CHEAT SHEET
COMMANDES IA NEOVIM

COMMANDES PERSONNALISÉES

`:Ask <question>`	Question sans contexte de code
`:AskBuffer`	Envoyer tout le fichier + question
`:Generate <demande>`	INSÈRE code directement dans fichier
`:Translate <lang>`	Traduire code (défaut : C)

EXEMPLES :ASK

- `:Ask qu'est-ce qu'une closure en C++11`
- `:Ask différence entre unique_ptr et shared_ptr`
- `:Ask comment gérer les threads en C++11`
- `:Ask explique le principe RAII`

EXEMPLES :TRANSLATE

`:Translate C++11`	Python → C++11
`:Translate Python`	C++ → Python
`:Translate Rust`	Vers Rust
`:Translate Java`	Vers Java

CODECOMPANION NATIF

`:CodeCompanion <prompt>`	Prompt direct inline
`:CodeCompanionChat Toggle`	Ouvrir/fermer chat
`:CodeCompanionChat Add`	Ajouter sélection au chat
`:CodeCompanionChat Submit`	Envoyer message
`:CodeCompanionChat Clear`	Effacer historique
`:CodeCompanionActions`	Menu actions (Explain, Fix...)

RACCOURCIS IA

`Espace` + `c` + `g`	Générer code dans fichier (`:Generate`)
`Espace` + `c` + `c`	Toggle chat (normal/visuel)
`Espace` + `c` + `a`	Menu actions sur sélection (insertion auto !)
`Ctrl` + `s`	Envoyer message (mode INSERT dans chat)
`Ctrl` + `Enter`	Envoyer (ne marche pas dans tous les terminaux)
`Espace` + `c` + `s`	Envoyer (mode NORMAL)

RACCOURCIS MAKE

`F2`	Compiler projet (pcmakeProjectCompile)
`F3`	Sauvegarder projet (pcmakeProjectSave)
`Espace` + `m` + `m`	Compiler avec make
`Espace` + `m` + `c`	make clean
`Espace` + `m` + `r`	Compiler ET exécuter
`Espace` + `m` + `t`	make ¡target personnalisé¿
`Espace` + `m` + `e`	Ouvrir liste erreurs
`Espace` + `m` + `n`	Erreur suivante
`Espace` + `m` + `p`	Erreur précédente

WORKFLOW TYPIQUE

1. **Insérer code automatiquement :** LE PLUS SIMPLE
   `V` → `Espace c a` → Custom → écris classe X
2. **Générer code (avec copie) :**
   `:Generate classe Vector3D en C++11`
3. **Question simple :**
   `:Ask comment faire X en C++11`
4. **Analyser fichier :**
   `:AskBuffer`
   `i` → taper question → `Ctrl+s`
5. **Traduire code :**
   `:Translate C++11` → `Ctrl+s`
6. **Compiler :**
   `Espace m m` → `Espace m n` (erreurs) → `Espace m r` (exécuter)
7. **Chat interactif :**
   `Espace c c` → `i` → question → `Ctrl+s`

MENU ACTIONS (Espace + c + a)

Explain	Expliquer le code
Fix	Corriger bugs
Optimize	Optimiser performance
Document	Ajouter documentation
Refactor	Refactoriser
Test	Générer tests unitaires
Custom	Demande personnalisée

BONNES PRATIQUES

Préciser "C++11" dans les questions

Utiliser `:AskBuffer` pour analyse complète

Utiliser `:Translate` pour conversion code

Utiliser `Espace+ca` pour actions rapides

`Ctrl+s` pour envoyer dans le chat (pas Enter seul)

À ÉVITER

Questions ambiguës sans langage

Supposer que l'IA voit votre code (`:Ask`)

Oublier 'i' en mode INSERT dans le chat

Utiliser Enter seul (utilisez `Ctrl+s`)

COMPARAISON COMMANDES

lightgray **Commande**	**Contexte**	**Insertion**	**Rapidité**	**Usage**
`:Generate`		Directe		Génération
`:Ask`		Chat		Questions
`:AskBuffer`		Chat		Analyse
`:Translate`		Chat		Traduction
`Espace+cc`	Variable	Chat		Chat libre
`Espace+ca`		Remplace		Actions

EXEMPLES AVANCÉS

Refactoring :

`:AskBuffer`
Refactorise en C++11 : smart pointers, RAII, lambdas

Debug :

`ggVG` → `Espace c a` → Fix

Documentation :

`:AskBuffer`
Ajoute documentation Doxygen complète

Tests :

`v%` → `Espace c a` → Test

Optimisation :

`:AskBuffer`
Optimise pour performance avec benchmark

Comparaison :

`:Ask compare std::vector vs std::list en C++11`

CONFIGURATION

Fichier config :	`~/.config/nvim/init.lua`
System prompt :	init.lua ligne 93
Modèle IA :	init.lua ligne 78
Clé API :	`~/.OPENROUTER_API_KEY`

Recharger config :	`:source ~/.config/nvim/init.lua`
Vérifier santé :	`:checkhealth codecompanion`
Vérifier clé :	`:lua print(vim.env.OPENROUTER_API_KEY)`

DÉPANNAGE

Pas de réponse	→ `echo $OPENROUTER_API_KEY`
Mauvais langage	→ Préciser "C++11" explicitement
Chat ne marche pas	→ `:Lazy sync` puis relancer
Commande inconnue	→ `:source ~/.config/nvim/init.lua`

DOCUMENTATION COMPLÈTE

Référence IA :	`cat ~/IA/NEOVIM/REFERENCE_IA.md`
Contexte :	`cat ~/IA/NEOVIM/GUIDE_CONTEXTE.md`
Langages :	`cat ~/IA/NEOVIM/GUIDE_LANGAGES.md`
Personnalisation :	`cat ~/IA/NEOVIM/GUIDE_PERSONNALISATION.md`
Index :	`cat ~/IA/NEOVIM/INDEX.md`

MÉMO

`:Ask` (question) | `:AskBuffer` (analyse) | `:Translate` (→)