

Cheat Sheet — Claude-Craft

Claude-Craft 7.13.0 — Framework et outils

The Bearded Bear

Février 2026



Cheat Sheet : Claude-Craft 7.13.0

Installation (NPX)

```
# Installation interactive
npx @the-bearded-bear/claude-craft install

# Installation directe
npx @the-bearded-bear/claude-craft install ~/projet \
  --tech=symfony --lang=fr

# Technologies : symfony, laravel, react, angular,
#   vuejs, flutter, reactnative, python, php, csharp
# Langues : fr, en, es, de, pt
```

TCL (Tiered Context Loading)

Niveau	Fichiers	Chargement
ALWAYS	CLAUDE.md, INDEX.md	Automatique
ON-DEMAND	skills/*.md	Via /skill
REFERENCE	references/**/*.*.md	Via @chemin

Version	Tokens auto	Economie
v3.x	~70,000	-
v7.x	~3,500	95%

Structure TCL

```
.claude/
├── CLAUDE.md           # Config minimale
├── INDEX.md            # Liens rapides
├── references/         # Doc complete
│   ├── base/          # Principes universels
│   └── <tech>/        # Specifique techno
├── skills/             # Best practices
├── agents/            # Agents IA
└── commands/          # Commandes slash
```

Skills (36)

```
/testing      # TDD/BDD
/security     # OWASP
/git-workflow # Git best practices
/documentation # Doc standards
/solid-principles # SOLID
/kiss-dry-yagni # Simplicite
```

Plan Mode Classification

Chaque commande inclut une guidance Plan Mode integree :

Niveau	Description	Exemples
MANDATORY	Plan mode auto avant execution	generate-*, /workflow:implement, /qa:tdd
RECOMMENDED	Recommande pour scope complexe	/workflow:plan, /workflow:design
CONDITIONAL	Auto si scope multi-modules	check-*, /workflow:analyze

Commandes (160 across 20 namespaces)

Common (13)

Commande	Description
/common:pre-commit-check	Valider avant commit
/common:ralph-run	Boucle IA continue
/common:setup-project-context	Configurer contexte
/common:add-technology	Ajouter stack
/common:architecture-decision	Documenter ADR
/common:daily-standup	Resume standup
/common:generate-changelog	Generer changelog
/common:pre-merge-check	Validation pre-merge
/common:release-checklist	Preparation release
/common:research-context7	Recherche Context7
/common:setup-ci	Setup CI pipeline
/common:sub-agents-patterns	Guide patterns agents
/common:init	Bootstrap structure

Workflow (9)

Commande	Description
/workflow:init	Initialiser (auto-detect track)
/workflow:analyze	Recherche, exploration
/workflow:plan	PRD, backlog
/workflow:design	Tech spec, architecture
/workflow:implement	Sprint de dev
/workflow:status	Afficher progression
/workflow:start	Demarrer sprint
/workflow:review	Sprint review
/workflow:retro	Retrospective

Team (4)

Commande	Description
/team:audit	Audit multi-tech parallele
/team:sprint	Sprint parallele
/team:security	Revue securite parallele
/team:delivery	Cycle complet sprint

QA (6)

Commande	Description
/qa:recette	Tests acceptance automatises
/qa:fix	Corriger bugs recette
/qa:status	Statut session
/qa:regression	Tests de regression
/qa:report	Rapport de recette
/qa:tdd	Correction TDD/BDD

UIUX (7)

Commande	Description
/uiux:audit	Audit UI/UX
/uiux:ally-audit	Audit WCAG
/uiux:ally-component	Composant accessible
/uiux:component-spec	Spec composant UI
/uiux:orchestrator	Orchestration UI/UX
/uiux:user-flow	Parcours utilisateur
/uiux:design-tokens	Design tokens

Sprint (5)

Commande	Description
/sprint:next-story	Prochaine story prete
/sprint:transition	Transition statut
/sprint:status	Metriques sprint
/sprint:auto-route	Auto-routage stories
/sprint:dev	Dev TDD sprint

Gate (6)

Commande	Description
/gate:validate-prd	Quality gate PRD (>=80%)
/gate:validate-story	Validation Story DoD

Commande	Description
/gate:validate-backlog	Validation backlog
/gate:validate-techspec	Validation tech spec
/gate:validate-sprint	Validation sprint
/gate:report	Rapport qualite

Project (22)

Commande	Description
/project:run-sprint	Sprint complet
/project:run-epic	Epic complet
/project:run-queue	Executer queue
/project:batch-status	Statut batch
+18 commandes	Gestion backlog, epics, stories

Docker (5)

Commande	Description
/docker:compose-setup	Docker-compose
/docker:architecture	Architecture Docker
/docker:debug	Diagnostic containers
/docker:cicd-pipeline	Pipeline CI/CD
/docker:optimize	Optimiser Docker

Technos (6-10 par stack)

Chaque techno a ses commandes :

- check-compliance, check-architecture, check-code-quality, check-security, check-testing
- generate-* (feature, component, crud...)

Agents (33)

Common (12)

Agent	Expertise
@api-designer	REST/GraphQL API
@database-architect	Base de donnees
@devops-engineer	CI/CD, Docker
@performance-auditor	Performance
@refactoring-specialist	Refactoring
@tdd-coach	TDD
@uiux-orchestrator	UI/UX coordination
@ui-designer	Design systems

Agent	Expertise
@ux-ergonome	Ergonomie cognitive
@accessibility-expert	WCAG 2.2 AAA
@research-assistant	Recherche technique
@alph-conductor	Orchestration boucle

Tech Reviewers (10)

@symfony-reviewer, @flutter-reviewer, @react-reviewer, @python-reviewer, @angular-reviewer, @laravel-reviewer, @vuejs-reviewer, @reactnative-reviewer, @csharp-reviewer, @php-reviewer

Docker (5)

@docker-dockerfile, @docker-compose, @docker-debug, @docker-cicd, @docker-architect

Coolify (4)

@coolify-architect, @coolify-deployment, @coolify-debug, @coolify-monitoring

Project (2)

@product-owner (CSPO), @tech-lead

Selection modele par agent

Type	Modele	Raison
Reviewers, Auditors	haiku	Economique
Engineers, Architects	sonnet	Puissant
Orchestrators	sonnet	Coordination

Technologies (10)

Stack	Version 2026	Focus
symfony	8.0 / PHP 8.5	Clean + DDD
laravel	12 / PHP 8.5	Eloquent + API
react	19 / TS 5.7	Hooks + State
angular	19.x / TS 5.7	Standalone
vuejs	3.5+ / TS 5.7	Composition API
flutter	3.38 / Dart 3.10	BLoC/Riverpod
reactnative	0.76+	Native Modules
python	3.13+	FastAPI
php	8.5	PSR Standards
csharp	.NET 10 / C# 14	CQRS + MediatR

Stack	Version 2026	Focus
-------	--------------	-------

BMAD v6 - Gestion de Projet

```
/workflow:init           # Initialiser
/workflow:status         # Statut projet
/sprint:next-story       # Prochaine story
/sprint:transition US-1 done # Transition
/gate:validate-prd       # PRD (>=80%)
/gate:validate-story     # Story DoD
/project:run-sprint      # Sprint complet
```

Quality Gates : PRD >=80%, Tech Spec >=90%, INVEST 6/6, Sprint Ready 100%, Story DoD 100%

Status : backlog -> ready-for-dev -> in-progress -> review -> done | blocked

Ralph Wiggum - Boucle IA Continue

```
/common:ralph-run "Implement feature X"
```

DoD Validators : command, output_contains, file_changed, hook, human

QA Recette - Tests Acceptance

```
/qa:recette --scope=story --id=US-001
/qa:recette --scope=sprint --id=Sprint-3
/qa:recette --dry-run
/qa:status
/qa:report
/qa:fix --session=REC-xxx
/qa:regression
```

Golden Rule : A fixed bug should NEVER reappear.

Architecture Clean

```
src/
├── Domain/           # 0 dependance
│   ├── Model/        # Entites, VO
│   └── Repository/    # Interfaces
├── Application/       # Use Cases
│   ├── Command/       # Write (CQRS)
│   └── Query/         # Read (CQRS)
├── Infrastructure/    # Implementations
└── UserInterface/    # API/Controllers
```

TDD avec Claude

RED -> GREEN -> REFACTOR

```
/testing
@tdd-coach "Tests pour OrderService"
```

Conventional Commits

type(scope): description

feat(order): add discount calculation

fix(auth): correct token expiration

Types : feat, fix, refactor, test, docs, chore, perf, build, ci, style

Comparaison v3.x vs v7.x

Aspect	v3.x	v7.x
Installation	git clone + make	npx
Tokens auto	~70,000	~3,500
Structure	rules/ + skills/	TCL (3 niveaux)
Technologies	5	10
Agents	~10	33
Commands	~30	160
Namespaces	-	20
Gestion projet	-	BMAD v6
QA Browser	-	QA Recette

En bref

33 agents	160 commands
10 stacks	5 langues
20 namespaces	36 skills

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