

# AHK Hotkeys Guide

## Inhaltsverzeichnis:

1. [RETURN \(Ein- oder Mehrzeilig?\)](#)
2. [Dedizierte Hotkeys in Programmen](#)
3. [Dedizierte Hotkeys in OBS](#)
4. [Tasten, Tastenkombinationen & Text](#)
5. [Programme Starten](#)
6. [Programme Starten & Schließen](#)

**GitHub:** <https://github.com/Takeo1337/DIY-Stream-Deck>

**Autohotkey Dokumentation DE:** <https://ahkde.github.io/docs/AutoHotkey.htm>

**Autohotkey Dokumentation EN:** <https://autohotkey.com/docs/AutoHotkey.htm>

### **Warnung:**

Aufgrund von Kompatibilitätsproblemen ist das Belegen der "Pause" Taste nicht zu empfehlen. NumLock wird eventuell mit ausgeführt!

Geschrieben von: **Takeo1337**

YouTube: <https://www.youtube.com/channel/UCmzQWRMCKtdnuH9CDuslQrw>

# 1. RETURN (Ein- oder Mehrzeilig?)

Wenn ein Befehl in einer Zeile geschrieben ist, muss kein `return` angegeben werden.

Ist jedoch der auszuführende Code in mehr als einer Zeile, so muss danach ein `return` angegeben werden. Ansonsten werden darunter stehende Befehle solange ausgeführt, bis irgendwann auf ein `return` getroffen wird.

Alles was mehr als ein einfacher Befehl ist, sollte immer in mehreren Zeilen geschrieben werden!

```
[HOTKEY] :: [BEFEHL]
```

oder

```
[HOTKEY] ::
```

```
[BEFEHL]
```

```
return
```

Beispiel:

```
numpad5::SendInput, {F13}
```

oder

```
numpad5::
```

```
SendInput, {F13}
```

```
return
```

[Dokumentation return](#)

## 2. Dedizierte Hotkeys in Programmen

Dedizierte Hotkeys in bestimmten Programmen können ziemlich hilfreich sein. Besonders wenn die Tasten für nichts anders verwendet werden. F13 - F24 sind solche Tasten. Sie existieren, jedoch gibt es auf den meisten Tastaturen keine F-Tasten nach F12. Das können wir uns zu nutze machen.

Mit `SendInput` können Tasten "gedrückt" werden.

```
[HOTKEY] :: SendInput, { [KEY] }
```

Beispiel:

```
numpad5 :: SendInput, { F13 }
```

Die zu sendenden Tasten müssen immer in geschweiften Klammern `{ }` geschrieben werden. Es wird so behandelt als hätte man F13 auf der Tastatur gedrückt.

[Dokumentation SendInput](#)

## 3. Dedizierte Hotkeys in OBS

Man sollte meinen das in OBS die gleichen Regeln gelten. Das sollte es auch, allerdings ist OBS oft nicht im Fokus, da z.B. Spiele zum aufzeichnen oder Streamen im Fokus stehen. Dies kann dazu führen das der `SendInput` OBS nicht erreicht.

Mit `controlSend` können wir allerdings **gezielt** einen Input an bestimmte Programme, wie z.B. OBS senden.

```
[HOTKEY] :: controlSend, , { [KEY] }, ahk_class [AHK_CLASS]
```

Beispiel:

```
numpad5 :: controlSend, , { F15 }, ahk_class Qt5QWindowIcon
```

Um die `ahk_class` heraus zu finden öffnen wir das Window Spy Programm (findet ihr im Startmenü unter AutoHotkey oder im Installationsverzeichnis) und klicken auf das gewünschte Fenster (OBS). Im ersten Kasten wird die `ahk_class` ausgegeben. Die `ahk_class` für OBS ist `Qt5QWindowIcon`.

Die `ahk_class` dient dazu das genaue Fenster anzusteuern.

[Dokumentation ControlSend](#)

## 4. Tasten, Tastenkombinationen & Text

Einen einfachen Text zu senden oder einfache Tastenkombinationen auszuführen kann im Workflow sehr hilfreich sein.

Mit `Send` kann Text gesendet werden oder Tasten(kombinationen) gedrückt werden.

```
[HOTKEY] :: Send { [KEY] }
```

```
[HOTKEY] :: Send [KEY / KEY COMBO]
```

```
[HOTKEY] :: Send [TEXT]
```

Beispiel:

```
numpad5 :: Send { F13 }
```

```
numpad5 :: Send ^a
```

```
numpad5 :: Send Hallo Welt
```

Das erste Beispiel drückt F13.

Das zweite Beispiel drückt STRG + a.

Das letzte Beispiel gibt den Text "Hallo Welt" aus.

Eine Liste der Modifizier Symbole und alles weitere findet ihr in der [Dokumentation für Send](#)!

[Dokumentation Send](#)

## 5. Programme Starten

Um mit Autohotkey Programme zu starten wird der `Run` Befehl verwendet.  
Mit dem `Run` Befehl lassen sich allerdings auch einfache Dateien öffnen.

```
[HOTKEY] :: Run, [PFAD\EXE]
```

oder

```
[HOTKEY] ::  
Run, [PFAD\EXE]  
return
```

Beispiel:

```
numpad5 :: Run, C:\Windows\System32\Taskmgr.exe
```

oder

```
numpad5 ::  
Run, C:\Windows\System32\Taskmgr.exe  
return
```

[Dokumentation "Run"](#)

OBS verhält sich (bei mir zumindest) nicht wie vermutet. Um OBS ohne Fehler zu starten muss eine Verknüpfung ausgeführt werden.

In meinem Fall:

```
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\OBS Studio\OBS Studio (64bit)
```

## 6. Programme Starten & Schließen

Ein Programm per Hotkey zu starten und (falls es offen ist) auch wieder zu beenden ist sehr angenehm. Um dies zu schaffen muss überprüft werden ob das Fragliche Programm offen ist, wenn nein wird es gestartet, wenn ja wird es beendet.

```
[HOTKEY] ::
IfWinNotExist, ahk_class [AHK_CLASS] ahk_exe [PROGRAMM.EXE]
{
    Run, [PFAD\EXE]
}
IfWinExist, ahk_class [AHK_CLASS] ahk_exe [PROGRAMM.EXE]
{
    WinClose, ahk_class [AHK_CLASS] ahk_exe [PROGRAMM.EXE]
}
return
```

**IfWinNotExist** überprüft ob das Fenster **nicht** vorhanden ist. Wenn das Fenster **nicht** vorhanden ist wird das Programm ausgeführt.

**IfWinExist** überprüft ob das Fenster vorhanden ist. Falls es vorhanden ist wird das Programm mit **WinClose** beendet.

Um die **ahk\_class** und die **ahk\_exe** heraus zu finden öffnen wir das Window Spy Programm (findet ihr im Startmenü unter AutoHotkey oder im Installationsverzeichnis) und klicken auf das gewünschte Fenster. Im ersten Kasten wird die **ahk\_class** und die **ahk\_exe** ausgegeben.

Natürlich kann auch darauf geprüft werden ob der Prozess existiert oder nicht. Das Resultat ist das gleiche.

[Dokumentation ahk\\_class](#)

[Dokumentation ahk\\_exe](#)

[Dokumentation IfWin\[Not\]Exist](#)

[Dokumentation WinClose](#)

[Dokumentation Prozessüberprüfung](#)

