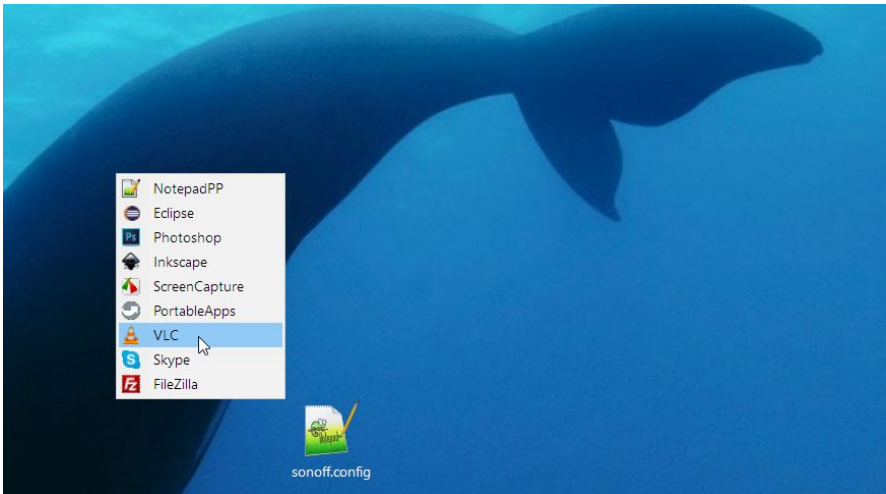


# I 10 strumenti essenziali se lavori con Angular e il front-end

## Parte 2: AutoHotkey per eliminare i task noiosi e ripetitivi



Con AutoHotkey hai un assistente personale, una sorta di copilota nello sviluppo.

Ma non solo: il tuo "collega" si occuperà dei task più noiosi, lasciando a te solo la parte divertente e creativa dello sviluppo.

## Autore

Ciao, sono Salvatore e trovi tutto su di me sul sito <https://linkedin.com/in/salvatoreromeo>. Ho scritto questo mini libro principalmente perché mi piace scrivere e condividere esperienza.

Nella vita mi occupo di sviluppo software fin da quando avevo 12 anni. Oggi è il mio lavoro principale. Insegno Ingegneria del Software all'Università degli Studi di Perugia e un corso di 3 giorni su Angular per le aziende insieme al Codemotion. Il corso è da poco stato trasformato in un libro che puoi trovare su Amazon.

Se vuoi maggiori informazioni su di me contattami pure tramite linkedin o tramite il sito <https://devexp.io>. Lì potrai anche trovare articoli sul front-end, altri libri del gruppo devexp.io e i codici sorgenti pubblicati su Github.

# Sommario

**I 10 strumenti essenziali se lavori con Angular e il front-end**  
**0**

**Parte 2: AutoHotkey per eliminare i task noiosi e ripetitivi** **0**

Autore	1
Sommario	2
Introduzione	4
Gli altri tool che andremo a conoscere nelle prossime puntate	5
Esigenza	5
AutoHotkey	8
Installazione	8
Un semplice script per iniziare	9
API di base	10
Come eseguire un comando da un hotkey	12
Come rimappare un hotkey in un altro hotkey	13
Come inserire snippet veloci con le hotstring	16
Portare una finestra in primo piano e inviare un testo alla finestra	18
Programmi già pronti	20
ALT+TAB con il mouse	20
Menu contestuale nel prompt	20
Copia dell'indirizzo della cartella corrente nella clipboard	23
Qualche trucco	24



## Introduzione

Questo articolo fa parte di una serie di articoli dedicati ai migliori strumenti per velocizzare il processo di sviluppo front-end.

Nello scorsa “puntata” abbiamo visto come usare **postman** per testare, debuggare, documentare e gestire in generale tutte le richieste e le risposte da un backend. In questa puntata ci occuperemo di un tool potentissimo, probabilmente il più potente in assoluto che uso ormai ogni giorno quotidianamente in più contesti. Ho anche un legame affettivo con questo strumento perché è stato il mio primo linguaggio di programmazione :-)

Vedremo una panoramica di **AutoHotkey**, un linguaggio di scripting per automatizzare task noiosi e ripetitivi con poche righe di codice. Condividerò inoltre alcuni degli script creati da me per automatizzare il processo di sviluppo front-end.

Visto che alcuni dei miei lettori sviluppano sotto Mac o Linux, preciso subito che **AutoHotkey funziona esclusivamente su Windows**. E’ un vero peccato che non ci sia una versione anche per gli altri sistemi operativi, ma tra poco vedremo che ci sono programmi simili. Tuttavia, AutoHotkey può comunque essere utile se si ha necessità di creare uno script di automazione per Windows per un amico o un cliente.

AutoHoteky è il motivo principale per cui preferisco programmare sotto Windows anche se spesso devo comunque usare linux nelle aziende.

Esistono alcune alternative per linux e Mac, come IronAHK (<https://github.com/Paris/IronAHK>) e AutoKey (<https://github.com/autokey/autokey>), ma AutoHotkey è probabilmente il più potente e completo strumento per automatizzare qualsiasi cosa sotto Windows e gli altri sistemi elencati non hanno tutte le sue funzionalità.

## Gli altri tool che andremo a conoscere nelle prossime puntate

I tool che andremo a vedere nelle varie puntate sono:

- **Postman** per lavorare con le richieste al server
- **AutoHotkey** per automatizzare qualsiasi task ripetitivo
- **Cmder** per ottimizzare l'uso della linea di comando
- **Http-server** per creare un server velocemente nella cartella corrente
- **Surge** per pubblicare la cartella corrente online con un comando
- **Ngrok** per rendere disponibile un server locale all'esterno
- **ColorPicker** per catturare i colori
- **Chrome** con i suoi strumenti base e avanzati
- **JSON Viewer** per visualizzare oggetti JSON

## Esigenza

In generale AutoHotkey copre tante esigenze:

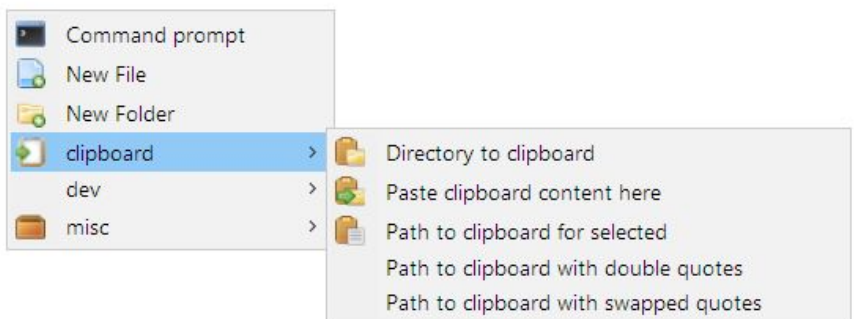
- rimappare hotkey su qualunque programma
- assegnare hotkey a script specifici
- mostrare menu contestuali personalizzati per svolgere funzioni su programmi esistenti

- auto-completamento di testi (es. snippet) a partire da stringhe di pochi caratteri (es. cap[SPAZIO] si trasforma in “ciao e a presto”)

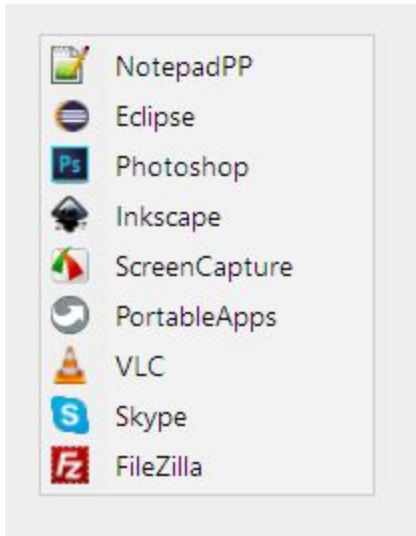
Di seguito ne vedremo alcune, come personalizzarle con script anche complessi e come sfruttarle per il processo di sviluppo.

Ecco qualche caso d’uso che ho implementato con uno script AutoHotkey di poche righe:

- durante lo sviluppo, posso fare lo switch tra *Webstorm* e *Chrome Source* nella stessa riga di codice usando solo un hotkey
- menu contestuali con comandi extra specifici per un particolare programma; ad esempio in Explorer ho un menu contestuale per tutta una serie di operazioni sui file:



- menu generici, ad esempio con la lista dei miei programmi preferiti:



- LEFT\_MOUSE+RIGHT\_MOUSE per chiudere un tab (in chrome, in webstorm, in notepad++)



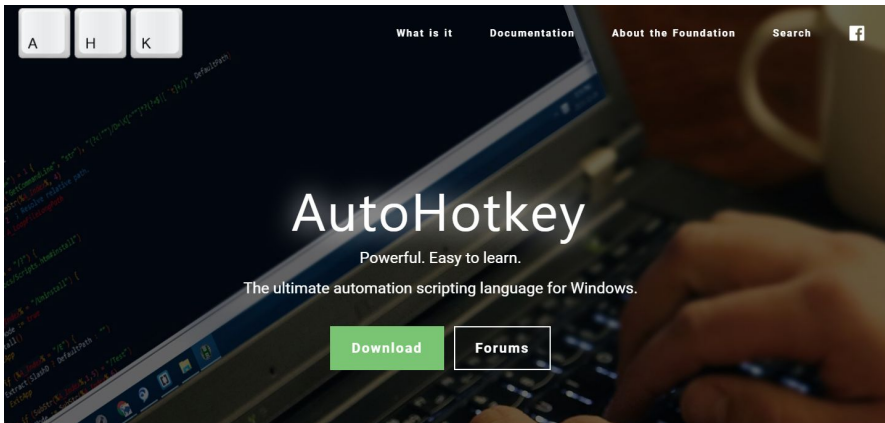
# AutoHotkey

AutoHotkey è uno strumento nato per fare macro, ma poi si è evoluto in un sistema di automazione molto più sofisticato che permette anche di creare interfacce grafiche e di usare librerie DLL per espandere le sue funzioni.

Qui vedremo come sfruttarlo per le funzioni più utili, che sono anche molto facili da implementare per via del linguaggio di scripting semplice ed intuitivo.

## Installazione

Per installare il sistema basta andare sul sito web ufficiale (<https://autohotkey.com/>) e premeri il pulsante Download.



Consiglio all'inizio di usare la versione installabile, ma esiste anche una versione portabile che si scarica con il link al pacchetto zip












Auton

Current version: v1.1.29.01 - June 2, 2018

[Download AutoHotkey Installer](#)

[Download AutoHotkey .zip](#) 

Dopo averlo installato, il pacchetto ha i seguenti file:

 Compiler	Cartella di file
 AutoHotkey	File della Guida HTML com...
 AutoHotkeyA32	Applicazione
 AutoHotkeyU32	Applicazione
 AutoHotkeyU64	Applicazione
 <b>Installer</b>	<b>File AHK</b>
 license	Documento di testo
 Template	File AHK
 WindowSpy	File AHK

Al momento non possiamo ancora eseguire nessun comando. AutoHotkey è infatti un linguaggio di programmazione o meglio un linguaggio di scripting. Tuttavia si tratta di un linguaggio semplicissimo. Lo impareremo presentando vari esempi.

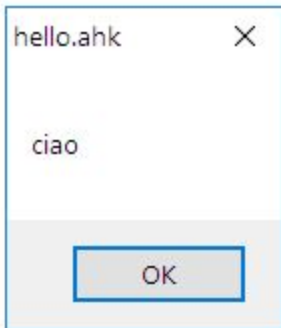
## Un semplice script per iniziare

Per creare uno script, iniziamo con il creare un file con estensione *.ahk*. Creiamo ad esempio il file *bello.ahk* e scriviamo:

```
lang:autohotkey  
MsgBox, ciao
```

(ignoriamo la riga *lang:autohotkey*, che viene aggiunto dal mio editor per indicare che si tratta di codice in linguaggio AutoHotkey appunto)

Trasciniamo quindi il file `hello.ahk` sul programma `AutoHotkeyU64.exe` e vedremo quanto segue:



Come primo script non c'è male, ma ora facciamo sul serio.

## API di base

Per eseguire un comando in AutoHotkey basta richiamare una specifica API su una riga con i rispettivi parametri. Ogni riga rappresenta un comando.

*Msgbox* è una delle tante API di scripting utili per mostrare informazioni. Tutte le API si possono esplorare sul sito

<https://autohotkey.com/docs/AutoHotkey.htm>.

Altre API utili sono:

- Eseguire file:  
`Run, programma.exe`  
 Dove volendo è possibile specificare la cartella dove si trova il programma  
`Run, programma.exe, c:\`
- Inviare stringhe ad una casella di testo o ad un editor, come la stringa *ciao*  
`Send, ciao`  
 o inviare caratteri particolari e combinazioni di tasti ad un programma  
`Send, {SPACE}{TAB}{LEFT}{CTRL DOWN}v{CTRL UP}`  
 Tutti i caratteri sono qui:  
<https://autohotkey.com/docs/KeyList.htm>
- Leggere il contenuto di un file e inserirlo in una variabile, che potremo utilizzare come parametro in un comando  
`FileRead, variabile, file.txt`  
`Send, %variabile%`

Direi che per ora può bastare. Creiamo qualche script utile.

## Come eseguire un comando da un hotkey

Spesso voglio associare ad una combinazione di tasti un comando particolare, come ad esempio eseguire il mio programma preferito, oppure inviare un hotkey premendo semplicemente F12 (che altrimenti non uso come tasto).

Un hotkey si definisce con il nome dell'hotkey seguito dai doppi punti :: . Lo script va inserito subito sotto e termina con un *return*, come se fosse una funzione.

Creiamo uno script *hotkeys.abk* e incolliamo il seguente codice:

```
lang:autohotkey
F12::
    Run, notepad.exe
return
```

PS: notepad non è il mio programma preferito, ma è presente su tutti i PC :-)

## Come rimappare un hotkey in un altro hotkey

Di tutti i programmi che uso, non c'è una volta che gli hotkey definiti dai programmatori per eseguire un comando facciano al caso mio. Preferisco usare i miei.

AutoHotkey permette di rimappare un qualsiasi hotkey con uno alternativo per eseguire lo stesso comando in un particolare programma.

Proviamo a mappare l'Hotkey CTRL+1 con CTRL+N solo per il programma Chrome.

Nel linguaggio, CTRL va scritto con il simbolo ^, SHIFT con + e ALT con !.

L'Hotkey CTRL+1 si scrive quindi ^1.

Modifichiamo il file precedente e aggiungiamo il seguente codice:

```
lang:autohotkey  
  
#IfWinExist ahk_class Chrome_WidgetWin_1  
^1::^n
```

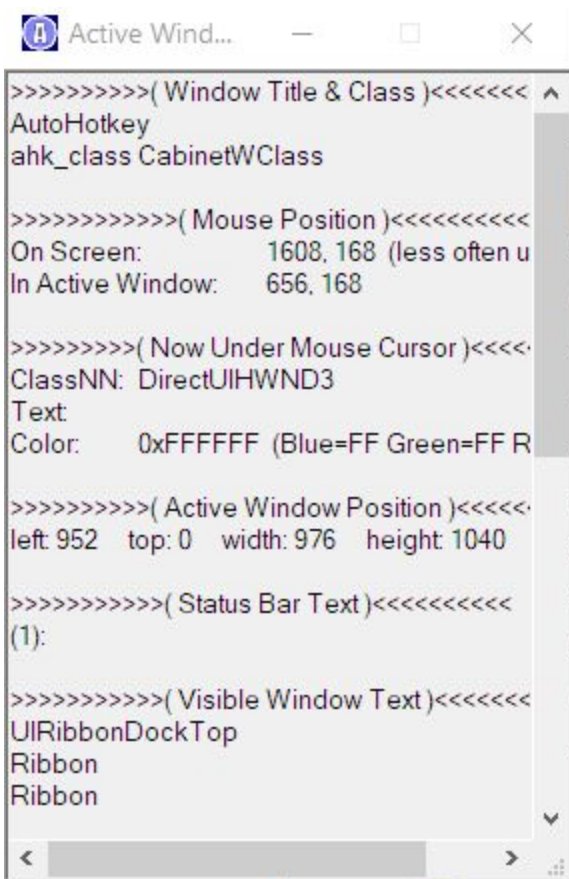
La seconda istruzione dello script mappa l'hotkey. La mappatura di un hotkey (CTRL+1 in questo caso) si può fare in una sola linea di codice usando due volte il carattere : seguito dal secondo hotkey (CTRL+N in questo caso).

L'istruzione *#IfWinExist* specifica che gli hotkey di seguito sono validi solo per il programma indicato con l'ID specificato.

L'ID è una stringa; *ahk\_class Chrome\_WidgetWin\_1* identifica Chrome.

L'ID di explorer, ad esempio, è *ahk\_class CabinetWClass*.

In generale per trovare l'ID di un programma si usa *au3\_spy.exe*, sempre disponibile, dopo aver installato AutoHotkey, nella cartella della installazione.



Possiamo in generale usare *abk\_class IDPROGRAMMA* con le API che lavorano sulle finestre.



## Come inserire snippet veloci con le hotstring

Autohotkey permette di definire delle stringhe brevi che, quando scritte in qualunque editor, si espandono in snippet. Si chiamano hotstring. La sintassi è simile agli hotkey, ma una hotstring si definisce facendo precedere con i `::` la stringa in fase di definizione.

Proviamo a fare in modo che quando scriviamo la stringa *clog* e premiamo lo spazio, automaticamente venga scritto *console.log()*. Niente di più semplice:

1. Creiamo un file hotstrings.ahk
2. Aggiungiamo il seguente codice (sempre ignorando `lang:autohotkey`):

```
lang:autohotkey  
::clog:: console.log(“”)
```

3. Salviamo e trasciniamo il file sul programma AutoHotkeyU64.exe come in precedenza per eseguire il file.

Proviamo quindi a scrivere *clog* e premere lo spazio... e vualà!! *clog* è stato trasformato in *console.log*.

Possiamo anche sofisticare lo script per eseguire istruzioni più complesse, come ad esempio inviare *console.log(“”)* e poi la freccia a sinistra due volte, che è generalmente quello che si fa quando si

scrive *console.log*. In questo modo si ha il cursore in mezzo agli apici pronti per scrivere il messaggio:

```
lang:autohotkey
::clog2::
    Send,console.log(""){left}{left}
return
```

Eseguiamo lo script e proviamo a scrivere *clog2* e spazio in qualunque editor. Molto utile :-)

## Portare una finestra in primo piano e inviare un testo alla finestra

Proviamo ora a portare Chrome in primo piano e aprire un URL in una nuova scheda. Per mettere a fuoco la barra degli indirizzi Chrome prevede di usare l'Hotkey CTRL+L. Possiamo sfruttare un hotkey esistente per creare macro più utili.

Creiamo quindi un nuovo script col codice:

```
lang:autohotkey

^!::
    WinActivate, ahk_class Chrome_WidgetWin_1
    WinWaitActive, ahk_class Chrome_WidgetWin_1
    Send, ^t
    Send, ^l
    Send, https://devexp.io{Enter}
return
```

I comandi *WinActivate* e *WinWaitActive* permettono di attivare una specifica finestra, che si richiama tramite il suo ID. Le istruzioni successive mettono a fuoco la barra degli indirizzi, inviano il sito <http://devexp.io> e infine viene simulata la pressione del tasto INVIO.



## Programmi già pronti

Come vi avevo promesso, ho sviluppato molti script che mi aiutano nello sviluppo software. Ve ne allego alcuni di seguito.

### ALT+TAB con il mouse

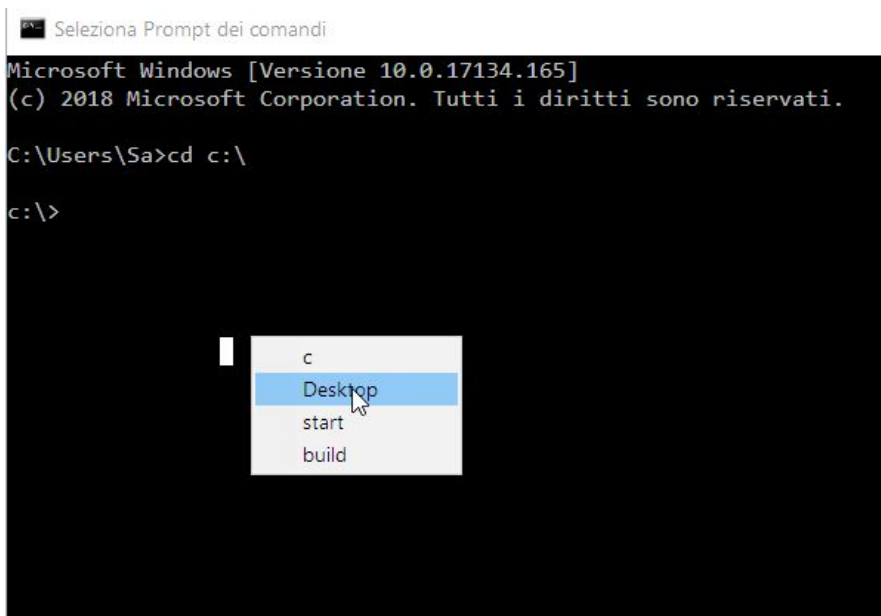
Avete un mouse con un quarto tasto vicino al pollice? Io uso questo tasto per tornare indietro nel browser, ma se è aperto un altro programma che non è il browser, allora uso questo tasto per premere ALT+TAB.

Trovate lo script qui:

<https://github.com/devexp-io/codici-articoli/blob/master/autohotkey/logitech.ahk>

### Menu contestuale nel prompt

Con questo script avrete un vostro menu contestuale da personalizzare per eseguire i vostri comandi più comuni nel prompt. Per richiamarlo uso il tasto centrale del mouse, che nel prompt non ha alcuna funzione:



```
Seleziona Prompt dei comandi
Microsoft Windows [Versione 10.0.17134.165]
(c) 2018 Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\Sa>cd c:\

c:\>
```

Premendo *Desktop* ad esempio mi sposterò sul desktop inviando il comando

*cd C:\Users\Sa\Desktop*

Se aprite lo script, le prime linee sono le seguenti:

```
lang:autohotkey
c=cd c:\
Desktop=cd C:\Users\Sa\Desktop
start=npm start
build=ng build --prod
```

che corrispondono proprio alle voci di menu (a sinistra) e ai comandi da lanciare per quella specifica voce (a destra dopo il

simbolo =). Potete modificarle con i vostri comandi preferiti. Questo script è particolare perché oltre alla logica contiene anche le informazioni per creare il menu, tutto in un unico file.

Analogamente qui trovate uno script per eseguire i programmi preferiti sempre personalizzando le prime righe dello script.



```
lang:autohotkey
```

```
NotepadPP=..\apps\Notepad++Portable\Notepad.exe
```

```
Eclipse=P:\dev\eclipse\eclipse.exe
```

```
Inkscape=..\apps\InkscapePortable\InkscapePortable.exe
```

```
ScreenCapture=..\apps\screen-capture\FSCapture.exe
```

```
PortableApps=..\apps\PortableApps\Start.exe
```

```
VLC=..\apps\VLCPortable\VLCPortable.exe
```

```
Skype=P:\Dropbox\apps\SkypePortable\SkypePortabe.exe
```

```
e
```

```
FileZilla=P:\Dropbox\apps\FileZillaPortable\FileZilla.exe
```

```
#NoTrayIcon
```

```
SetWorkingDir, %A_ScriptDir%
```

```
return
```

```
^q::
```

```
...
```

Si tratta sempre di un menu che include anche l'icona del programma. Dovete personalizzare i percorsi con i vostri programmi ovviamente, ma vi assicuro che è molto meglio di molti launcher.

Io lo richiamo con l'hotkey *CTRL+Q* ( *^q::* alla riga 17), ma potete personalizzare anche questo hotkey.

Trovate gli script rispettivamente ai link:

<https://github.com/devexp-io/codici-articoli/blob/master/autorun/cmd-menu.ahk>

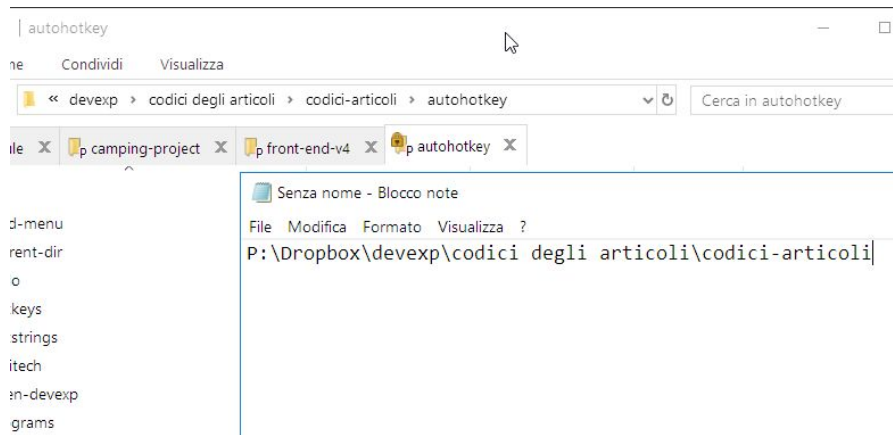
<https://github.com/devexp-io/codici-articoli/blob/master/autorun/programs.ahk>

## Copia dell'indirizzo della cartella corrente nella clipboard

Infine uno scriptino utilissimo: cliccando col tasto sinistro nel titolo di una cartella vi ritroverete l'indirizzo della cartella nella



## clipboard:



Trovate lo script qui:

<https://github.com/devexp-io/codici-articoli/blob/master/autohotkey/current-dir.ahk>

## Qualche trucco

Quello che cerco di trasmettere sempre con i miei articoli è un po' della mia esperienza. Spesso si tratta di tecniche che migliorano l'esperienza del processo di sviluppo.

Anche per sviluppare script Autohotkey è utile non perdere tempo nel processo salva-trascina-riesegui. Vi allego perciò un codice semplice semplice da aggiungere ad ogni vostro script. Questo codice vi permetterà di auto-aggiornare e auto-avviare lo script ogni volta che lo salvate:

```
lang:autohotkey
```

```
SetTimer, UPDATEDSCRIPT, 1000
```

```
UPDATEDSCRIPT:
```

```
FileGetAttrib,attribs,%A_ScriptFullPath%
```

```
IfInString,attribs,A
```

```
{
```

```
FileSetAttrib,-A,%A_ScriptFullPath%
```

```
SplashTextOn,,,Updated script,
```

```
Sleep,300
```

```
Reload
```

```
}
```

```
Return
```

## Conclusioni

Come abbiamo visto AutoHotkey è un tool potentissimo per automatizzare ogni aspetto di Windows. E' un vero peccato che non sia presente un tool altrettanto potente per altri sistemi operativi, ma se si usa Windows o si deve creare uno script per un amico, AutoHotkey può sicuramente esserci d'aiuto.

Qui abbiamo visto solo alcuni script base molto semplici, utili per iniziare ed esplorare le potenzialità, ma questo tool può davvero diventare una sorta di co-pilota nella programmazione di ogni giorno.

Buono scripting e se hai suggerimenti, commenti o feedback in generale non esitare a scrivermi.