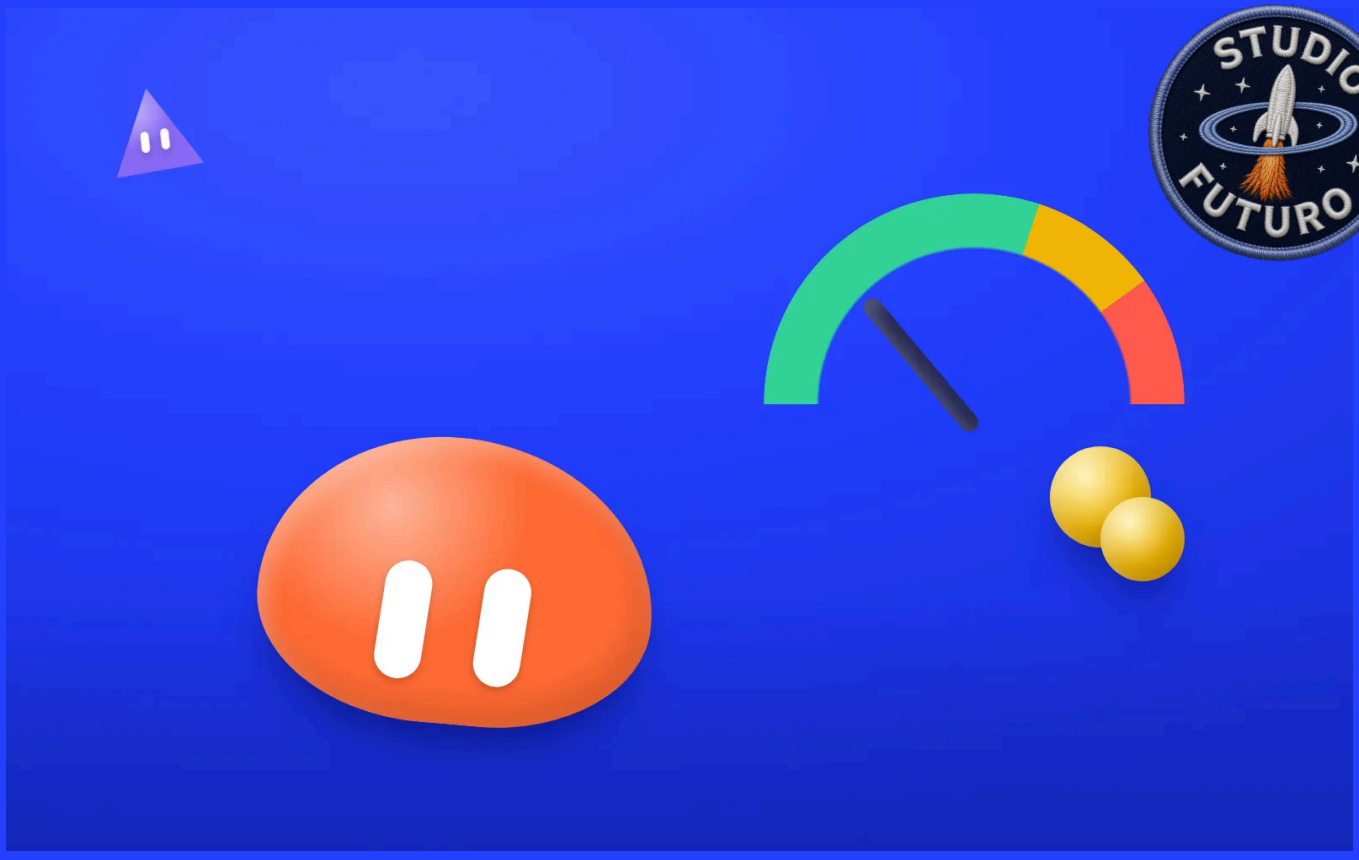


UNA PILLOLA PER CHI USA CLAUDE CODE, ANCHE DA POCO

Claude Code consuma meno.

Sette abitudini per non bruciare milioni di token con una chat sola, e Quack, il plugin gratuito che te le ricorda da solo.

7 abitudini

Da usare da subito, a mano.

1 plugin gratis

Quack le rende automatiche.

2 comandi

Per installarlo, a pagina 6.

PRIMA DI INIZIARE

5 parole da sapere.



Non serve altro per seguire questa guida. Se le conosci già, salta alla pagina dopo.

Token	Il "carburante" di Claude. Ogni parola che legge o scrive consuma qualche token. Un file di codice di 500 righe può valerne diverse migliaia.
Contesto	Tutto quello che Claude ha davanti mentre ti risponde: la conversazione, i file che ha letto, le istruzioni del progetto. A ogni messaggio lo rilegge da capo.
Cache	Una memoria temporanea che rende più economica questa rilettura. Dopo una pausa scade: in genere 5 minuti se paghi a consumo, un'ora con l'abbonamento.
/compact e /clear	Due comandi da scrivere nella chat. /compact riassume la conversazione e continua dal riassunto. /clear la svuota e riparte da zero, ma i file del progetto restano.
Subagent ed effort	Un subagent è un aiutante che lavora in uno spazio suo, magari con un modello più economico, e ti riporta solo il riassunto. L'effort è quanto Claude ragiona prima di rispondere: basso, medio o alto.

COS'È

Claude non lavora troppo. Rilegge troppo.

Claude non ricorda la chat come una persona. A ogni messaggio rilegge tutto il contesto, dall'inizio. Se la chat è lunga, anche un "ok, grazie" costa quanto rileggere un libro.

Pensa a un collega che, prima di risponderti, rilegge ad alta voce tutte le email della settimana.

UN ESEMPIO CON NUMERI TONDI: 40 MESSAGGI IN UN POMERIGGIO

6 mln

token letti con una chat sola da 150.000 token, usata da giorni

800.000

token letti con una chat per compito, intorno ai 20.000 token

7 volte

meno lettura per lo stesso lavoro

DA LEGGERE BENE

È un esempio per capire l'ordine di grandezza, non una misura. La cache abbassa il prezzo della rilettura, ma non la elimina, e dopo una pausa scade: il primo messaggio quando torni rilegge tutto a prezzo pieno.

Vale per tutti: se paghi a consumo cresce la spesa, se hai un abbonamento il limite finisce prima.

LE ABITUDINI, A MANO

Chat leggere, modelli giusti.

1 Una chat per compito

Quando passi a un lavoro diverso, svuota la chat. La nuova parte leggera. Le regole del progetto e i file, Claude li ritrova da solo.

SCRIVI

```
/clear
```

2 Riassumi a una pausa

Se un compito è lungo, a metà strada fai riassumere. Claude continua dal riassunto. Fallo a un punto fermo, non nel mezzo di una correzione delicata.

SCRIVI

```
/compact
```

3 Il modello e l'effort giusti

Per il lavoro di tutti i giorni basta un modello medio. Con opusplan il modello più forte prepara il piano e quello medio lo esegue. Per le piccole modifiche, effort basso. Non cambiare modello a metà compito: la cache riparte da zero.

SCRIVI

```
/model sonnet
/model opusplan
/effort low
```

4 Gli aiutanti per le ricerche lunghe

Cercare una funzione, leggere documentazione, fare ricerche sul web riempie la chat di testo che poi non serve. Chiedi a Claude di affidarlo a un subagent: lavora nel suo spazio e ti riporta il riassunto.

CHIEDI

```
"Affida la ricerca a un subagent"
```

UNA GIORNATA TIPO

NUOVO COMPITO	A METÀ STRADA	RICERCA LUNGA	FINE GIORNATA
/clear	/compact	subagent	annota dove sei

LE ULTIME TRE

Leggere poco, guardare i numeri.

5 Un CLAUDE.md corto

È il file di istruzioni del progetto, e Claude lo legge a ogni messaggio. La documentazione di Claude Code consiglia di restare sotto le 200 righe. Il resto va in documenti che Claude apre solo quando servono.

OBIETTIVO

`sotto 200 righe`

6 File letti a pezzi

Per correggere una funzione non serve leggere 2.000 righe. Chiedi a Claude di cercare la funzione e di leggere solo quelle righe.

CHIEDI

`"Cerca la funzione, leggi solo quelle righe"`

7 Guardare i numeri

/context mostra cosa riempie la chat adesso, /usage quanto hai consumato. Guardali una volta al giorno per una settimana: si impara in fretta dove finiscono i token.

SCRIVI

```
/context
/usage
```

SE HAI POCO TEMPO: PARTI DA DUE

Quasi tutto lo spreco viene dalla chat unica. Inizia con **/clear tra un compito e l'altro** e con **/model sonnet per il lavoro di routine**. Le altre abitudini, o il plugin, le aggiungi dopo.



IL PLUGIN GRATUITO

Quack lo fa in automatico.

Un plugin per Claude Code che trasforma le abitudini in regole e avvisi. Gratis e open source, di Alek Dobrohotov e Studio Futuro.

COME SI INSTALLA: IN CLAUDE CODE, UNO ALLA VOLTA, POI RIAVVIA

```
/plugin marketplace add AlekDob/quack-marketplace
/plugin install quack@quack-marketplace
```

Regole all'avvio

Risposte brevi, file letti a pezzi, ricerche agli aiutanti, una chat per compito.

Avvisi al momento giusto

Ogni 100.000 token: "Nuovo compito? /clear. Stesso compito? /compact." E quando una pausa ha fatto scadere la cache.

File grandi

Prima che Claude legga per intero un file di oltre 400 righe, gli ricorda di cercare solo la parte giusta.

La mappa del codice

/quack:index crea un indice di cartelle, file e funzioni con le righe esatte. Si aggiorna da solo.

La squadra di paperelle

Cinque aiutanti, ognuno con il modello giusto: pianificano, scrivono codice, cercano, studiano, prendono appunti. Pagina 7.

Memoria tra le chat

Il diario del progetto dice alla chat nuova dove eri arrivato: /clear non vuol dire ricominciare.

Controllo del setup

/quack:checkup misura cosa pesa su ogni messaggio e propone correzioni. Applica solo quelle che approvi.

Cosa non può fare

Scrivere /clear o /compact per te, o cambiare modello alla chat principale. Ti avvisa, il comando lo scrivi tu.

Costa circa 750 token a sessione, meno di una sola lettura di file evitata. Gira sul tuo computer. Manda solo un segnale anonimo quando lo installi o lo aggiorni, che puoi spegnere. Tutto su **quack.build**.

LA SQUADRA DI PAPERELLE

Quack è la paperella. E porta la squadra.

Tra programmatori si dice "rubber duck debugging": spieghi il problema a una paperella di gomma e la soluzione arriva. Quack è quella paperella, e porta cinque aiutanti. Ognuno lavora con il suo modello e alla chat riporta solo il riassunto.



Pia

PIANIFICATRICE · OPUS · ALTO

Pianifica i compiti grandi o poco chiari. Non tocca il codice.



Kit

CODER · OPUS · MEDIO

Costruisce il piano di Pia o un lavoro lungo, nel suo spazio. Restituisce solo cosa ha cambiato.



Marco

ESPLORATORE · HAIKU · BASSO

Trova dove stanno le cose. Percorsi e righe, mai file interi.



Iris

RICERCATRICE · SONNET · MEDIO

Cerca sul web e risponde con le fonti.



Dante

SCRIVANO · HAIKU · BASSO

Diario, problemi noti, decisioni: la memoria della chat dopo.

A CHI VA IL LAVORO

Compito piccolo o breve	**La tua chat**
Chat su Sonnet, compito grande o poco chiaro	**Pia pianifica, Kit costruisce**
Chat su Opus, lavoro lungo	**Kit, la chat resta leggera**
Cercare, studiare, prendere appunti	**Marco, Iris, Dante**

COSA ABBIAMO MISURATO: STESSO PROGETTO, STESSI COMPITI, DUE PROVE CIASCUNO

COMPITO	SENZA QUACK	CON QUACK
Spiegare un flusso del codice	0,40 $	0,41 $
Limite di richieste su 3 file e un test	0,56 $	0,54 $
Stesso compito, chat su Sonnet	0,25 $	0,25 $
Stesso compito, Pia e Kit (Opus), chat su Opus	–	0,92 $
Stesso compito, Pia e Kit (Opus), chat su Sonnet	–	0,63 $

Sui compiti singoli Quack cambia il conto di pochi punti. La leva più grande è il modello: su Sonnet lo stesso lavoro costa meno della metà. Pia e Kit danno qualità sui compiti difficili, non sconti. Il risparmio di Quack sta nelle chat lunghe: si vede con /usage dopo una settimana. Tutte le prove su github.com/AlekDob/quack-marketplace.

DOPO L'INSTALLAZIONE

Un prompt e sei pronto.



PRONTO DA COPIARE, DENTRO IL TUO PROGETTO

```
Usa la skill quack:checkup su questo progetto.
Mostrami cosa pesa su ogni messaggio in una tabella semplice: voce, quanto pesa, cosa conviene fare.
Proponimi le correzioni una alla volta e applica solo quelle che approvo.
Poi usa la skill quack:index per creare la mappa del codice.
Alla fine spiegami in parole semplici cosa è cambiato e come conviene lavorare da domani.
```

PROVA PICCOLA, UNA SETTIMANA

Prima di installare, apri la chat che usi da giorni e scrivi /context: guarda quanti token si porta dietro. Poi installa Quack, scrivi /clear e affronta il prossimo compito in una chat nuova. Dopo una settimana confronta /usage con la settimana prima.

DOMANDE FREQUENTI

Devo saper programmare?

Devi saper usare Claude Code. Il resto sono due comandi da copiare.

Abbonamento o API?

Tutti e due. Con l'abbonamento il limite dura di più, con l'API spendi meno.

I miei file vanno da qualche parte?

No. Quack gira sul tuo computer. Quando lo installi o lo aggiorni manda un solo segnale anonimo: versione e un codice casuale. Con QUACK_NO_PING=1 lo spegni.

Funziona con Cursor o Codex?

No, è un plugin per Claude Code. Le sette abitudini invece valgono con qualunque assistente.

ATTENZIONE A

Il plugin avvisa. Decidi tu.

- **L'esempio dei 6 milioni è un esempio.** Quanto risparmi dipende da come lavori oggi.
- **/clear e /compact restano a te.** Sei tu a sapere quando un compito è finito.
- **Un riassunto può perdere un dettaglio.** Prima di /compact, fai annotare dove sei arrivato.
- **Limiti e prezzi cambiano.** Controlla /usage e la documentazione di Claude Code.

FONTI

Guida ai costi di Claude Code (code.claude.com/docs/en/costs). PageIndex di VectifyAI per l'idea della mappa ad albero. caveman per le risposte brevi. RTK per accorciare l'output dei comandi.

Quack è gratis.

Open source, di Alek Dobrohotov e Studio Futuro. Vuoi che lo impostiamo noi sul tuo team? Misuriamo i consumi prima e dopo, sui vostri progetti veri.

quack.build
hello@studiofuturo.ai

