

ekolo dale

```
C:\Users\Administrator PowerShell  
1 int main(int argc, char** argv, char** envp) {  
2     if (argc <= 2)  
3 }  
4  
5  
6  
7  
8
```

00:30 / 00:45



Windows Terminal

Schritt-für-Schritt-Anleitung

Microsoft hat die Kommandozeile neu entworfen. Unter dem Namen *Windows Terminal* ist dabei eine Applikation herausgekommen, welche viele Shells (z.B. CMD, PowerShell und WSL) unter einer Benutzeroberfläche vereint.

In diesem Artikel wird beschrieben, wie man das *Windows Terminal* installiert, an die eigenen Bedürfnisse anpasst – wir möchten die Arbeit mit Git angenehmer machen - und das Terminal in Visual Studio Code einbindet.

Schritt-für-Schritt-Anleitung

Voraussetzungen:

- [PowerShell](#) (mindestens V6) ist installiert.
- [Git für Windows](#) ist installiert.
- [Visual Studio Code](#) ist installiert.

Neben *Windows Terminal* werden wir auch folgende Erweiterungen installieren:

- Posh-Git ist ein PowerShell-Modul, welches den Status eines Git-Repositories anzeigen kann und die Git-Kommandos beim Tippen ergänzen kann.
- [Oh-My-Posh](#) ist eine Erweiterung, welche verschiedene Themes (Farben) anbietet.
- Get-ChildItemColor ist eine Erweiterung, welche die Rückgabe des Cmdlets einfärbt.
- [Cascadia-Font](#) für die korrekte Darstellung der Zeichen für den Git-Status.

1. *Windows Terminal* installieren.
Der empfohlene Weg ist die Installation über den [Store](#).
2. Erweiterungen installieren.
PowerShell als Administrator öffnen, danach die Policy ändern:

```
Set-ExecutionPolicy RemoteSigned -Scope CurrentUser -Confirm
```

Danach werden die Erweiterungen installiert:

```
PowerShellGet\Install-Module posh-git -Scope CurrentUser -Force  
Install-Module oh-my-posh -Scope CurrentUser  
Install-Module -AllowClobber Get-ChildItemColor
```

- Erweiterungen beim Start von *Windows Terminal* laden.
Damit die Module beim Start des Terminals geladen werden, müssen sie in eine Profile-Datei geschrieben werden.

Dazu ruft man zuerst mithilfe von Visual Studio Code diese Datei auf:

```
code $profile
```

Sollte die Datei noch nicht existieren, erzeugt Visual Studio diese automatisch. Diese Datei muss um die folgenden Zeilen ergänzt werden:

```
Import-Module posh-git  
Import-Module oh-my-posh  
Import-Module Get-ChildItemColor  
  
Set-Theme Avit
```

Hier wird das Theme Avit geladen, andere Themes können bei [Oh-My-Posh](#) eingesehen werden.

- Font installieren und verwenden.
Nach dem [Download](#) des Zip-Paketes muss dieses entpackt werden. Wir verwenden hier *CascadiaCodePL.ttf*, deswegen muss diese (z.B. mithilfe der rechten Maustaste) installiert werden.

Jetzt müssen wir *Windows Terminal* starten und *Settings* aufrufen. In dieser Datei machen wir unter *profiles* → *defaults* folgende Einträge:

```
...  
  
"profiles":  
{  
  "defaults":  
  {  
    // Put settings here that you want to apply to all profiles.  
    "fontFace": "Cascadia Code PL",  
    "fontSize": 10  
  },  
}
```

```
    "list":  
    [  
        ...  
    ]  
}  
...
```

Damit ist für alle Profile der gleiche Font gesetzt.

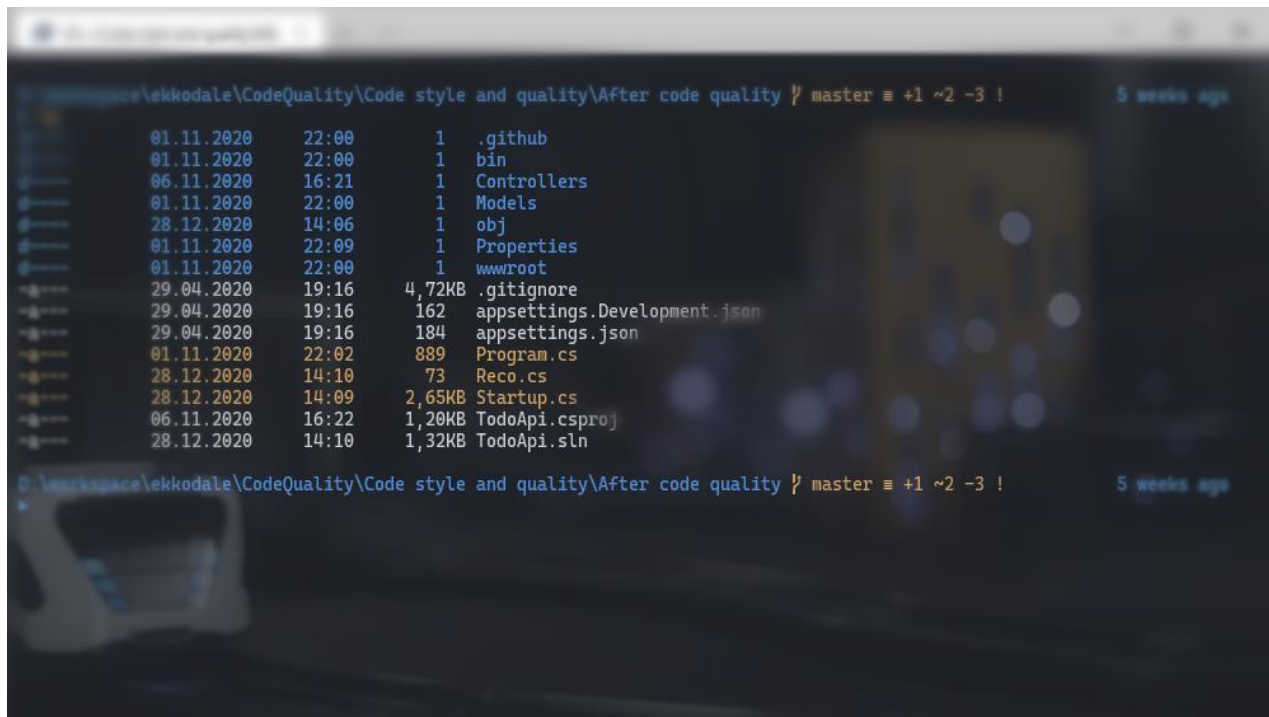
5. Hintergrund von *Windows Terminal* ändern.
Alle Profile sollen das gleiche Hintergrundbild erhalten, deswegen ergänzen wir den Abschnitt *defaults*:

```
...  
  
"profiles":  
  "defaults": {  
    ...  
    "backgroundImage": "C:/images/background.png",  
    "backgroundImageOpacity": 0.3  
  },  
  
...
```

Wer den Acrylic-Effekt von Windows 10 nutzen möchte, fügt noch Folgendes hinzu:

```
"useAcrylic": true,  
"acrylicOpacity": 0.3
```

Mein Terminal sieht jetzt so aus, wenn ich in ein Verzeichnis mit einem Git-Repository navigiere:



```
D:\workspace\ekkodale\CodeQuality\Code style and quality\After code quality > master == +1 ~2 ~3 ! 5 weeks ago
├── .github
├── bin
├── Controllers
├── Models
├── obj
├── Properties
├── wwwroot
├── 4,72KB .gitignore
├── 162 appsettings.Development.json
├── 184 appsettings.json
├── 889 Program.cs
├── 73 Reco.cs
├── 2,65KB Startup.cs
├── 1,20KB TodoApi.csproj
└── 1,32KB TodoApi.sln

D:\workspace\ekkodale\CodeQuality\Code style and quality\After code quality > master == +1 ~2 ~3 ! 5 weeks ago
```

Durch eine einfache Änderung des Themes in der *Profile*-Datei lassen sich die Git-Informationen auf verschiedene Weise darstellen.

6. *Windows Terminal* in Visual Studio Code benutzen.

Wenn man Visual Studio Code aufruft und dann *Terminal* → *Neues Terminal* aufruft und zu PowerShell wechselt, sollte alles bereits so aussehen wie im *Windows Terminal*. Sollten die Glyphen nicht richtig angezeigt werden, muss man nur noch in der Datei *settings.json* (aufrufbar z.B. über die Kommando-Palette) den gleichen Font wie vorher eintragen:

```
"terminal.integrated.fontFamily": "Cascadia Code PL"
```

Damit sieht das Terminal in Visual Studio Code so aus:

