter: /workflow:implement

Classification Plan Mode (v7.13.0)

Niveau	Quand	Exemples
MANDATORY		generate-*, /workflow:implement
RECOMMENDED		/workflow:plan, /workflow:design
CONDITIONAL		check-*, /workflow:analyze

Exercice 3 : [OK] Completed / [KO] Needs review Difficultés rencontrées : _____

Module 4 : Nouveau Projet Symfony**Architecture Clean**

```
src/
├── Domain/      → _____
├── Application/ → _____
├── Infrastructure/ → _____
└── UserInterface/ → _____
```

Commandes de génération

Commande	Ce qu'elle génère
/symfony:generate-feature	
/symfony:generate-entity	
/symfony:generate-crud	

Exercice 4 : [OK] Completed / [KO] Needs review Ce que j'ai créé : _____

Difficultés rencontrées : _____

Jour 2 : Projets Existants & Pratique Avancée

Module 5 : Intégrer sur un Projet Existant

Commandes d'audit

Commande	Ce qu'elle vérifie
/symfony:check-compliance	
/symfony:check-architecture	
/symfony:check-code-quality	
/symfony:check-security	
/symfony:check-testing	

Audit avec Plan Mode

```
/plan
/symfony:check-compliance
# Analyser les résultats sans modifier
/plan off
# Corriger les issues
```

Exercice 5 : [OK] Completed / [KO] Needs review Scores de mon audit : - Architecture : **/100** - Qualité : /100 - Sécurité : **/100** - Tests : %

Plan de remédiation (3 actions prioritaires) : 1. _____ 2. _____
3. _____

Module 6 : Qualité et Sécurité

Skills pour la qualité

Skill	Commande	Contenu
TDD/BDD	/testing	
OWASP	/security	
Git	/git-workflow	

Cycle TDD

_____ → _____ → _____

Conventional Commits

Type	Usage
feat	
fix	
refactor	
test	

Exercice 6 : [OK] Completed / [KO] Needs review Vulnérabilités corrigées : 1. _____

2. _____ 3. _____

Module 7 : Agents Spécialisés, BMAD et Docker

Skills vs Agents

Aspect	Skills	Agents
Invocation		
Chargement		
Usage		

Charger un skill

```
/_____ # Charger le skill testing
/_____ # Charger le skill security
```

Utiliser un agent

```
"Agis comme @_____ et revois ce code"
"Agis comme @_____ et écris les tests"
```

context.yaml

```
triggers:
  testing:
    keywords: [_____, _____, _____]
    auto_load: true
```


Exercice 7 : [OK] Completed / [KO] Needs review Issues identifiées par les agents : 1.

2.

3.

Module 8 : Outils Avancés

Hooks (13 événements)

Event	Déclencheur
PreToolUse	Avant l'exécution d'un outil
PostToolUse	Après le succès d'un outil
PostToolUseFailure	Après l'échec d'un outil
PermissionRequest	Lors d'une demande de permission
UserPromptSubmit	Quand l'utilisateur soumet un prompt
Stop	Fin de réponse Claude
SubagentStop	À l'arrêt d'un sous-agent
SubagentStart	Au lancement d'un sous-agent
Notification	Lors d'une notification
PreCompact	Avant la compaction du contexte
SessionStart	Au démarrage/reprise d'une session
SessionEnd	À la fin d'une session
Setup	Lors de l'initialisation

Configuration hook

```
{
  "hooks": {
    "PostToolUse": [{
      "matcher": "_____",
      "command": "_____ "
    }]
  }
}
```

Background Tasks

```
/tasks run "_____" # Lancer
/tasks list          # Lister
/tasks status _____ # Statut
/tasks stop _____ # Arrêter
```

Keybindings Mon fichier : `~/ .claude/keybindings.json`

Raccourci personnalisé : _____

Ralph Wiggum Boucle continue IA qui exécute Claude jusqu'à complétion de la tâche.

`/common:ralph-run "_____"`

Types de validateurs DoD :

Type	Description
command	
output_contains	
file_changed	
hook	
human	

QA Recette Tests d'acceptation automatisés via Claude in Chrome.

```
/qa:recette --scope=story --id=US-001
/qa:recette --scope=sprint --id=Sprint-3
/qa:recette --scope=story --id=US-001 --dry-run
```

Golden Rule : Un bug corrigé ne doit JAMAIS réapparaître.

Extended Thinking Mode de réflexion approfondie pour les tâches complexes.

```
# Activer le thinking étendu
claude --thinking
```

```
# Avec budget de tokens
claude --thinking-budget 10000
```

Quand l'utiliser : - Architecture complexe - Debug difficile - Décisions de conception

MCP (Model Context Protocol) Protocole pour connecter des serveurs de contexte externes.

```
/mcp                                # Lister les serveurs
/mcp add <nom> <commande>         # Ajouter un serveur
/mcp remove <nom>                  # Supprimer un serveur
```

Serveurs MCP utiles :

Serveur	Usage
filesystem	Accès fichiers
github	Intégration GitHub
postgres	Accès base de données

Module 9 : Atelier Pratique

Challenge d'équipe Mon binôme : _____

Notre feature : _____

Ce qu'on a réussi : - ☐ Plan Mode pour analyse - ☐ Skills chargés appropriés - ☐ Architecture Clean - ☐ Tests TDD - ☐ Hooks configurés - ☐ BMAD initialisé - ☐ Quality gate validé - ☐ QA Recette dry-run

Ce qu'on n'a pas eu le temps de faire : _____

Auto-évaluation finale

Niveau de confiance (1 = Pas confiant, 5 = Très confiant)

Compétence	1	2	3	4	5
Installation Claude Code 2.1.45					
Utilisation Plan Mode					
Installation Claude-Craft 7.13.0 (npx)					
Compréhension TCL					
Utilisation des workflows					
Chargement et utilisation des skills					
Configuration des hooks					
Génération de features					
Audit de projets					
Utilisation des agents					
TDD avec Claude					
BMAD v6					

Compétence	1	2	3	4	5
Ralph Wiggum					
QA Recette					
Extended Thinking					
MCP					

Mes points forts

1. _____
2. _____
3. _____

Mes axes d'amélioration

1. _____
2. _____
3. _____

Plan d'action personnel**Semaine 1 (après la formation)**

- ☐ Installer Claude-Craft 7.13.0 sur mon projet
- ☐ Configurer les hooks de base
- ☐ _____

Mois 1

- ☐ Maîtriser le Plan Mode
- ☐ Créer un skill custom pour mon projet
- ☐ _____

Questions à poser au formateur (suivi)

1. _____
2. _____

3. _____

Ressources à garder

Liens utiles

- Documentation Claude Code : <https://docs.anthropic.com/claude-code>
- Claude-Craft GitHub : <https://github.com/thebeardedcto/claude-craft>
- Claude-Craft NPM : <https://npmjs.com/@the-bearded-bear/claude-craft>
- Mon compte Anthropic : _____

Commandes essentielles

Installation Claude-Craft

```
npx @the-bearded-bear/claude-craft install
```

Plan Mode

```
/plan
```

```
/plan off
```

Skills

```
/testing
```

```
/security
```

```
/skills
```

Hooks

```
/hooks
```

Background Tasks

```
/tasks list
```

```
/tasks run "..."
```

Nouvelles commandes

```
/compact
```

```
/doctor
```

```
/mcp
```

BMAD v6

```
/workflow:init
```

```
/workflow:status
```

QA Recette

```
/qa:recette --scope=story --id=US-001
```

```
/qa:recette --dry-run
```

Extended Thinking

claude --thinking

claude --thinking-budget 10000

Contacts

- Formateur : _____
- Support : _____
- Collègue référent : _____

Notes libres

Feedback formation

Ce que j'ai le plus apprécié

Ce qui pourrait être amélioré

Ma note globale (1-10) : ____

Merci pour votre participation !

Formation Claude Code 2.1.45 + Claude-Craft 7.13.0 Version 3.0.0 - 2026 The Bearded CTO