

Prompt: skills

Sottoinsieme dei prompt del libro filtrati per tema: skills Capitoli inclusi: Capitolo 4, Capitolo 8, Capitolo 11

Prompt 1

```
# Struttura consigliata per un progetto team
.claude/
├── skills/
│   ├── deploy/          # /deploy - esegui il deploy
│   │   └── SKILL.md
│   ├── changelog/       # /changelog - genera il changelog
│   │   └── SKILL.md
│   ├── review-pr/       # /review-pr - revisiona la PR
│   │   └── SKILL.md
└── CLAUDE.md
```

Prompt 2

```
# Template description ben strutturata
---
description: [Cosa fa in una frase].
  Usa quando [trigger phrases, frasi naturali dell'utente].
  NON usare quando [distingui da casi simili].
---
```

Prompt 3

```
# Skill che solo tu puoi invocare
---
description: Deploy dell'applicazione in produzione
disable-model-invocation: true
---
```

Prompt 4

```
/create-api-endpoint users GET "Retrieve a paginated list of users"
```

Prompt 5

```
# Struttura consigliata per skill complessa
mkdir -p ~/.claude/skills/security-audit/{examples,scripts}
touch ~/.claude/skills/security-audit/SKILL.md
touch ~/.claude/skills/security-audit/checklist.md
touch ~/.claude/skills/security-audit/examples/{good,bad}-example.md
touch ~/.claude/skills/security-audit/scripts/scan.sh
```

Prompt 6

```
# Installa tutte e tre le skill di esempio
mkdir -p ~/.claude/skills/{commit-it,docfunc}
mkdir -p .claude/skills/sec-check

# Poi crea i SKILL.md con il contenuto degli esempi sopra
```

Prompt 7

```
# Aggiungi una nota nel CLAUDE.md del progetto
## Skill disponibili

Questo progetto include skill Claude Code in `~/.claude/skills/`:

- `/commit-it` - genera commit message in italiano (conventional commits)
- `/sec-check` - security check rapido prima di committare
- `/changelog` - genera changelog da commit
- `/review-interno` - code review strutturata del branch

Prima di usarle, accetta il workspace trust dialog che Claude Code mostra.
```

Prompt 8

```
# Review standard – controlla tutto
/simplify

# Review mirata – solo efficienza memoria
/simplify focus on memory efficiency

# Review pre-PR – qualità e leggibilità
/simplify focus on readability and naming conventions
```

Prompt 9

```
# Migrazione framework
/batch migrate src/ from Solid to React – preserve all existing tests and behavior

# Aggiornamento API
/batch update all uses of fetch() to use our new httpClient wrapper in src/api/

# Aggiunta logging
/batch add structured JSON logging to all service functions in src/services/

# Conversione TypeScript
/batch convert all .js files in src/ to .ts with strict mode – add types based on usage
```

Prompt 10

```
# Analisi generica
/debug

# Problema specifico con descrizione
/debug il subagent code-reviewer si avvia ma non restituisce output

# Problema di performance
/debug le risposte sono molto lente negli ultimi 10 minuti
```

Prompt 11

```
# Aspetta che un deploy finisca (controlla ogni 5 minuti)
/loop 5m check if the deployment to production succeeded – look at the deployment logs

# Monitor test suite durante sviluppo
/loop 2m run the test suite and report any new failures

# Senza intervallo – Claude decide il ritmo
/loop watch the build output and notify me when it completes or fails

# Usa .claude/loop.md come configurazione
/loop
```

Prompt 12

```
# Carica reference API (Claude rileva il linguaggio automaticamente)
/claude-api

# Migra il codice al modello più recente
/claude-api migrate

# Crea un nuovo Managed Agent guidato
/claude-api managed-agents-onboard
```

Prompt 13

```
# Esegui l'ottimizzazione dei permessi
/fewer-permission-prompts
```

Prompt 14

```
/brainstorming
Ho un negozio online di prodotti artigianali.
Voglio aumentare le vendite del 20% entro tre mesi.
Aiutami a esplorare le strategie possibili.
```

Prompt 15

/systematic-debugging

Il mio form di login funziona sul mio computer ma non su quello del cliente.
L'errore nel log è "Cannot read property 'userId' of undefined".
Ho già verificato le credenziali database – sono corrette.

Prompt 16

/content-calendar

Canali: Instagram (3/settimana), LinkedIn (2/settimana)
Periodo: giugno 2026
Topic: lancio del mio corso di fotografia online (297€)
Tono Instagram: ispirazionale e personale. Tono LinkedIn: professionale.

Prompt 17

/conversion-copywriting

Tipo: email di benvenuto (primo messaggio dopo iscrizione)
Prodotto: corso online fotografia, 297€, 8 settimane
Audience: hobbisti 35-55 anni che vogliono migliorare ma non diventare professionisti
Obiettivo: portarli alla pagina di acquisto entro 48 ore
Framework: PAS

Prompt 18

/skill-creator

Voglio creare una skill per generare release notes.
Ogni volta che faccio un deploy, voglio una nota strutturata con:
- Versione rilasciata
- Lista delle modifiche (dal git log)
- Eventuale nota per i clienti
Il file deve essere salvato in docs/releases/ con la data nel nome.

Prompt 19

```
---
name: safe-git
description: Git operations without permission prompts. Use when working with git.
disable-model-invocation: true
allowed-tools: Bash(git status) Bash(git diff *) Bash(git log *) Bash(git add *) Bash(git commit *)
---
```

Esegui operazioni git in modo sicuro:

- git status e git diff: sempre permessi, non chiedere
- git add e git commit: chiedi conferma prima
- git push: MAI senza conferma esplicita dell'utente

Prompt 20

/skill-creator

Voglio creare una skill che ogni volta che la invoco:

- Analizza i file nella cartella corrente
- Mi dice quanti file per tipo ci sono
- Suggerisce come organizzarli in sottocartelle
- Mi chiede conferma prima di fare qualsiasi spostamento

Prompt 21

```
---
name: organizza-file
description: Analizza i file nella cartella corrente, raggruppa per tipo, e propone
una struttura di organizzazione con spostamento confermato. Usare quando l'utente
chiede di organizzare file, riordinare una cartella, o capire cosa c'è in una
directory e come strutturarla.
when_to_use: >
  Trigger: "organizza questa cartella", "riordina i file", "cosa c'è in Downloads",
"crea una struttura", "dividi i file per tipo"
disable-model-invocation: false
allowed-tools: Read Bash(ls *) Bash(find *) Bash(mv *) Bash(mkdir *)
---
```

Cosa fare

1. Elenca tutti i file nella cartella corrente con `ls -la`
2. Raggruppa mentalmente per estensione:
 - Immagini: .jpg, .jpeg, .png, .gif, .webp, .svg, .heic
 - Documenti: .pdf, .docx, .doc, .xlsx, .pptx, .txt, .md
 - Video: .mp4, .mov, .avi, .mkv, .webm
 - Audio: .mp3, .wav, .aac, .flac, .ogg
 - Archivi: .zip, .tar, .gz, .rar, .7z
 - Altro: tutto il resto
3. Mostra la tabella riepilogativa: Tipo | Count | Dimensione totale
4. Proponi la struttura di cartelle (Immagini/, Documenti/, Video/, ecc.)
5. Chiedi conferma: "Vuoi che sposti i file in queste cartelle? Digita 's' per confermare o 'n' per annullare."
6. Se confermato: esegui gli spostamenti uno per volta mostrando ogni operazione
7. Al termine: mostra il riepilogo degli spostamenti eseguiti

Regole

- NON spostare file nascosti (che iniziano con .)
- NON spostare file aperti o in uso (estensioni .lock, .tmp)
- Mostrare sempre il piano completo PRIMA di eseguire
- In caso di conflitto di nomi (file con stesso nome già esistente), aggiungere suffisso numerico: documento.pdf → documento-2.pdf
- Se la cartella è vuota, comunicarlo e terminare

Prompt 22

/skill-creator

Testa la skill organizza-file con questi scenari:

1. Una cartella Downloads con 50 file misti (il caso normale)
2. Una cartella vuota (edge case)
3. Una cartella con solo file nascosti (edge case)
4. Una cartella dove tutti i file hanno già la struttura corretta (idempotenza)

Prompt 23

/remotion-powered-by-truant

Reel 9:16, durata 15 secondi

Testo: "La creatività è intelligenza che si diverte." – Albert Einstein

Stile: dark mode, font bold, animazione testo reveal

Prompt 24

name: analisi-legale

description: >

Analizza documenti legali e identifica clausole potenzialmente problematiche.

Usa quando l'utente chiede di rivedere un contratto, analizzare termini e condizioni, o identificare rischi in documenti legali.

when_to_use: >

Trigger: "rivedi questo contratto", "analizza questi termini", "ci sono problemi in questo documento", "cosa significa questa clausola", "è pericoloso questo accordo?"

argument-hint: [percorso-file-o-testo]

disable-model-invocation: false

allowed-tools: Read Bash(cat *) Bash(wc *)

model: opus

effort: high

Analisi documento legale

Esegui questi passaggi nell'ordine:

1. Leggi l'intero documento prima di commentare
2. Identifica il tipo di documento (contratto, NDA, EULA, altro)
3. Elenca le clausole potenzialmente problematiche con:
 - Citazione testuale della clausola
 - Perché potrebbe essere problematica
 - Rischio (basso/medio/alto)
4. Fornisci una valutazione generale (1-5 stelle per semplicità)
5. Raccomanda i punti da negoziare prima di firmare

Avvertenza

Sei un assistente per la comprensione dei documenti, non un avvocato.

Ricorda sempre all'utente che per decisioni legali importanti deve consultare un professionista.

Prompt 25

```
---
name: status-progetto
description: Mostra lo stato completo del progetto Git corrente. Usa quando
  l'utente chiede lo stato, le modifiche in sospeso, o cosa è cambiato.
---

## Stato corrente del progetto

**Branch e ultimo commit:**
!`git log --oneline -5`

**Modifiche in sospeso:**
!`git status --short`

**Dimensione delle modifiche:**
!`git diff --stat HEAD`

## Istruzioni

Analizza questi dati e fornisci:
1. Un riassunto in 3 bullet di cosa è cambiato
2. Eventuali rischi (file critici, test mancanti, conflitti potenziali)
3. Il messaggio di commit suggerito per le modifiche in sospeso
```

Prompt 26

```
/organizza-file
(esegui nella cartella ~/Downloads reale)
```

Prompt 27

Analizza questo SKILL.md e aiutami a migliorarlo. Cerca:

1. Istruzioni ridondanti o contraddittorie
2. Sezioni che potrebbero andare in file separati (reference.md, examples/)
3. La description: potrebbe essere più precisa per il triggering automatico?
4. Opportunità per usare dynamic context injection (!`comando`)
5. Regole che potrebbero essere semplificate

[incolla il contenuto del SKILL.md]

Poi scrivi la versione migliorata mantenendo tutto il comportamento utile, ma eliminando ridondanze e contraddizioni.

Prompt 28

```
# Crea la directory della skill nel progetto
mkdir -p .claude/skills/mia-skill

# Copia il SKILL.md
cp ~/temp/mia-skill/SKILL.md .claude/skills/mia-skill/

# Aggiungi al repository
git add .claude/skills/mia-skill/
git commit -m "feat: aggiungi skill mia-skill per automazione X"
```

Prompt 29

Leggi il file ~/.claude/skills/[nome-skill-sorgente]/SKILL.md e dimmi:

1. Quali sezioni sono riutilizzabili per [descrizione del tuo use case]?
2. Quali sezioni sono specifiche per l'uso originale e dovrei rimuovere?
3. Cosa dovrei aggiungere per adattarla a [tuo use case specifico]?
4. Ci sono problemi strutturali di cui devo essere consapevole?

Prompt 30

```
---
name: mio-fork-specializzato
description: >
```

```
Fork specializzato di [skill-sorgente] per [tuo use case specifico].
Usa quando [trigger specifici per il tuo contesto, diversi dall'originale].
# Basato su [skill-sorgente] di [autore originale]
# Versione originale: [link o riferimento]
# Modifiche principali:
# - Aggiunto: [cosa hai aggiunto]
# - Rimosso: [cosa hai tolto]
# - Modificato: [cosa hai cambiato]
---
```