



Dominare i token in Claude: 7+1 trucchi anti-blocco

Otto trucchi semplici per consumare meno token ed evitare i blocchi

Gianluca Bonomo · www.gianlu.it · info@gianlu.it



Indice

Come usare questa dispensa 3

1. Perché Claude ti blocca la chat 3

1.1 Il decimo messaggio costa più del primo 4

2. Trucco 1: resetta il contesto con un riassunto 5

3. Trucco 2: riscrivi la storia invece di correggerla 6

3.1 L'errore da evitare 7

3.2 Il modo giusto 7

4. Trucco 3: avvia la sessione in anticipo per averne due 8

4.1 Approccio base e approccio avanzato 8

5. Trucco 4: proteggi il Prompt Caching 9

5.1 Le azioni che azzerano la cache 9

6. Trucco 5: usa i Progetti per i documenti che ti servono sempre 11

7. Trucco 6: disattiva i plugin che non usi 12

8. Trucco 7: scegli il modello giusto per il compito 13

9. Trucco extra: scrivi prompt brevi e concisi 14

9.1 Altri esempi concreti 15

10. La check-list in una pagina 16

Esercitazione pratica 18

Le domande che arrivano in aula 19

Glossario 19

Per approfondire 20

Come usare questa dispensa

Questa dispensa raccoglie, in forma scritta e ordinata, tutto quello che vediamo nella lezione. Non serve studiarla a memoria: serve a ritrovare le cose con calma, dopo, quando la lezione e' finita e resta una domanda in sospeso.

> **OBIETTIVO**
>
> Capire che cosa consuma i token in Claude e applicare otto trucchi pratici per lavorare più a lungo, senza blocchi improvvisi e con meno spreco.

Che cosa saprai fare alla fine

- Spiegare perché il decimo messaggio di una chat costa molto più del primo.
- Resettare una chat lunga con un riassunto (il «reset della chat») e usare /compact in Claude Code.
- Correggere un errore con la matita invece di scrivere un nuovo messaggio.
- Sfruttare le sessioni da 5 ore e proteggere il Prompt Caching.
- Usare Progetti, plugin e modelli (Haiku, Sonnet, Opus) in modo mirato.
- Scrivere prompt brevi e concisi, senza perdere precisione.

1. Perché Claude ti blocca la chat

Il blocco non dipende dai messaggi che invii: dipende da quanti **token** consumi.

Il modello non conta quanti messaggi mandi, conta quanto testo deve leggere e scrivere. L'unità di misura è il **token**, e un token equivale a poco meno di una parola.



I token sono l'unità di energia della chat



Si legge tutto in blocco, si risponde un pezzo alla volta

OBIETTIVO PRIMARIO

Lavorare più a lungo consumando meno token.

1.1 Il decimo messaggio costa più del primo

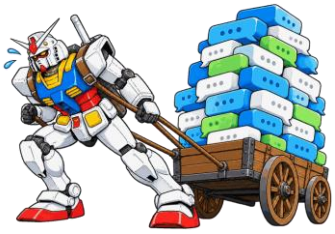
Claude non legge mai solo la tua ultima frase. Per rispondere **rilegge l'intera conversazione dall'inizio**. Più avanzi nella chat, più token bruci soltanto per fargli rileggere il passato.

Messaggio	Che cosa rilegge Claude	Consumo
Messaggio 1	Solo il messaggio 1	Basso
Messaggio 2	Messaggio 1 + messaggio 2	Più alto
Messaggio 5	Tutti i messaggi dall'1 al 5	Alto
Messaggio 10	Tutti i messaggi dall'1 al 10	Critico

Schema semplificato: ogni messaggio trascina con sé tutta la storia precedente.



Il decimo messaggio costa più del primo



Ogni messaggio trascina la storia accumulata

ESEMPIO CONCRETO

Stai preparando un'offerta commerciale e fai domande una dopo l'altra nella stessa chat: titolo, prezzi, condizioni, tono, correzioni.

Al decimo messaggio Claude deve rileggere anche tutte le bozze e le correzioni dei nove precedenti. La domanda è breve, ma **il costo non lo è**.

DA RICORDARE

Più la chat è lunga, più ogni nuovo messaggio costa. Gli otto trucchi che seguono servono a spezzare questa spirale.

2. Trucco 1: resetta il contesto con un riassunto

Dopo **10-15 messaggi** il consumo diventa critico: è il momento di staccare la chat e ripartire pulito.



Una chat troppo carica pesa su ogni risposta

Il **reset della chat** azzera il peso della conversazione senza perdere il lavoro fatto: porti con te solo un riassunto, non tutta la storia.

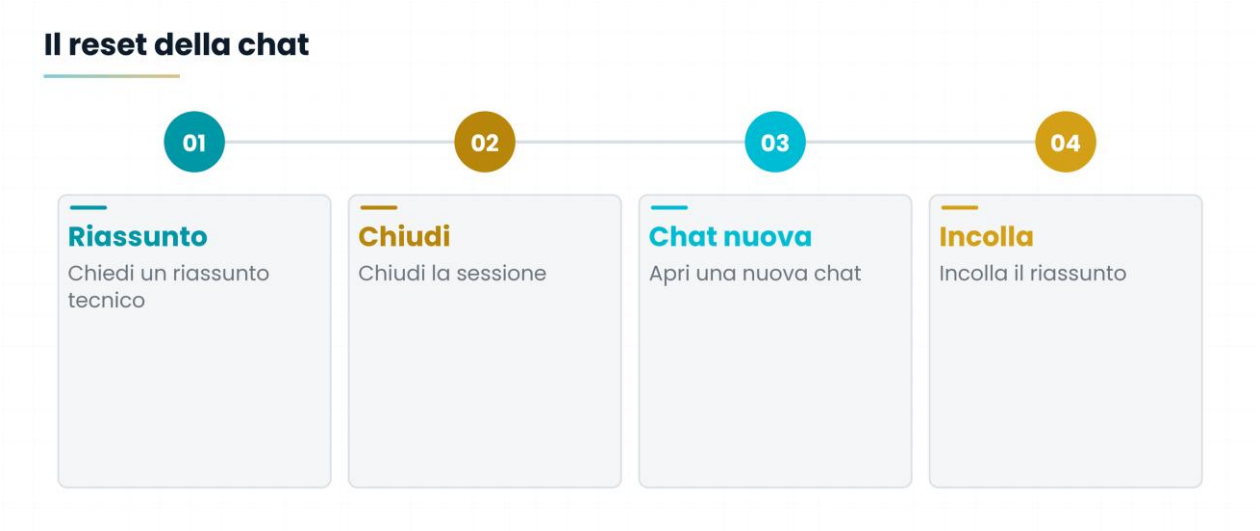
1. Chiedi a Claude un **riassunto tecnico e dettagliato** dell'intera chat.
2. **Chiudi** la sessione attuale.
3. Apri una **nuova chat** e incolla il riassunto: il contesto è ripristinato a costo quasi zero.



Il riassunto comprime la chat



Nuova chat, contesto pulito



Il reset della chat in quattro passi

ESEMPIO CONCRETO: IL PROMPT DI RIASSUNTO

Alla chat lunga scrivi: «Riassumi in modo tecnico e dettagliato tutta la conversazione: obiettivo, decisioni prese, dati importanti, punti ancora aperti.»

Nella chat nuova incolli il riassunto e continui con la domanda successiva: «Partendo da qui, scrivi la bozza del punto 3.»

SE USI CLAUDE CODE

La funzione **/compact** esegue questa operazione automaticamente, di default.

3. Trucco 2: riscrivi la storia invece di correggerla

Quando Claude sbaglia, non rispondere «no, volevo un'altra cosa»: **modifica il prompt originale con la matita**.

3.1 L'errore da evitare

Inviare un nuovo messaggio del tipo «No, non intendevo questo, volevo X» ti fa consumare token per fargli rileggere l'errore. In più sporchi la memoria della chat: **la risposta sbagliata resta lì** e l'IA continuerà a tenerla a mente.

3.2 Il modo giusto

Clicca l'**icona matita** sul tuo prompt originale, aggiungi i dettagli mancanti e sovrascrivi. Claude cancella la prima risposta e ne genera una nuova: **contesto pulito, consumo di token dimezzato**.



La matita corregge il prompt originale



Rewind: si torna al passo giusto



Correggere con un nuovo messaggio o con la matita

ESEMPIO CONCRETO

Prompt originale: «Scrivi una mail per sollecitare un pagamento.» Claude risponde con un testo molto formale.

Sbagliato: scrivere «No, più amichevole e più corta» → la mail formale resta in chat e viene riletta a ogni messaggio.

Corretto: matita sul prompt originale → «Scrivi una mail per sollecitare un pagamento, tono amichevole, massimo 5 righe.» → la risposta vecchia sparisce.

4. Trucco 3: avvia la sessione in anticipo per averne due

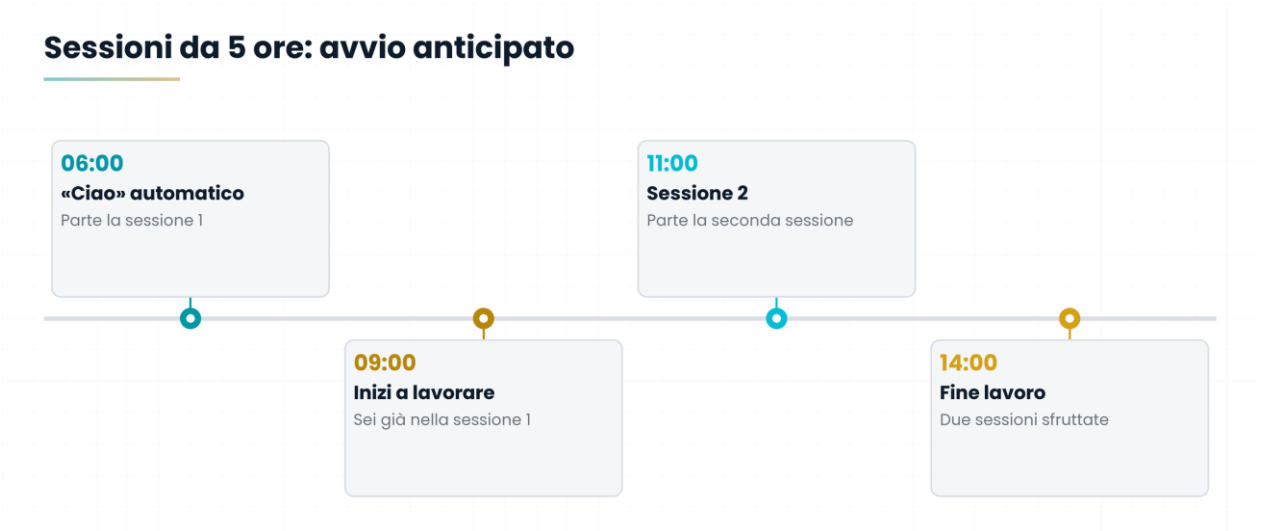
Ogni sessione di lavoro ha un limite massimo di token e dura **esattamente 5 ore dal primo messaggio** inviato.

4.1 Approccio base e approccio avanzato

- **Approccio base:** inizi a lavorare alle 09:00, la sessione scade alle 14:00. Hai sfruttato **una sola sessione** utile.
- **Approccio avanzato:** imposti un'automazione (per esempio con Make o Co-work) che invia un semplice «Ciao» alle **06:00**. La prima sessione parte prima e le successive si spostano di conseguenza.



Si programma l'avvio della sessione



Sessioni da 5 ore: base e avanzato

ESEMPIO CONCRETO

Lavori dalle 09:00 alle 14:00. Con l'avvio alle 06:00 la sessione 1 copre dalle 06:00 alle 11:00 e la sessione 2 parte alle 11:00: nel tuo orario di lavoro hai **due sessioni** di token invece di una.

ATTENZIONE

Raddoppi i token giornalieri disponibili a parità di orario lavorativo, ma **non alteri il cap settimanale**.

5. Trucco 4: proteggi il Prompt Caching

Claude non rilegge letteralmente tutto ogni volta: usa il **Prompt Caching** per tenere in memoria parti della chat recente.

- **Il sistema:** il Prompt Caching salva in memoria le parti della conversazione già elaborate. L'attivazione è automatica.
- **Il vantaggio:** ti fa risparmiare molti token sulle informazioni già processate.
- **Il rischio:** la cache è **fragile e temporanea**. Richiede un flusso di lavoro continuo.

5.1 Le azioni che azzerano la cache

Se compi una di queste azioni, la cache si azzera: Claude dovrà rileggere tutta la chat da zero e brucerai token all'istante.

- **Pause temporali prolungate** tra un messaggio e l'altro.
- **Cambio del modello** in corsa (per esempio da Sonnet a Opus).
- **Modifica del livello di Effort** richiesto al sistema.
- **Collegamento o scollegamento** di plugin o server MCP.



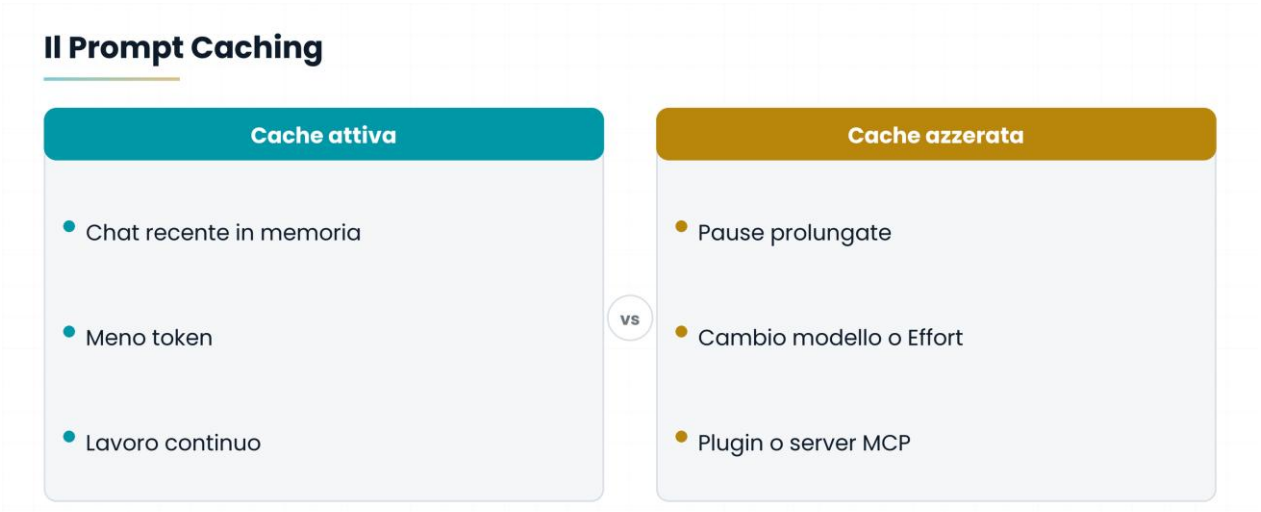
Cache attiva: la parte già letta è protetta



Cache azzerata: lo scudo si rompe



Dopo una lunga pausa la cache scade



Cache attiva e cache azzerata

ESEMPIO CONCRETO

Hai una chat lunga su un'analisi. Vai a pranzo, torni dopo molto tempo e al secondo messaggio passi da Sonnet a Opus.

Hai azzerato la cache due volte in un colpo: Claude rilegge tutto da capo. **Meglio:** scegli modello e impostazioni prima di iniziare e lavora a blocchi continui.

6. Trucco 5: usa i Progetti per i documenti che ti servono sempre

Caricare lo stesso file in ogni chat consuma token inutilmente: i documenti ricorrenti vanno nei **Progetti**.

- **Procedura standard (da evitare):** caricare manualmente lo stesso PDF in ogni singola chat.
- **Procedura consigliata:** carica documenti e linee guida ricorrenti nella funzione **Progetti**.
- **Il vantaggio:** i file nei Progetti restano stabilmente in cache. Interrogarli costa solo una minima frazione dei token rispetto al caricamento manuale transitorio.



Nel Progetto l'archivio si carica una volta



Lo stesso file caricato due volte: spreco



Upload ripetuti o Progetto

ESEMPIO CONCRETO

Ogni settimana prepari preventivi e ti serve sempre lo stesso listino in PDF, più le linee guida di stile dell'azienda.

Invece di ricaricarli in ogni chat, creali una volta nel Progetto «Preventivi» e apri lì tutte le nuove conversazioni.

7. Trucco 6: disattiva i plugin che non usi

Ogni strumento attivo carica **istruzioni invisibili** nel contesto della chat, anche quando non lo usi.

L'errore classico è tenere tutti i plugin e i server MCP attivi di default, «perché non si sa mai». In realtà ogni strumento acceso consuma token preziosi in background, in modo totalmente inutile.



Meno strumenti attivi, meno zavorra

REGOLA PRATICA

Disattiva tutto ciò che non è strettamente necessario per il lavoro che stai facendo.



Tutto attivo o solo l'essenziale

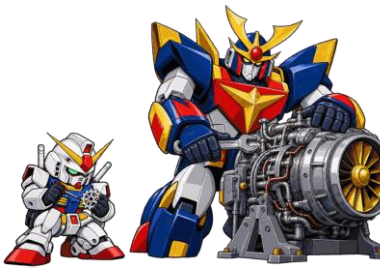
> **ESEMPIO CONCRETO**
>
> Devi solo correggere un testo, ma hai attivi tanti plugin e server MCP di altri progetti.
>
> **Spegni** quelli che non servono **prima** di aprire la chat: lavori con un contesto più leggero. Ricorda che collegare o scollegare strumenti a chat avviata fa cadere la cache (Trucco 4).

8. Trucco 7: scegli il modello giusto per il compito

Non usare il modello più potente per un compito semplice: **Haiku è la scelta giusta** per consumare meno token.

Modello	Quando usarlo	Consumo di token
Opus / Sonnet	Programmazione complessa, analisi profonda, ragionamento logico	Altissimo
Haiku	Correzioni grammaticali, traduzioni, riscritture veloci, task ripetitivi	Minimo

Scegli il modello in base alla difficoltà del compito.



Il modello giusto per il compito



Il ragionamento serve ai compiti difficili



Il modello giusto per il compito

ESEMPIO CONCRETO

Haiku: correggere gli errori di una mail, tradurre un paragrafo, riscrivere una frase più chiara.

Sonnet o Opus: scrivere una funzione complessa, analizzare un contratto, ragionare su un problema logico a più passaggi.

Ricorda: il modello si sceglie **a inizio chat**, perché cambiarlo in corsa fa cadere la cache (Trucco 4).

9. Trucco extra: scrivi prompt brevi e concisi

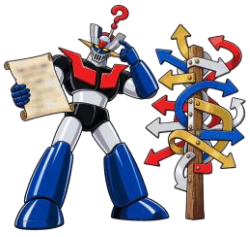
L'IA non ha sentimenti e non si offende: elimina cortesia, fronzoli e sintassi complessa. **Meno parole digiti in input, meno token consumi.**

Prompt	Esempio
Lungo e cortese (spreco)	«Ciao Claude, saresti così gentile da leggere questo file e poi farmi una tabella per favore?»
Breve e conciso (efficienza)	«Analizza file X. Estrai dati Y. Formatta tabella.»

Stesso risultato, molte meno parole in ingresso.



Prompt lungo contro prompt breve



Prompt vago: tante strade, molto spreco



Prompt preciso: un obiettivo a fuoco



Output sintetico: solo l'essenziale

9.1 Altri esempi concreti

- Invece di «Potresti per favore tradurre questo testo in inglese, se non è un disturbo?» scrivi: **«Traduci in inglese.»**
- Invece di «Vorrei un riassunto di questo documento, magari non troppo lungo» scrivi: **«Riassumi in 5 punti.»**
- Invece di «Mi aiuteresti a correggere gli errori di questo paragrafo?» scrivi: **«Correggi errori.»**

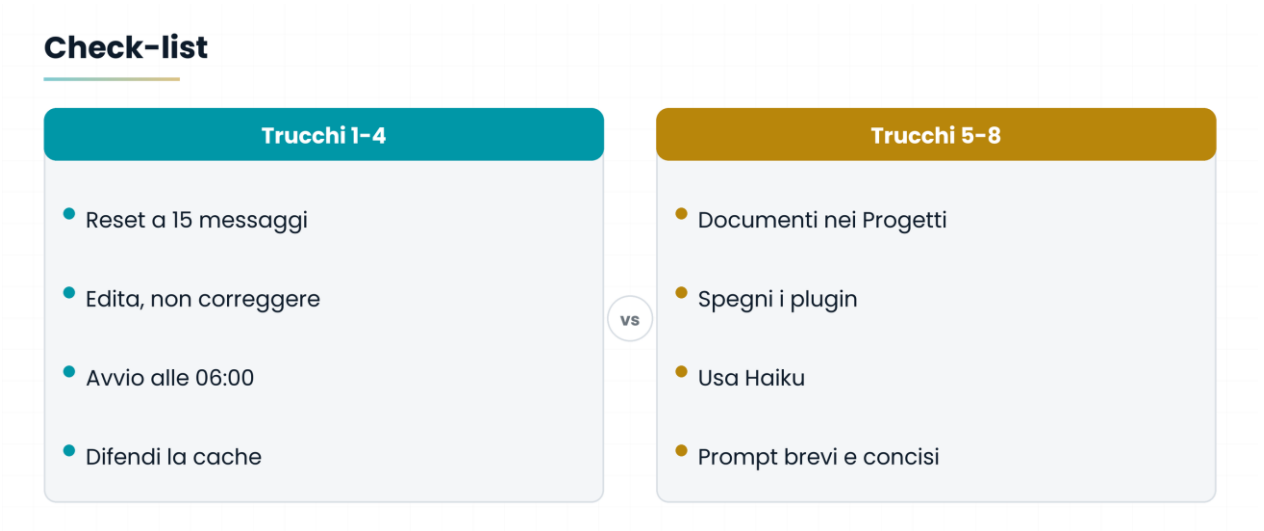


Il file indicato direttamente con @

> **BREVE NON VUOL DIRE VAGO**
>
> Taglia la cortesia, non le informazioni: obiettivo, file e formato del risultato devono restare chiari.

10. La check-list in una pagina

Gli otto trucchi in un colpo d'occhio: stampala e tienila accanto alla tastiera.



La check-list

1. **Reset a 15 messaggi** (usa i riassunti).
2. **Edita, non correggere** (usa la matita).
3. **Avvio anticipato alle 06:00** (raddoppia le sessioni).
4. **Difendi la cache** (evita pause e cambi di impostazioni).
5. **Documenti nei Progetti** (basta upload ripetuti).

6. **Spegni i plugin** (spegni i server MCP inattivi).
7. **Usa Haiku** (per i micro-task quotidiani).
8. **Prompt brevi e concisi** (zero fronzoli, massima resa).



Si controllano consumo e contesto

IN SINTESI

Gestendo meglio il lavoro riuscirai a ottimizzare le spese e ad aumentare l'efficienza.

Esercitazione pratica

Quindici minuti, Claude aperto e una tua chat recente e lunga.



Un compito complesso diviso in fasi

1. Conta i messaggi della chat: sono più di 10-15? Se sì, chiedi a Claude un riassunto tecnico e dettagliato.
2. Apri una chat nuova, incolla il riassunto e prosegui il lavoro.
3. Trova una risposta che non ti convinceva e correggila con la matita sul prompt originale.
4. Riscrivi un tuo prompt lungo e cortese in forma breve e concisa (come «Analizza file X. Estrai dati Y. Formatta tabella.»).
5. Controlla plugin e server MCP attivi e spegni quelli che non ti servono per il lavoro di oggi.
6. Scegli un compito semplice (una traduzione, una correzione) e provalo con Haiku.

> **ATTENZIONE**
>
> Hai ripartito da una chat nuova con un riassunto?
>
> Hai usato la matita invece di un messaggio di correzione?
>
> Hai riscritto almeno un prompt in forma breve e concisa?

Le domande che arrivano in aula

Quanti messaggi posso fare in una chat prima di ripartire?

La regola pratica è **10–15 messaggi**: oltre questa soglia il consumo diventa critico. A quel punto chiedi un riassunto e apri una chat nuova.

Perché dopo un'ora di pausa la chat consuma di più?

Perché la cache del Prompt Caching è temporanea: dopo una pausa prolungata la cache si azzera e Claude deve rileggere tutta la chat da zero.

L'avvio anticipato alle 06:00 aumenta anche il limite settimanale?

No. Raddoppia i token giornalieri disponibili a parità di orario di lavoro, ma non altera il cap settimanale.

Meglio la matita o un nuovo messaggio per correggere?

La matita: cancella la prima risposta e ne genera una nuova, con contesto pulito e consumo di token dimezzato.

Un prompt breve e conciso peggiora la risposta?

No, se resta preciso: taglia cortesie e giri di parole, ma lascia obiettivo, file e formato del risultato.

Glossario

Termine	Che cosa significa
/compact	Funzione di Claude Code che esegue in automatico il reset della chat.
Effort	Livello di impegno richiesto al sistema: cambiarlo a chat avviata azzera la cache.
Haiku, Sonnet, Opus	I modelli di Claude: Haiku è leggero, Sonnet e Opus sono pesanti e consumano molto di più.
Plugin e server MCP	Strumenti collegati a Claude: se attivi caricano istruzioni invisibili nel contesto.
Progetti	Funzione che conserva file e linee guida ricorrenti, mantenuti stabilmente in cache.
Prompt Caching	Memoria temporanea che salva le parti recenti della chat e fa risparmiare token.

Termine	Che cosa significa
Reset della chat	Riassumere la chat, chiuderla e ripartire in una chat nuova incollando il riassunto.
Sessione	Finestra di lavoro con un limite massimo di token: dura 5 ore dal primo messaggio inviato.
Token	L'unità di consumo dell'IA: equivale a poco meno di una parola.

Per approfondire

Gianluca Bonomo, Niente paura, è solo IA - il libro di riferimento

> **ATTENZIONE**
>
> Funzioni, nomi dei menu, limiti e modelli cambiano spesso: controlla sempre la versione di Claude che usi. Per approfondire, il libro di riferimento è «Niente paura, è solo IA».