

Git Commit Cheatsheet

Git Commit Message Cheatsheet

This cheatsheet expands the list of commit types with **simple explanations, easy-to-remember sample messages, and real-world use cases.**

```
// example  
git commit -m "feat: add search bar"
```

Type	* Description**	Example Message	When to Use (Use Case)
feat	Adding something new	feat: add search bar	New feature like button, page, API endpoint, or option.
fix	Solving a problem or bug	fix: stop crash on save	When the app errors, crashes, or shows wrong data.
docs	Writing or updating explanations	docs: update setup guide	README changes, tutorials, inline comments.
style	Code look/format changes (not functionality)	style: clean up spaces	Adjusting indentation, renaming files for consistency.
refactor	Rewriting code for clarity or efficiency without changing results	refactor: simplify login check	Improving messy or hard-to-read code.
test	Writing or adjusting tests	test: add profile test	New or updated unit/integration tests.
chore	Routine updates or maintenance	chore: update dependencies	Dependency upgrades, config edits, background cleanup.
perf	Making things faster or lighter	perf: speed up dashboard load	Optimizing queries, reducing memory usage.
build	Changing how project is built or packaged	build: add production	Webpack, Babel, Gradle, Dockerfile

Type	* Description**	Example Message	When to Use (Use Case)
		config	updates.
ci	Automation (continuous integration/deployment)	ci: add test workflow	GitHub Actions, Travis, Jenkins scripts.
revert	Undoing a change that broke things	revert: remove login patch	Rollback faulty commits.
BREAKING CHANGE	A change that forces others to adjust their code	feat: switch to UUID BREAKING CHANGE: IDs updated	Incompatible API changes, removed features.
merge	Merging branches	merge: combine dev into main	When you pull changes from another branch.
hotfix	Urgent fix in production	hotfix: patch payment error	Emergency bug fix for deployed app.
security	Fixing vulnerabilities	security: fix SQL injection	Password leaks, injection flaws, insecure config.
ui	Changes to look and feel (not logic)	ui: adjust button color	CSS, themes, or design tweaks.
ux	Improving user experience	ux: simplify signup flow	Shortening steps, making navigation clearer.
env	Environment configuration changes	env: add staging config	Adding .env values, server setup, secrets.
db	Database changes	db: add users table	New tables, migrations, schema changes.
deps	Dependency changes	deps: bump lodash to v4.17.21	Adding, updating, or removing libraries.
config	Project settings or configuration	config: update ESLint rules	Changing settings files like .eslintrc , .gitignore .
infra	Infrastructure changes	infra: update docker-compose	Cloud, servers, network, or containers.
release	Tagging or preparing a release	release: version 1.2.0	New app version or release notes.
analytics	Adding or modifying tracking/metrics	analytics: add Google	Adding or editing logs, events, metrics.

Type	* Description**	Example Message	When to Use (Use Case)
		Analytics	
monitor	Changes to monitoring/alerts	monitor: add error alert rule	Observability tools, logging, alarms.
i18n	Internationalization (translations, languages)	i18n: add Spanish translations	Adding multi-language support.
a11y	Accessibility improvements	a11y: improve contrast on buttons	Screen reader labels, keyboard navigation, ARIA fixes.
content	Updating static content	content: update homepage text	Changing wording, labels, copywriting.
prototype	Experimental changes or temporary proof-of-concept	prototype: add draft payment flow	Trying out an idea before formalizing.
spike	Research or exploration commit	spike: test new API library	Trying out tools, not final code.
wip	Work in progress (not finished, avoid pushing to main)	wip: redesign settings page	Early commits still under development.

Practical Tips

- Use **present tense** (“add feature”, not “added feature”).
- Keep it short, 50 characters or less for the title.
- Add details in the body if needed, especially for complex commits.
- Remember: **commits are history notes**—make them clear enough that future you (or your teammate) won’t be confused.