

Node.js Update Script für Windows 11 (mit NVM)

Hier ist ein PowerShell-Script, das alles automatisch erledigt:

update-node.ps1

```
# =====  
# Node.js Update Script für Windows 11 (NVM)  
# =====  
  
Write-Host "`nNode.js Update Script gestartet...`n" -ForegroundColor Cyan  
  
# 1. NVM-Liste anzeigen und neueste Version finden  
Write-Host "Suche neueste Node.js Version..." -ForegroundColor Yellow  
  
# NVM list available ausgeben und analysieren  
$nvmOutput = cmd /c "nvm list available" 2>&1  
  
Write-Host "`nVerfügbare Versionen:" -ForegroundColor Gray  
Write-Host $nvmOutput -ForegroundColor DarkGray  
  
# Die erste LTS Version aus der Tabelle extrahieren (erste Zeile nach Header, erste Spalte)  
$lines = $nvmOutput -split "`n"  
$latestLTS = $null  
  
foreach ($line in $lines) {  
    # Suche nach Zeilen die mit einer Versionsnummer beginnen (nach |)  
    if ($line -match '\\s*(\d+\.\d+\.\d+)\s*|') {  
        $latestLTS = $matches[1]  
        break  
    }  
}
```

```

}

# Fallback: Versuche alternative Parsing-Methode
if (-not $latestLTS) {
    foreach ($line in $lines) {
        if ($line -match '(\d+\.\d+\.\d+)') {
            $latestLTS = $matches[1]
            break
        }
    }
}

# Wenn immer noch nichts gefunden, manuelle Eingabe
if (-not $latestLTS) {
    Write-Host "`n⚠ Konnte Version nicht automatisch ermitteln." -ForegroundColor Red
    Write-Host "    Bitte schaue oben in der Tabelle nach der neuesten LTS Version.`n" -
ForegroundColor Yellow
    $latestLTS = Read-Host "    Gib die gewünschte Version ein (z.B. 22.16.0)"
}

if (-not $latestLTS) {
    Write-Host "⚠ Keine Version angegeben. Abbruch." -ForegroundColor Red
    exit 1
}

Write-Host "`n → Ausgewählte Version: $latestLTS" -ForegroundColor Green

# 2. Neueste Version installieren
Write-Host "`n👉 Installiere Node.js $latestLTS..." -ForegroundColor Yellow
cmd /c "nvm install $latestLTS"

# 3. Neue Version aktivieren
Write-Host "`n👉 Aktiviere Node.js $latestLTS..." -ForegroundColor Yellow
cmd /c "nvm use $latestLTS"

# Kurze Pause damit NVM die Änderung übernimmt
Start-Sleep -Seconds 2

# 4. Überprüfen

```

```
Write-Host "`nAktuelle Versionen:" -ForegroundColor Yellow
try {
    $nodeVersion = cmd /c "node -v" 2>&1
    $npmVersion = cmd /c "npm -v" 2>&1
    Write-Host "    Node.js: $nodeVersion" -ForegroundColor Green
    Write-Host "    NPM: $npmVersion" -ForegroundColor Green
} catch {
    Write-Host "    ⚠ Konnte Versionen nicht abfragen - ggf. neues Terminal öffnen" -
ForegroundColor Yellow
}
```

5. NPM selbst aktualisieren

```
Write-Host "`nAktualisiere NPM auf neueste Version..." -ForegroundColor Yellow
cmd /c "npm install -g npm@latest"
```

6. Globale Pakete installieren/aktualisieren

```
Write-Host "`nInstalliere globale Pakete..." -ForegroundColor Yellow
```

```
$packages = @(
    "less",
    "sass",
    "typescript",
    "npm-check-updates",
    "nodemon",
    "live-server"
)
```

```
foreach ($package in $packages) {
    Write-Host "    → Installiere $package..." -ForegroundColor Gray
    cmd /c "npm install -g $package" 2>$null
}
```

7. Zusammenfassung

```
Write-Host "`n" -NoNewline
Write-Host ("=" * 50) -ForegroundColor Cyan
Write-Host "  FERTIG! Zusammenfassung:" -ForegroundColor Green
Write-Host ("=" * 50) -ForegroundColor Cyan
```

```
$finalNode = cmd /c "node -v" 2>&1
```

```
$finalNpm = cmd /c "npm -v" 2>&1

Write-Host "`n  Node.js Version: $finalNode" -ForegroundColor White
Write-Host "  NPM Version: $finalNpm" -ForegroundColor White
Write-Host "`n  Installierte globale Pakete:" -ForegroundColor White
cmd /c "npm list -g --depth=0"

Write-Host "`n  Alles erledigt!`n" -ForegroundColor Green
Write-Host "  Tipp: Öffne ein NEUES Terminal damit alle Änderungen aktiv sind.`n" -
ForegroundColor Yellow
```

☐☐ Einrichtung & Nutzung

Option 1: Direkt ausführen

```
# PowerShell als Administrator öffnen, dann:
Set-ExecutionPolicy -Scope CurrentUser RemoteSigned
.\update-node.ps1
```

Option 2: Als Batch-Datei (Doppelklick)

Erstelle `update-node.bat`:

```
@echo off
PowerShell -ExecutionPolicy Bypass -File "%~dp0update-node.ps1"
pause
```

⚙️ Pakete anpassen

Bearbeite diese Liste im Script nach deinen Wünschen:

```
$packages = @(
    "less",          # LESS Compiler
    "sass",          # SCSS/SASS Compiler
```

```
"typescript",    # TypeScript
"npm-check-updates", # Dependency Updates checken
"nodemon",       # Auto-Restart bei Änderungen
"live-server"    # Dev-Server mit Hot-Reload
# Füge hier weitere hinzu:
# "gulp-cli",
# "webpack-cli",
# "eslint",
# "prettier"
)
```

Wichtig

- **Als Administrator ausführen** (Rechtsklick → "Als Administrator ausführen")
- NVM für Windows muss bereits installiert sein
- Falls nicht: <https://github.com/coreybutler/nvm-windows/releases>

Soll ich das Script noch anpassen oder weitere Pakete hinzufügen?

Revision #1

Created 2026-05-31 14:11:02 UTC by art10m

Updated 2026-05-31 14:12:48 UTC by art10m