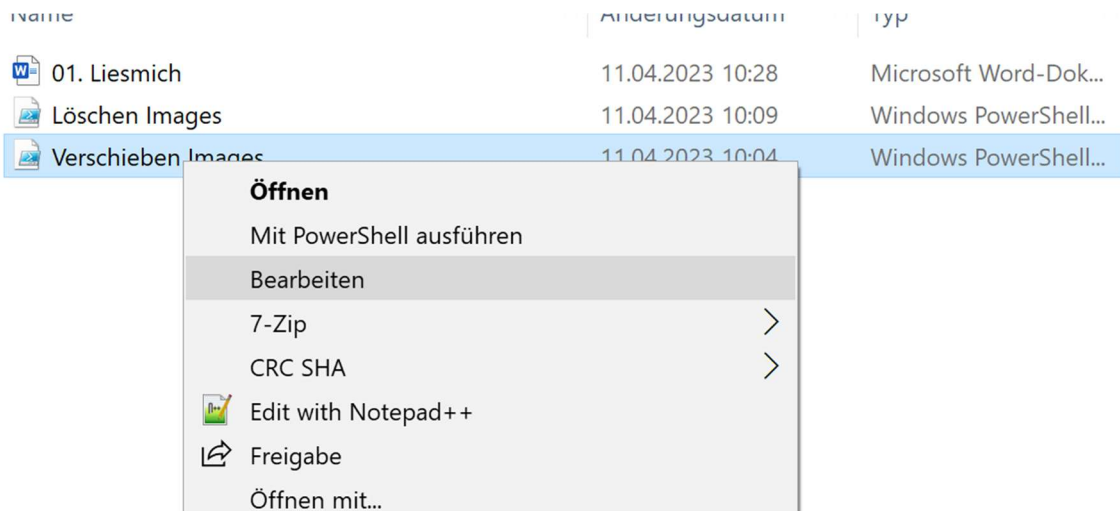
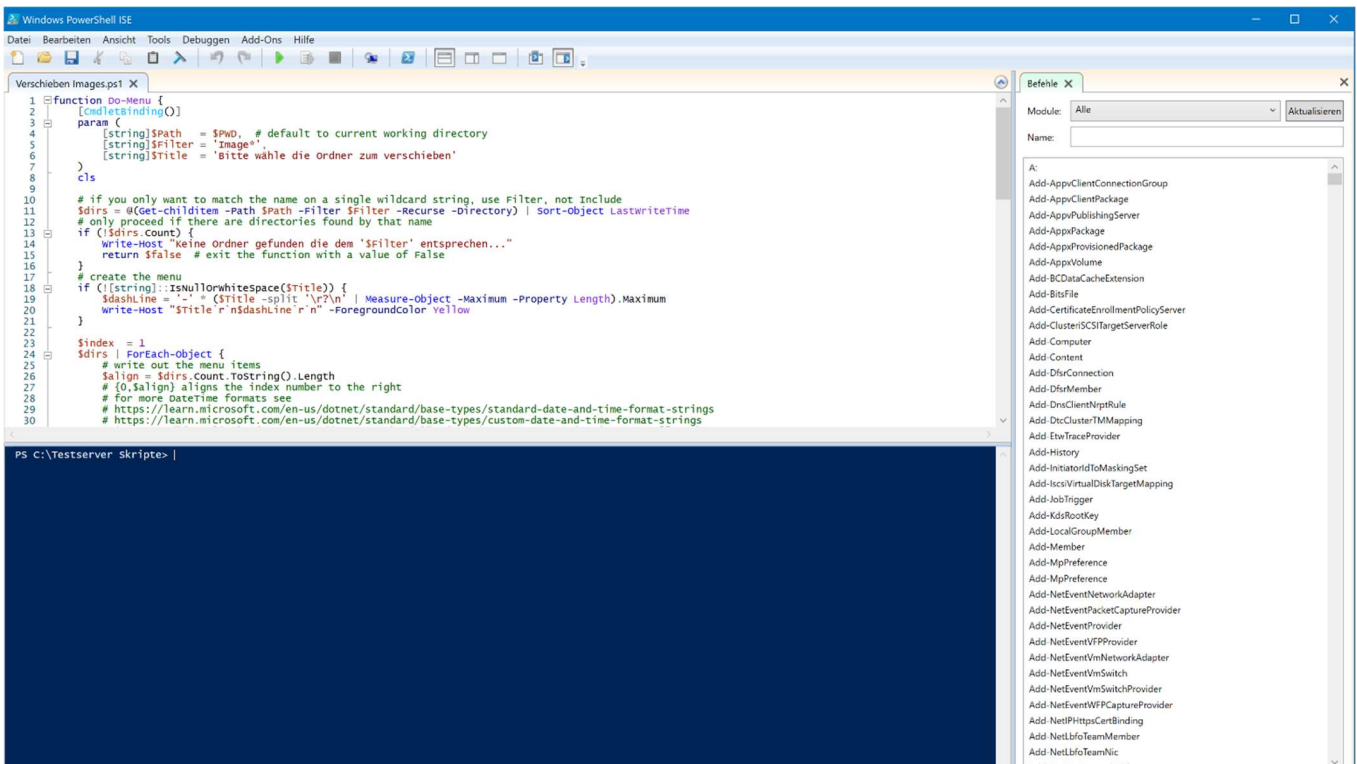


## LIESMICH KURZANLEITUNG

Starte das jeweilige Skript indem du es mit der rechten Maustaste anklickst und dann **bearbeiten** wählst.



Dann öffnet sich das Script in der Powershell ISE Konsole.



Nun klicke mit der linken Maustaste in das blaue Fenster rein um da den Cursor zu setzen.

Da das Skript auf einem Remote Computer ausgeführt werden soll, muss erst die Verbindung zu diesem gemacht werden. Dies geschieht durch folgenden Befehl:

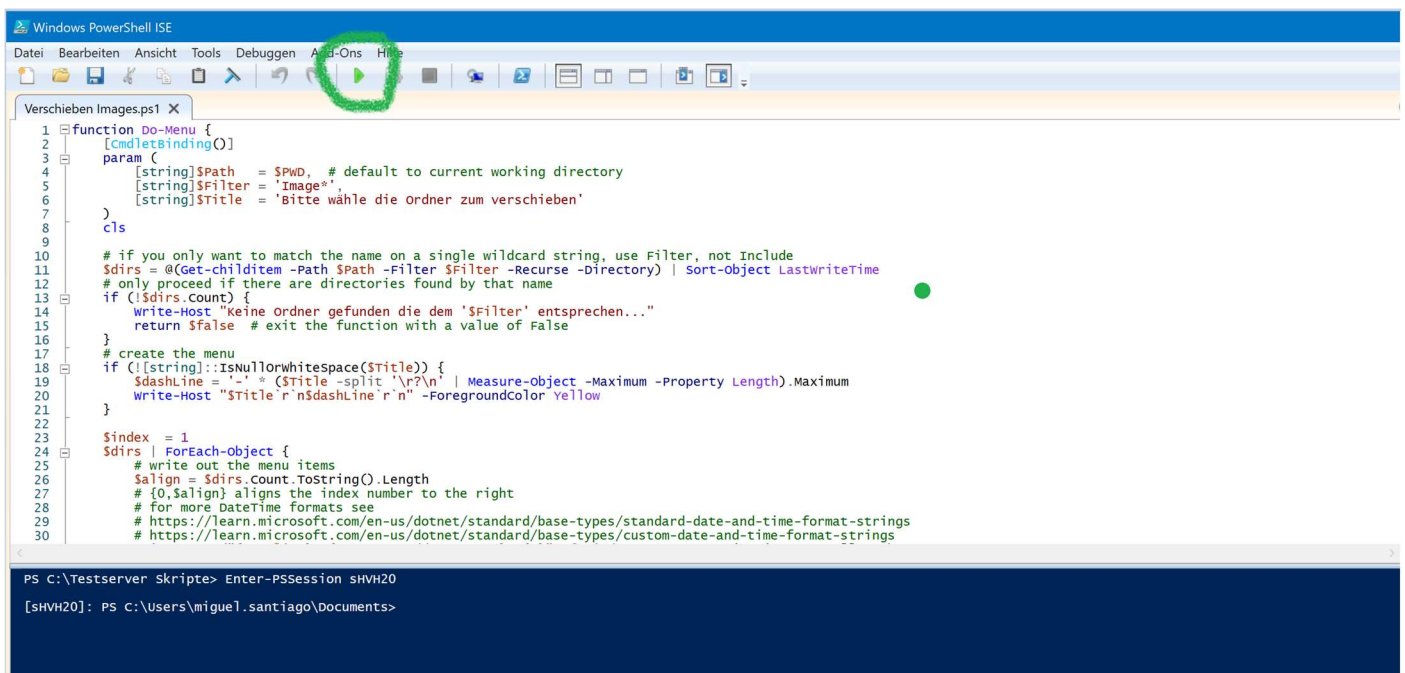
Enter-PSSession + **Computername**

In unserem Fall wäre der Befehl wie folgt: **Enter-PSSession SHVH20** und **ENTER** um den Befehl zu bestätigen.

Da der Befehl funktioniert hat, siehst du daran, dass die nächste Zeile mit **[SHVH20]** beginnt.

Warte einen Moment, bis der rote Stopp Taster nicht mehr rot leuchtet.

Klicke nun auf die grüne Playtaste um das Skript laufen zu lassen und folge dann den Anweisungen die im blauen Bildschirm der Konsole folgen.



The screenshot shows the Windows PowerShell ISE interface. The script editor displays a PowerShell script named 'Verschieben Images.ps1'. The script defines a function 'Do-Menu' that interacts with the user to move files. The console window at the bottom shows the command 'Enter-PSSession SHVH20' being executed, followed by the prompt '[SHVH20]: PS C:\Users\miguel.santiago\documents>'.

```
1 function Do-Menu {
2     [CmdletBinding()]
3     param (
4         [string]$Path = $PWD, # default to current working directory
5         [string]$Filter = 'Image*',
6         [string]$Title = 'Bitte wähle die Ordner zum verschieben'
7     )
8     cls
9
10    # if you only want to match the name on a single wildcard string, use Filter, not Include
11    $dirs = @(Get-Childitem -Path $Path -Filter $Filter -Recurse -Directory) | Sort-Object LastWriteTime
12    # only proceed if there are directories found by that name
13    if ($dirs.Count) {
14        Write-Host "Keine ordner gefunden die dem '$Filter' entsprechen..."
15        return $false # exit the function with a value of False
16    }
17    # create the menu
18    if ([string]::IsNullOrEmpty($Title)) {
19        $dashline = '-' * ($Title -split '\r?\n' | Measure-Object -Maximum -Property Length).Maximum
20        Write-Host "$Title`r`n$dashline`r`n" -ForegroundColor Yellow
21    }
22
23    $index = 1
24    $dirs | ForEach-Object {
25        # write out the menu items
26        $align = $dirs.Count.ToString().Length
27        # {0,$align} aligns the index number to the right
28        # for more DateTime formats see
29        # https://learn.microsoft.com/en-us/dotnet/standard/base-types/standard-date-and-time-format-strings
30        # https://learn.microsoft.com/en-us/dotnet/standard/base-types/custom-date-and-time-format-strings
```

PS C:\Testserver Skripte> Enter-PSSession SHVH20  
[SHVH20]: PS C:\Users\miguel.santiago\documents>