



Prodotti Lavora con noi

notizia



## Programmazione in C64 BASIC

Scopri come programmare usando C64 BASIC, disponibile su THEC64 Mini e THEC64.

Il CAROSSELLO GIOCHI su **THEC64 Mini** ti dà accesso al linguaggio di programmazione BASIC. Sul **THEC64** full-size , avviare o passare alla modalità classica per accedere a BASIC.

Se si desidera seguire gli esempi BASIC, si consiglia vivamente di collegare una tastiera USB a **THEC64 Mini** , piuttosto che provare a utilizzare la tastiera virtuale. Leggere il **CAPITOLO 6: TASTIERE** Nel PDF della Guida dell'utente per THEC64 Mini prima di iniziare questa introduzione a BASIC.

Utilizzare la tastiera fornita se si accede a BASIC dalla modalità classica sul **THEC64** full size .

Se una memory stick USB compatibile (formattata su FAT32) è collegata al **THEC64** full size o al **THEC64 Mini** , è possibile salvare il codice BASIC sulla memory stick. Affinché la chiavetta USB venga rilevata su **THEC64 Mini** , collegarla prima **dell'avvio di** BASIC. Vedere anche **Salvataggio e caricamento di** seguito per ulteriori informazioni.

Senza una chiavetta USB, si salva in uno dei quattro slot di salvataggio disponibili per BASIC su THEC64 Mini o THEC64 stesso.

## CHE COS'È IL BASIC?

Quando accendi un computer C64, la prima cosa che vedi è BASIC. Noi forniamo l'accesso al BASIC per coloro che desiderano esperienza di programmazione utilizzando la versione 2 della variante C64 dei **B**eginners **A**ll-**s**cope **s**ymbolic **h**o **n**struction **C**ode.

BASIC è un linguaggio di programmazione di alto livello, progettato per essere facile da usare. È un linguaggio interpretato, il che significa che tutte le istruzioni BASIC digitate devono essere tradotte dal computer prima che possano essere eseguite. Questo in realtà accade abbastanza rapidamente e viene fatto in modo invisibile e automatico dal computer.

Computer come nativamente 'parlano' nel codice macchina, che è una serie di zeri (0) e uno (1). Quando usano il codice macchina, funzionano molto velocemente. Tuttavia, le persone non programmano direttamente nel codice macchina, quindi usiamo un modo diverso per inviare istruzioni al computer, usando linguaggi di programmazione più vicini a ciò a cui siamo abituati.

Alcuni linguaggi di programmazione sono pochi passi più vicini al codice macchina e quindi vengono interpretati (o tradotti se preferisci) più rapidamente di quelli che non lo sono.

BASIC è sempre stata un'eccellente introduzione alla programmazione per principianti completi anche se si trova a pochi passi dal codice macchina.

Imparare a programmare usando C64 BASIC ti aiuta a pensare in modo programmatico e ti insegna anche a digitare su una tastiera. Entrambe queste abilità sono importanti da imparare.

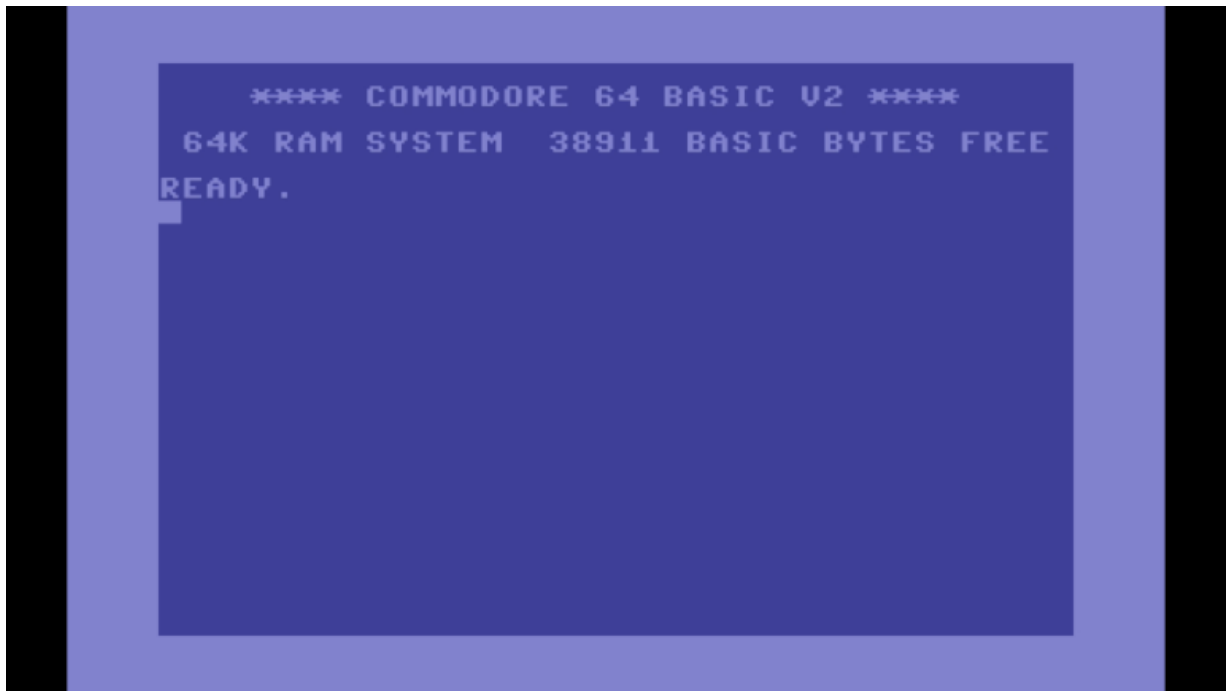
## PAROLE CHIAVE DI BASIC

C64 BASIC ha una serie di parole chiave che indicano al computer cosa fare dopo. Impara quelli e scriverai il codice BASIC in pochissimo tempo. Ci sono 71 parole chiave BASIC da scoprire.

Se pensi che 71 suoni molto, non ti preoccupare. Non devi impararli tutti in una volta. Molti di questi sono legati al fare calcoli matematici e entrano in gioco solo quando inizi a fare codifiche più complesse. Maggiori informazioni sulle parole chiave più tardi!

## LANCIO C64 DI BASIC

Quando si avvia BASIC per la prima volta su un C64, viene visualizzata una schermata blu. Inizia rivelando quanta memoria del computer è disponibile per la programmazione. Sotto quell'informazione c'è un prompt PRONTO e sotto c'è un cursore quadrato che lampeggia costantemente.

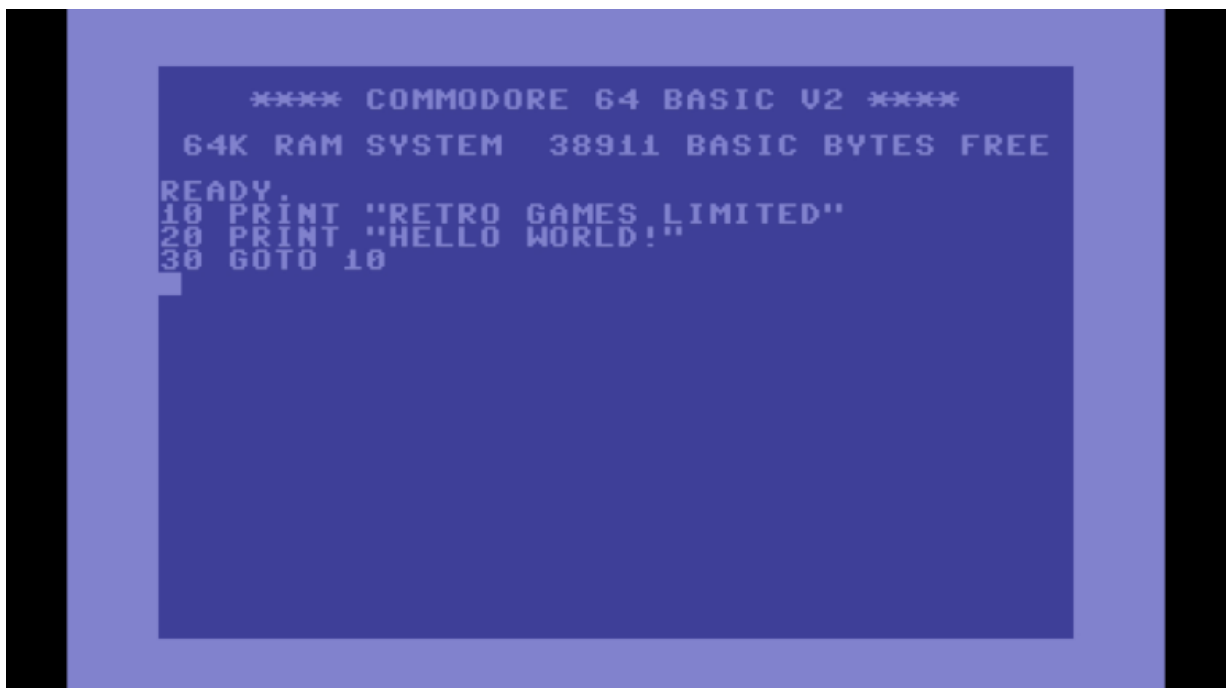


Fissare il cursore non ti porterà molto lontano. È pronto e ti aspetta per inserire i tuoi primi comandi BASIC.

Esistono due modi per inserire il codice BASIC. Puoi inserirlo e premere INVIO. Il computer proverà immediatamente ad eseguire il codice. Tuttavia, per ripetere lo stesso codice in un secondo momento, è necessario inserirlo nuovamente. Non va molto bene. Il secondo metodo è molto più efficiente. Metti i numeri all'inizio di ogni riga del codice che indica al computer in quale ordine eseguirli. La cosa fantastica è che l'uso dei numeri di riga significa anche che non eseguirà il codice finché non si digita la parola chiave RUN \* e \* significa che non è necessario digitarlo nuovamente (purché non si spenga il computer).

## IL TUO PRIMO PROGRAMMA BASE

Di seguito è riportato un esempio di programma BASIC. Vedremo a turno ogni riga di codice prima di farlo davvero.



Ci sono tre righe separate di codice BASIC da spiegare. La riga 10 utilizza la parola chiave PRINT che invia tutto ciò che appare tra virgolette "" sullo schermo del computer (e non a una stampante come ci si potrebbe aspettare!).

```
10 PRINT "RETRO GAMES LIMITED"
20 PRINT "HELLO WORLD!"
30 GOTO 10
```

La riga 20 fa lo stesso della riga 10 ma invierà parole diverse sullo schermo.

La linea 30 indica al computer di passare alla linea 10, che inserisce il computer in un ciclo della linea 10 in esecuzione, quindi la linea 20, quindi la linea 30 che gli dice di tornare nuovamente alla linea 10 e così via.

I numeri di riga sono in unità di dieci nel caso in cui si desideri inserire righe di codice aggiuntive, ad esempio una riga 15 che si trova tra la riga 10 e la riga 20. Se si digita una riga di codice che inizia con lo stesso numero di riga di un'esistente riga, la nuova riga sostituirà quella precedente non appena si preme INVIO, senza alcun avviso. Se si digita solo il numero di riga e nient'altro, quindi premere INVIO, si elimina quel numero di riga.

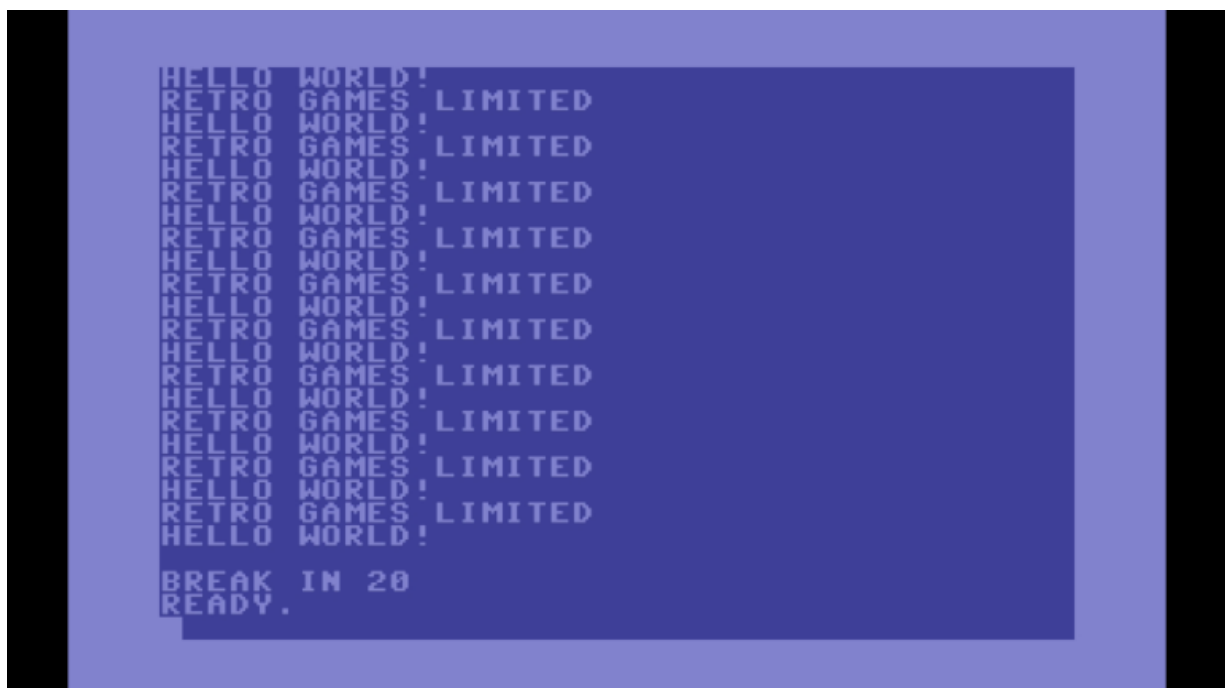
Ora digita ogni riga esattamente come la vedi (sopra). Premere INVIO alla fine di ogni riga numerata per impegnare quella riga nella memoria del computer e spostarsi lungo una riga, prima di iniziare a digitare la successiva.

Premere INVIO alla fine della riga 30.

Se si commette un errore, utilizzare il tasto Backspace sulla tastiera USB (o premere il pulsante C sul joystick quando si utilizza la tastiera virtuale). Ad ogni pressione il carattere verrà immediatamente cancellato a sinistra della posizione del cursore sullo schermo.

Se sei contento di aver digitato tutto correttamente, digita la parola chiave RUN (senza un numero di riga). Il tuo codice verrà quindi eseguito!

Se ti stai chiedendo come fermare il tuo nuovo codice BASIC per sempre, prova a premere ESC (su una tastiera USB collegata a **THEC64 Mini**) o RUN / STOP (se usi la tastiera virtuale o la tastiera **THEC64** full-size). Hai richiesto al computer di BREAK nel tuo codice. Non preoccuparti, non è così male come sembra!



Per vedere il tuo codice ancora intatto, puoi digitare la seguente parola chiave nel cursore quadrato: Le

**LIST**

tue tre righe di codice sono visualizzate, sicure e valide proprio come lo erano state l'ultima volta che le hai viste.

**Parole chiave di base**

Che ci crediate o no, avete già usato quattro parole chiave BASIC. Per dimostrarlo, ecco una tabella che mostra tutti e 71.

<b>addominali</b>	<b>E</b>	<b>ASC</b>	<b>ATN</b>	<b>CHR \$</b>	<b>VICINO</b>	<b>CLR</b>	<b>CMD</b>
<b>CONT</b>	<b>COS</b>	<b>DATI</b>	<b>DEF</b>	<b>DIM</b>	<b>FINE</b>	<b>EXP</b>	<b>FN</b>
<b>PER</b>	<b>FRE</b>	<b>OTTENERE</b>	<b>GET \$</b>	<b>GOSUB</b>	<b>VAI A</b>	<b>SE</b>	<b>INGRESSO</b>
<b>\$ INPUT</b>	<b>INT</b>	<b>LEFT \$</b>	<b>LEN</b>	<b>PERMETTERE</b>	<b>ELENCO</b>	<b>CARICARE</b>	<b>LOG</b>
<b>MID \$</b>	<b>NUOVO</b>	<b>IL PROSSIMO</b>	<b>NON</b>	<b>SU</b>	<b>APERTO</b>	<b>O</b>	<b>SBIRCIARE</b>
<b>POKE</b>	<b>POS</b>	<b>STAMPA</b>	<b>STAMPA \$</b>	<b>LEGGERE</b>	<b>REM</b>	<b>RISTABILIRE</b>	<b>RITORNO</b>
<b>\$ DESTRA</b>	<b>RND</b>	<b>CORRERE</b>	<b>SALVA</b>	<b>SGN</b>	<b>PECCATO</b>	<b>SPC</b>	<b>SQR</b>
<b>STATO</b>	<b>PASSO</b>	<b>FERMARE</b>	<b>STR \$</b>	<b>SYS</b>	<b>TAB</b>	<b>TAN</b>	<b>POI</b>
<b>TEMPO</b>	<b>TEMPO \$</b>	<b>PER</b>	<b>USR</b>	<b>VAL</b>	<b>VERIFICARE</b>	<b>ASPETTARE</b>	

Ora cambieremo leggermente il tuo codice in modo che stampi le due righe di testo solo cinque volte prima di interrompere, anziché continuare all'infinito. Dovresti immediatamente notare che abbiamo aggiunto un numero di riga 5 e abbiamo modificato ciò che era sulla riga 30.

```
5 FOR A=1 TO 5
10 PRINT "RETRO GAMES LIMITED"
20 PRINT "HELLO WORLD!"
30 NEXT A
```

Guarda queste due nuove righe di codice per un momento e ignora le linee 10 e 20 nel mezzo. Queste linee introducono due nuovi concetti alle tue conoscenze BASIC e due nuove parole chiave.

Un concetto è la definizione e l'utilizzo di variabili nel codice. In questo esempio, stiamo usando 'A' come contenitore per memorizzare temporaneamente un numero all'interno. La lettera A è stata scelta ma avrebbe potuto facilmente essere la parola "MONDO", quindi avrebbe potuto leggere ... .. e il tuo programma avrebbe comunque funzionato. Ad esempio, potresti aver modificato il numero alla fine della riga di codice da 5 a 6. I nomi delle variabili sono scelti da te e possono (quasi) essere qualsiasi cosa ti piaccia. Tuttavia, ci sono delle regole per nominare queste variabili utili in cui archiviare le cose.

```
5 FOR WORLD=1 TO 5
10 PRINT "RETRO GAMES LIMITED"
20 PRINT "HELLO WORLD!"
30 NEXT WORLD
```

Non è possibile utilizzare parole chiave riservate utilizzate da BASIC o da C64 per le proprie variabili di sistema. Come fai a sapere quali non possono essere utilizzati? Bene, se avessi usato LIST come nome della tua variabile, l'esecuzione del programma avrebbe prodotto:

```
?SYNTAX ERROR IN 5
```

Non è il più utile degli errori, ma se il computer restituisce quell'errore, sai che qualcosa non va. Ricordi quell'indice delle parole chiave BASIC C64? Non usare nessuno di questi come nomi per le variabili e non puoi sbagliare.

Cerca di mantenere i nomi delle variabili brevi, ma pertinenti dove puoi. Se il tuo codice non è eccessivamente complesso, puoi semplicemente usare A, B, C e così via, purché ti ricordi a cosa servono!

## Tipi di variabili

L'unica altra cosa da sapere è che ci sono tre tipi di variabili e sono definiti da quale tipo di informazione è memorizzata in essi.

I due più comuni che userete sono variabili **intere** e **stringa**.

Un numero intero è semplicemente un numero intero, quindi nessuna frazione o punto decimale. Una stringa è costituita da lettere o lettere e numeri.

Come si dice quale tipo di variabile si desidera utilizzare? Per gli interi, basta usare un nome per la variabile, senza nient'altro, ad esempio MONDO. Per le stringhe, aggiungi \$ alla fine del nome della variabile, quindi RECIPE \$ potrebbe essere utilizzato per memorizzare il nome della tua ricetta preferita di torta o biscotto, ad esempio "Rocky Road".

Quindi, guardando il tuo esempio BASIC modificato, sappiamo che 'A' è usato per memorizzare un numero intero (che può effettivamente essere compreso tra -32768 e 32767). Nel nostro esempio, raggiungerà solo un valore massimo di 5.

Ora parliamo delle parole chiave FOR e NEXT. Questi sono accoppiati insieme e creano un ciclo FOR ... NEXT. Quello che fa è dalla parola chiave FOR, il computer memorizza un numero all'interno di 'A' che inizia da 1 e termina a 5 (in questo esempio particolare).

Ogni volta che il codice in esecuzione incontra la "variabile" SUCCESSIVA (NEXT A nel nostro esempio), ritorna alla riga con la parola chiave FOR e incrementa (ne aggiunge una) il numero corrente memorizzato nella variabile chiamata "A". Il codice quindi esegue ogni riga di codice che incontra in seguito (ovvero riga 10 e riga 20) fino a quando non colpisce di nuovo SUCCESSIVO A, quindi torna alla parola chiave FOR e ripete il processo fino a quando il valore di "A" è uguale a 5, quindi fermate.

Quindi, aggiungi nella riga 5 esattamente come mostrato in precedenza, cambia la riga 30, quindi esegui il codice modificato e vedi cosa succede.

Ora, sarebbe utile mostrare qual è il valore corrente di "A". Aggiungi una nuova riga 25:

```
25 Print A
```

Ora esegui di nuovo il tuo codice.

```

5 FOR A=1 TO 5
10 PRINT "RETRO GAMES LIMITED"
20 PRINT "HELLO WORLD!"
25 PRINT A
30 NEXT A
READY.
RUN
RETRO GAMES LIMITED
HELLO WORLD!
1
RETRO GAMES LIMITED
HELLO WORLD!
2
RETRO GAMES LIMITED
HELLO WORLD!
3
RETRO GAMES LIMITED
HELLO WORLD!
4
RETRO GAMES LIMITED
HELLO WORLD!
5
READY.

```

Ora puoi vedere le due righe del testo stampato seguite da un numero che aumenta di una volta ogni volta, ad esempio 1, 2, 3, 4, 5.

È utile vedere il valore memorizzato nella variabile 'A' come ora puoi vedere come cambia all'interno di quel ciclo FOR ... NEXT, aggiungendo uno automaticamente ogni volta che raggiunge NEXT.

Ora rendiamo questo aspetto un po' migliore aggiungendo qualcosa alla riga 25.

```
25 PRINT "VARIABLE 'A' IS NOW";A
```

Il punto e virgola alla fine dice a BASIC di mantenere il cursore sulla stessa linea del testo che sta stampando sullo schermo, e quindi visualizzare il valore della variabile UN.

ESEGUI quest'ultima versione del tuo codice.

```

5 FOR A=1 TO 5
10 PRINT "RETRO GAMES LIMITED"
20 PRINT "HELLO WORLD!"
25 PRINT "VARIABLE 'A' IS NOW";A
30 NEXT A
READY.
RUN
RETRO GAMES LIMITED
HELLO WORLD!
VARIABLE 'A' IS NOW 1
RETRO GAMES LIMITED
HELLO WORLD!
VARIABLE 'A' IS NOW 2
RETRO GAMES LIMITED
HELLO WORLD!
VARIABLE 'A' IS NOW 3
RETRO GAMES LIMITED
HELLO WORLD!
VARIABLE 'A' IS NOW 4
RETRO GAMES LIMITED
HELLO WORLD!
VARIABLE 'A' IS NOW 5
READY.

```

Va un po' meglio. Ora è chiaro cosa significano quei numeri e abbiamo imparato come posizionare il valore di una variabile in un comando PRINT allo stesso tempo!

Ciò conclude quasi l'introduzione a C64 BASIC. Tuttavia, prima di terminare, impariamo come è possibile conservare (salvare) il codice in modo che possa essere eseguito e modificato in futuro.

## Salvataggio e caricamento

Hai alcuni metodi disponibili per salvare i tuoi programmi BASIC, alcuni dei quali dipendono dal prodotto che stai utilizzando.

### **THEC64 / THEC64 Mini:**

Premi il pulsante MENU e seleziona "Salva / carica partita", quindi salva su uno slot disponibile come faresti per una partita o un programma. BASIC ha quattro slot disponibili nello stesso modo in cui fanno tutti i giochi della giostra.

Se una memory stick USB compatibile (formattata su FAT32 e utilizzando MBR) è collegata mentre si utilizza BASIC, è possibile salvare il programma su un disco virtuale memorizzato sulla chiavetta USB:

### **THEC64 Mini :**

È possibile salvare il codice BASIC in un file del disco virtuale sulla chiavetta USB se è stata inserita la chiavetta USB *prima* dell'avvio di BASIC dal carosello. All'avvio di BASIC, THEC64 Mini cerca un file immagine disco specifico sulla memory stick. Se il file non viene trovato, il file *THEC64-drive8.d64* viene creato automaticamente sulla chiavetta USB per te. Sarai quindi in grado di salvare e caricare da e verso questo file da BASIC.

**Tenere presente che su THEC64 Mini, per contenere contemporaneamente il Joystick THEC64, una tastiera USB e una memory stick USB, è necessario collegare un hub USB alimentato separato (non in dotazione) che consente l'accesso a porte USB aggiuntive. Retro Games Ltd non può garantire che tutti gli hub USB funzionino con THEC64 Mini.**

I comandi per salvare sul disco virtuale sono mostrati di seguito.

### **THEC64 :**

Come su THEC64 Mini, il file del disco virtuale *THEC64-drive8.d64* viene creato automaticamente sulla chiavetta USB collegata se non si trova nella radice della chiavetta. Tuttavia, a differenza della Mini, puoi scegliere di inserire un'altra immagine del disco dalla chiavetta USB e salvarla in quella invece se preferisci. Basta premere il pulsante MENU in BASIC, selezionare "Accesso ai media", quindi sfogliare e inserire il disco desiderato premendo FUOCO.

Ora segui le istruzioni per l'uso del comando save come spiegato di seguito.

## SALVA

Con una memory stick USB compatibile collegata, è possibile salvare sul file del disco virtuale utilizzando il comando SAVE to disk standard, ad es

```
SAVE "RGL",8
```

Il nome del file va tra i segni di discorso e può essere quello che ti piace, purché non sia più lungo di 15 caratteri. In questo esempio, è bello e breve. Il numero 8 dopo il nome del file è un numero ID dispositivo per l'unità disco virtuale che utilizza il file del disco virtuale archiviato nella chiavetta USB.



Premere INVIO in seguito per iniziare il salvataggio.

BASIC riporterà "SALVATAGGIO" seguito dal nome del file scelto e, una volta completato, ritorna al prompt READY e al cursore quadrato.

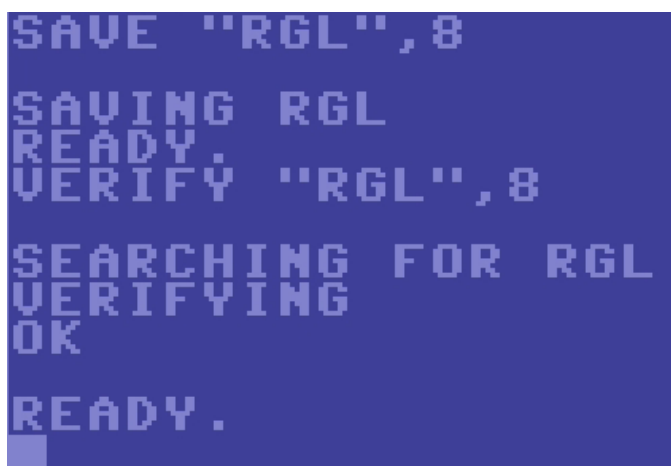
Per salvare su un file esistente con lo stesso nome file, aggiungi @ 0: in primo piano, in questo modo:

```
SAVE "@0:RGL",8
```

Tenere presente che C64 BASIC non fornirà alcun avviso in merito alla sovrascrittura del file durante questa operazione.

## VERIFICARE

Puoi verificare che il salvataggio abbia funzionato usando la parola chiave VERIFY.



```
SAVE "RGL",8
SAVING RGL
READY.
VERIFY "RGL",8
SEARCHING FOR RGL
VERIFYING
OK
READY.
```

```
VERIFY "RGL",8
```

Se tutto va bene, vedrai VERIFYING seguito da OK. Cosa fa confronta il file salvato con ciò che è attualmente nella memoria del computer. Se hai cambiato il tuo programma BASIC in qualche modo da quando hai salvato, VERIFY fallirà.

## CARICARE

Un'altra volta, assicurarsi che lo stesso dispositivo di archiviazione USB sia collegato come prima, quindi digitare il comando seguente in BASIC per ripristinare il programma.

```
LOAD "RGL",8
```

## Cosa c'è sul mio disco?

Da BASIC, puoi guardare un disco e vedere quali file ci sono. Per fare ciò è necessario l'uso del comando LOAD, ma in modo leggermente diverso da prima. Invece di digitare un nome file stiamo usando la variabile riservata \$.

```
LOAD "$",8
```

BASIC segnalerà la RICERCA DI \$ seguita da LOADING e quindi ritornerà nuovamente al prompt READY.

Per vedere cosa c'è sul disco è semplice. Basta digitare il seguente comando:

**LIST**

Invece di elencare il codice BASIC, questa volta il comando mostra ciò che è sul disco.

```

READY.
LOAD "$",8

SEARCHING FOR $
LOADING
READY.
LIST

0 WHH054 " 01 2A)
1 "RGL" PRG
663 BLOCKS FREE.
READY.

```

Nell'esempio sopra, abbiamo solo un programma sul disco ed è il programma che abbiamo appena salvato chiamato RGL. Il programma utilizza fino a 1 blocco sul disco e ci sono 663 blocchi ancora liberi per salvare i programmi in futuro.

Tenere presente che se si carica la directory di un disco e si dispone attualmente di un codice BASIC sul C64, l'elenco \$ sostituirà l'elenco BASIC nella memoria del computer e si perderà il programma (se non lo si è già salvato) .

## Sola lettura

Se non hai una chiavetta USB collegata, ma decidi comunque di provare a salvare sul dispositivo 8, BASIC sembrerà che sia stato salvato con successo ma quando carichi la directory del disco sarà vuota. Questo perché senza archiviazione USB, BASIC utilizza un'immagine del disco che può solo essere letta, non salvata. Il segno distintivo è il nome del disco, che è READONLY anziché THEC64. Il disco è completamente vuoto e non può essere salvato nulla.

Per ovviare a questo, è sempre possibile salvare in uno dei quattro slot di salvataggio, quindi in seguito inserire una memory stick USB in una porta USB di riserva, tornare a BASIC, ripristinare lo slot di salvataggio e quindi salvare sul file del disco virtuale utilizzando il comando SALVA dettagliato in precedenza.

## Per saperne di più

Prima di terminare, quanti comandi BASIC C64 hai usato ora?

<b>addominali</b>	<b>E</b>	<b>ASC</b>	<b>ATN</b>	<b>CHR \$</b>	<b>VICINO</b>	<b>CLR</b>	<b>CMD</b>
<b>CONT</b>	<b>COS</b>	<b>DATI</b>	<b>DEF</b>	<b>DIM</b>	<b>FINE</b>	<b>EXP</b>	<b>FN</b>
<b>PER</b>	<b>FRE</b>	<b>OTTENERE</b>	<b>GET \$</b>	<b>GOSUB</b>	<b>VAI A</b>	<b>SE</b>	<b>INGRESSO</b>

\$ INPUT	INT	LEFT \$	LEN	PERMETTERE	ELENCØ	CARICARE	LOG
MID \$	NUOVO	IL PROSSIMO	NON	SU	APERTO	O	SBIRCIARE
POKE	POS	STAMPA	STAMPA \$	LEGGERE	REM	RISTABILIRE	RITORNO
\$ DESTRA	RND	CORRERE	SALVA	SGN	PECCATO	SPC	SQR
STATO	PASSO	FERMARE	STR \$	SYS	TAB	TAN	POI
TEMPO	TEMPO \$	PER	USR	VAL	VERIFICARE	ASPETTARE	

Sei già in comando con comandi 11/71! Speriamo che questa piccola introduzione ti abbia dato l'interesse per saperne di più. Ci sono molte risorse online disponibili. Inoltre, dai un'occhiata alla nostra pagina dei [collegamenti](#).

Un video di questo breve tutorial introduttivo (usando **THEC64 Mini**) è disponibile anche da guardare qui sotto.

**Programming in C64 BASIC**





[Prodotti](#) [Contatto](#) [Di](#) [politica sulla riservatezza](#)

Copyright © 2019 Retro Games Limited, Retro Games, THEC64 MINI sono marchi registrati di Retro Games Limited, la società registrata in Inghilterra e Galles, 09745704. Tutti i giochi presentati su questo sito Web sono concessi in licenza dai rispettivi proprietari di copyright.