

ChatGPT e le skill riutilizzabili: come si fa con Codex?

Maria Cattini | 07/04/2026 | Intelligenza Artificiale

Il problema che nessuno ti dice su ChatGPT

Ogni volta che apri una nuova sessione su [ChatGPT](#), il modello non ricorda nulla di te. Non sa come scrivi, non conosce il tuo settore, non ha idea che tu abbia già spiegato tre volte le stesse cose nelle sessioni precedenti. Ricominciare da capo è la norma — e diventa sfibrante.

Claude gestisce questo problema con le "skills": file di istruzione che carica automaticamente ogni volta che li attivi. ChatGPT non ha nulla di simile in modo nativo. Punto. Però c'è una via alternativa che funziona: [usare OpenAI Codex](#) come backend per costruire template di prompt strutturati e richiamabili, creando di fatto una libreria di competenze personali.

Vediamo come fare — passo dopo passo, senza promesse esagerate.

La domanda giusta non è "ChatGPT può fare quello che fa Claude?" La domanda è: "Posso costruirmi un sistema che mi faccia risparmiare tempo ogni settimana?" La risposta è sì.

Cos'è Codex e perché entra in gioco

OpenAI Codex è il modello ottimizzato per la comprensione e la generazione di codice. Nella pratica, oggi è accessibile tramite l'API di OpenAI oppure attraverso ChatGPT stesso quando attivi la modalità di ragionamento avanzato o interagisci via API con il modello code-davinci o i suoi successori.

Il punto non è usare Codex per scrivere software. Il punto è sfruttare la sua capacità di gestire istruzioni molto strutturate — quasi come dei file di configurazione — per costruire prompt che ChatGPT esegue con coerenza. Codex, in questo flusso, diventa il motore che interpreta e restituisce le istruzioni formattate. ChatGPT le riceve e le applica.

La differenza tra un prompt e uno skill template

Un prompt normale è una richiesta: "Riscrivi questo testo in modo più formale." Uno **skill template** è qualcosa di più profondo — è un documento di istruzione che definisce il contesto, le regole, i vincoli, il tono e il formato attesi, prima ancora che tu scriva la richiesta specifica.

Pensa a un istruttore di cucina. Non gli dici solo "fai una pasta": gli dai il manuale dello chef, le ricette standard, le regole della cucina. Quello è uno skill template.

Come configurare il sistema: guida passo dopo passo

Passo 1 — Definisci le aree di competenza che ti servono

Prima di scrivere anche solo una riga, chiediti: in quali attività ripetitive uso ChatGPT? Scrittura di

articoli? Ricerca e sintesi? Analisi di dati? Email professionali? Ogni area diventa uno skill separato.

Tieni la lista corta: tre o quattro skill ben costruiti valgono molto più di venti raffazzonati.

Passo 2 — Costruisci il template base per ogni skill

Ogni skill ha una struttura precisa. Ecco il formato consigliato:

```
## SKILL: [Nome della competenza]

**Contesto:** [Chi sei, in che settore lavori, per quale scopo usi questo skill]

**Regole di output:**
- Formato richiesto (es. paragrafi narrativi, liste numerate, markdown)
- Lunghezza indicativa
- Tono (formale, diretto, tecnico, divulgativo)
- Cosa evitare (es. frasi generiche, conclusioni prevedibili)

**Variabili di input:**
- [TESTO_DA_ELABORARE]: il contenuto su cui lavorare
- [OBIETTIVO]: l'intenzione specifica per questa sessione

**Esempio di chiamata:**
"Usa lo skill SCRITTURA_SEO. TESTO_DA_ELABORARE: [incolla qui]. OBIETTIVO: ottimizzare per la keyword principale."
```

Passo 3 — Archivia gli skill in un file di testo accessibile

Salva ogni skill in un documento di testo semplice — Notion, Obsidian, anche una cartella su Google Drive. L'importante è che tu possa aprirlo, copiarlo e incollarlo all'inizio di una nuova sessione ChatGPT in meno di dieci secondi.

Non serve nessuna integrazione tecnica per iniziare. La velocità di accesso è l'unico criterio che conta.

Passo 4 — Usa l'API con Codex per automatizzare il caricamento

Se vuoi andare oltre il copia-incolla manuale, puoi usare l'API di OpenAI per passare il tuo skill template come **system prompt** automatico. In questo modo, ogni volta che chiami il modello tramite script o interfaccia personalizzata, lo skill viene già caricato.

```
import openai

skill_scrittura = open("skill_scrittura_seo.txt").read()

risposta = openai.ChatCompletion.create(
    model="gpt-4o",
    messages=[
        {"role": "system", "content": skill_scrittura},
        {"role": "user", "content": "Riscrivi questo articolo: [testo]"}
    ]
)
```

Con questa struttura, il tuo skill viene iniettato ad ogni chiamata. ChatGPT lo tratta come istruzione permanente per quella sessione.

Passo 5 — Testa, raffina e versiona i tuoi skill

Uno skill non è mai finito alla prima stesura. Usalo dieci volte, nota dove ChatGPT devia dalle tue aspettative, aggiusta le istruzioni. Tieni le versioni numerate — `skill_scrittura_v1.txt`, `skill_scrittura_v2.txt` — così puoi tornare indietro se una modifica peggiora le cose.

Esempi pratici di skill da costruire

Skill per la scrittura SEO

Definisci il tono editoriale, le regole per evitare le formule tipiche dell'AI (quelle che tutti riconoscono ormai), la struttura H2/H3, la densità della keyword principale. Aggiungi una nota esplicita del tipo: "Evita le frasi segnale come 'in conclusione', 'è importante notare che', 'in un mondo in continua evoluzione'."

Skill per la ricerca e sintesi

Specifica il formato di output per le sintesi — quante fonti citare, se usare virgolette dirette o parafrasi, come strutturare la sezione "cosa manca ancora" alla fine di ogni analisi. Questo tipo di skill dimezza il tempo di elaborazione per chi lavora in ricerca, giornalismo o analisi competitiva.

Skill per l'analisi di testi

Definisci il tipo di analisi: retorica, sintattica, sentiment, tematica. Includi esempi di output atteso così ChatGPT ha un riferimento concreto — non solo regole astratte. Un esempio vale mille istruzioni.

Attenzione: gli skill template diventano obsoleti se non li aggiorni. Se il tuo stile di lavoro cambia, cambia anche il template. Un file dimenticato in una cartella è peggio che non averlo affatto — produce output incoerenti che non capisci perché sbagliati.

Pros e contro di questo approccio

Vantaggi

- Funziona senza abbonamenti extra o strumenti di terze parti
- Portabile: uno skill è un file di testo, gira ovunque
- Raffinabile nel tempo con piccole modifiche incremental
- Compatibile con qualsiasi modello GPT via API
- Zero dipendenze da aggiornamenti della piattaforma

Limiti

- Richiede un minimo di sforzo iniziale nella costruzione dei template
- Il copia-incolla manuale è meno elegante delle skill native di Claude
- La versione API richiede competenze di base in Python o simili
- ChatGPT può ignorare parti del system prompt su sessioni molto lunghe

Quando usare questo metodo — e quando no

Questo sistema è la scelta giusta se usi ChatGPT per attività ripetitive e strutturate, se hai già una

chiara idea di cosa vuoi dal modello, e se ti interessa costruire un flusso di lavoro personale che migliora con il tempo.

Non è la scelta giusta se cerchi qualcosa di più dinamico e interattivo — in quel caso, i **GPT personalizzati** (disponibili su ChatGPT Plus) sono superiori: possono essere configurati con un'interfaccia grafica, hanno memoria contestuale più gestita e possono accedere a plugin esterni.

Criterio	Skill template + Codex	GPT personalizzato	System prompt manuale
Setup	File di testo — veloce	Interfaccia guidata — intuitivo	Copia-incolla — ripetitivo
Portabilità	Alta — funziona via API	Solo ChatGPT Plus	Alta
Scalabilità	Ottima con versioning	Limitata a un GPT per caso d'uso	Bassa
Costo	Solo token API	Richiede abbonamento Plus	Nessuno (uso base)

Se arrivi a costruire anche solo due o tre skill template solidi — uno per la scrittura, uno per la ricerca, uno per l'analisi — il tempo risparmiato nelle sessioni settimanali è misurabile in ore. Non in minuti.

Vale la pena provarci. E una volta che hai capito la logica, puoi estendere il sistema a qualsiasi flusso di lavoro ripetitivo: traduzioni, brainstorming, revisione editoriale, analisi di dati. Il limite è quanto sei disposto a investire nella fase di configurazione iniziale.

Unisciti alla community dove parliamo ogni giorno di AI, OSINT e strumenti digitali.

- Newsletter: coondivido.substack.com
- Telegram: t.me/osintaipertutti
- Telegram: t.me/osintprojectgroup

Il problema che nessuno ti dice su ChatGPT

Ogni volta che apri una nuova sessione su [ChatGPT](#), il modello non ricorda nulla di te. Non sa come scrivi, non conosce il tuo settore, non ha idea che tu abbia già spiegato tre volte le stesse cose nelle sessioni precedenti. Ricominciare da capo è la norma — e diventa sfibrante.

Claude gestisce questo problema con le "skills": file di istruzione che carica automaticamente ogni volta che li attivi. ChatGPT non ha nulla di simile in modo nativo. Punto. Però c'è una via alternativa che funziona: [usare OpenAI Codex](#) come backend per costruire template di prompt strutturati e richiamabili, creando di fatto una libreria di competenze personali.

Vediamo come fare — passo dopo passo, senza promesse esagerate.

La domanda giusta non è "ChatGPT può fare quello che fa Claude?" La domanda è: "Posso costruirmi un sistema che mi faccia risparmiare tempo ogni settimana?" La risposta è sì.

Cos'è Codex e perché entra in gioco

OpenAI Codex è il modello ottimizzato per la comprensione e la generazione di codice. Nella pratica, oggi è accessibile tramite l'API di OpenAI oppure attraverso ChatGPT stesso quando attivi la modalità di ragionamento avanzato o interagisci via API con il modello code-davinci o i suoi successori.

Il punto non è usare Codex per scrivere software. Il punto è sfruttare la sua capacità di gestire istruzioni molto strutturate — quasi come dei file di configurazione — per costruire prompt che ChatGPT esegue con coerenza. Codex, in questo flusso, diventa il motore che interpreta e restituisce le istruzioni formattate. ChatGPT le riceve e le applica.

La differenza tra un prompt e uno skill template

Un prompt normale è una richiesta: "Riscrivi questo testo in modo più formale." Uno **skill template** è qualcosa di più profondo — è un documento di istruzione che definisce il contesto, le regole, i vincoli, il tono e il formato attesi, prima ancora che tu scriva la richiesta specifica.

Pensa a un istruttore di cucina. Non gli dici solo "fai una pasta": gli dai il manuale dello chef, le ricette standard, le regole della cucina. Quello è uno skill template.

Come configurare il sistema: guida passo dopo passo

Passo 1 — Definisci le aree di competenza che ti servono

Prima di scrivere anche solo una riga, chiediti: in quali attività ripetitive uso ChatGPT? Scrittura di articoli? Ricerca e sintesi? Analisi di dati? Email professionali? Ogni area diventa uno skill separato.

Tieni la lista corta: tre o quattro skill ben costruiti valgono molto più di venti raffazzonati.

Passo 2 — Costruisci il template base per ogni skill

Ogni skill ha una struttura precisa. Ecco il formato consigliato:

```
## SKILL: [Nome della competenza]
```

```
**Contesto:** [Chi sei, in che settore lavori, per quale scopo usi questo skill]
```

```
**Regole di output:**
```

- Formato richiesto (es. paragrafi narrativi, liste numerate, markdown)
- Lunghezza indicativa
- Tono (formale, diretto, tecnico, divulgativo)
- Cosa evitare (es. frasi generiche, conclusioni prevedibili)

```
**Variabili di input:**
```

- [TESTO_DA_ELABORARE]: il contenuto su cui lavorare
- [OBIETTIVO]: l'intenzione specifica per questa sessione

```
**Esempio di chiamata:**
```

```
"Usa lo skill SCRITTURA_SEO. TESTO_DA_ELABORARE: [incolla qui]. OBIETTIVO: ottimizzare per la keyword principale."
```

Passo 3 — Archivia gli skill in un file di testo accessibile

Salva ogni skill in un documento di testo semplice — Notion, Obsidian, anche una cartella su Google Drive. L'importante è che tu possa aprirlo, copiarlo e incollarlo all'inizio di una nuova sessione ChatGPT in meno di dieci secondi.

Non serve nessuna integrazione tecnica per iniziare. La velocità di accesso è l'unico criterio che conta.

Passo 4 — Usa l'API con Codex per automatizzare il caricamento

Se vuoi andare oltre il copia-incolla manuale, puoi usare l'API di OpenAI per passare il tuo skill template come **system prompt** automatico. In questo modo, ogni volta che chiami il modello tramite script o interfaccia personalizzata, lo skill viene già caricato.

```
import openai
```

```
skill_scrittura = open("skill_scrittura_seo.txt").read()

risposta = openai.ChatCompletion.create(
    model="gpt-4o",
    messages=[
        {"role": "system", "content": skill_scrittura},
        {"role": "user", "content": "Riscrivi questo articolo: [testo]"}
    ]
)
```

Con questa struttura, il tuo skill viene iniettato ad ogni chiamata. ChatGPT lo tratta come istruzione permanente per quella sessione.

Passo 5 — Testa, raffina e versiona i tuoi skill

Uno skill non è mai finito alla prima stesura. Usalo dieci volte, nota dove ChatGPT devia dalle tue aspettative, aggiusta le istruzioni. Tieni le versioni numerate — skill_scrittura_v1.txt, skill_scrittura_v2.txt — così puoi tornare indietro se una modifica peggiora le cose.

Esempi pratici di skill da costruire

Skill per la scrittura SEO

Definisci il tono editoriale, le regole per evitare le formule tipiche dell'AI (quelle che tutti riconoscono ormai), la struttura H2/H3, la densità della keyword principale. Aggiungi una nota esplicita del tipo: "Evita le frasi segnale come 'in conclusione', 'è importante notare che', 'in un mondo in continua evoluzione'."

Skill per la ricerca e sintesi

Specifica il formato di output per le sintesi — quante fonti citare, se usare virgolette dirette o parafrasi, come strutturare la sezione "cosa manca ancora" alla fine di ogni analisi. Questo tipo di skill dimezza il tempo di elaborazione per chi lavora in ricerca, giornalismo o analisi competitiva.

Skill per l'analisi di testi

Definisci il tipo di analisi: retorica, sintattica, sentiment, tematica. Includi esempi di output atteso così ChatGPT ha un riferimento concreto — non solo regole astratte. Un esempio vale mille istruzioni.

Attenzione: gli skill template diventano obsoleti se non li aggiorni. Se il tuo stile di lavoro cambia, cambia anche il template. Un file dimenticato in una cartella è peggio che non averlo affatto — produce output incoerenti che non capisci perché sbagliati.

Pros e contro di questo approccio

Vantaggi

- Funziona senza abbonamenti extra o strumenti di terze parti
- Portabile: uno skill è un file di testo, gira ovunque
- Raffinabile nel tempo con piccole modifiche incremental
- Compatibile con qualsiasi modello GPT via API

- Zero dipendenze da aggiornamenti della piattaforma

Limiti

- Richiede un minimo di sforzo iniziale nella costruzione dei template
- Il copia-incolla manuale è meno elegante delle skill native di Claude
- La versione API richiede competenze di base in Python o simili
- ChatGPT può ignorare parti del system prompt su sessioni molto lunghe

Quando usare questo metodo — e quando no

Questo sistema è la scelta giusta se usi ChatGPT per attività ripetitive e strutturate, se hai già una chiara idea di cosa vuoi dal modello, e se ti interessa costruire un flusso di lavoro personale che migliora con il tempo.

Non è la scelta giusta se cerchi qualcosa di più dinamico e interattivo — in quel caso, i **GPT personalizzati** (disponibili su ChatGPT Plus) sono superiori: possono essere configurati con un'interfaccia grafica, hanno memoria contestuale più gestita e possono accedere a plugin esterni.

Criterio	Skill template + Codex	GPT personalizzato	System prompt manuale
Setup	File di testo — veloce	Interfaccia guidata — intuitivo	Copia-incolla — ripetitivo
Portabilità	Alta — funziona via API	Solo ChatGPT Plus	Alta
Scalabilità	Ottima con versioning	Limitata a un GPT per caso d'uso	Bassa
Costo	Solo token API	Richiede abbonamento Plus	Nessuno (uso base)

Se arrivi a costruire anche solo due o tre skill template solidi — uno per la scrittura, uno per la ricerca, uno per l'analisi — il tempo risparmiato nelle sessioni settimanali è misurabile in ore. Non in minuti.

Vale la pena provarci. E una volta che hai capito la logica, puoi estendere il sistema a qualsiasi flusso di lavoro ripetitivo: traduzioni, brainstorming, revisione editoriale, analisi di dati. Il limite è quanto sei disposto a investire nella fase di configurazione iniziale.

Unisciti alla community dove parliamo ogni giorno di AI, OSINT e strumenti digitali.

- Newsletter: coondivido.substack.com
- Telegram: t.me/osintaipertutti
- Telegram: t.me/osintprojectgroup