

# Label

Ein `Label` ist ein **reines Anzeige-Control** für Text.

Es dient zur Beschreibung, Anzeige oder Strukturierung von UI-Inhalten.

Ein `Label` ist **nicht für direkte Benutzerinteraktion vorgesehen**, kann jedoch auf Ereignisse wie Mausklicks reagieren.

## Label erstellen

```
# Klassisch
$label = New-Object System.Windows.Forms.Label

# .NET-Style
$label = [System.Windows.Forms.Label]::new()
```

## Eigenschaften - *Label*

### Eigenschaften

- **Property** - *Standardwert*  
Beschreibung oder Erläuterung der Eigenschaft

---

- **Text** - ""  
Angezeigter Text
- **AutoSize** - *\$false*  
Passt Größe automatisch an den Inhalt an
- **TextAlign** - *TopLeft*  
Ausrichtung des Textes innerhalb des Labels
- **BorderStyle** - *None*  
Rahmenstil des Labels
- **Dock** - *None*  
Layout innerhalb des Parent-Containers

- **Anchor** - (*Top, Left*)  
Verhalten bei Größenänderung des Containers
- **Padding** - (*0, 0, 0, 0*)  
Innenabstand
- **Margin** - (*3, 3, 3, 3*)  
Außenabstand
- **UseMnemonic** - *\$true*  
Aktiviert Zugriff über Alt+Key (z.B. &Datei)

## geerbt von `Control`

- **Width** - *100*  
Breite des Labels
- **Height** - *23*  
Höhe des Labels
- **Size** - (*Width=100, Height=23*)  
Größe des Labels
- **Location** - (*0, 0*)  
Position innerhalb des Containers
- **ForeColor** - *SystemColors.ControlText*  
Textfarbe (systemabhängig)
- **BackColor** - *Transparent*  
Hintergrundfarbe (übernimmt Parent-Hintergrund)
- **Font** - *Microsoft Sans Serif, 8.25pt*  
Standard-Systemschrift
- **Enabled** - *\$true*  
Aktiviert das Label
- **Visible** - *\$true*  
Sichtbarkeit des Labels
- **TabIndex** - *0*  
Reihenfolge beim Durchtabben
- **TabStop** - *\$false*  
Label ist standardmäßig nicht per Tab erreichbar
- **Name** - *""*  
Interner Name des Controls

`Text`

**Typ:** `system.String`

Die Eigenschaft **Text** legt den angezeigten Inhalt eines Labels fest.

```
$label.Text = "Hallo Welt"
```

```
# Mehrzeilig
```

```
$label.Text = "Zeile 1`nZeile 2"
```

PowerShell macht implizite Konvertierung:

```
$Label.Text = 123
```

funktioniert, weil es intern zu `"123"` wird.

---

## AutoSize

**Typ:** `System.Boolean`

Die Eigenschaft **AutoSize** bestimmt, ob sich das Label automatisch an die Größe seines Inhalts anpasst.

```
# Automatische Größe
```

```
$label.AutoSize = $true
```

---

## Size

**Typ:** `System.Drawing.Size`

```
# Feste Größe
```

```
$label.Size = [System.Drawing.Size]::new(200, 30)
```

---

## TextAlign

**Typ:** `System.Drawing.ContentAlignment`

```
# Text zentrieren
```

```
$label.TextAlign = "MiddleCenter"
```

---

# Events - Label

## Events

- **Even**  
Wird ausgelöst, ...

---

## Interaktion (*User Input*)

- **Click**  
... wenn das Label mit der Maus angeklickt wird
- **DoubleClick**  
... wenn das Label doppelt angeklickt wird
- **MouseEnter**  
... wenn der Mauszeiger das Label betritt
- **MouseLeave**  
... wenn der Mauszeiger das Label verlässt
- **MouseMove**  
... wenn der Mauszeiger innerhalb des Labels bewegt wird
- **MouseDown**  
... wenn eine Maustaste auf dem Label gedrückt wird
- **MouseUp**  
... wenn eine Maustaste auf dem Label losgelassen wird
- **MouseHover**  
... wenn der Mauszeiger für kurze Zeit auf dem Label verweilt

## Zustand / Inhalt

- **TextChanged**  
... wenn sich der Text des Labels ändert
- **EnabledChanged**  
... wenn sich der Enabled-Status des Labels ändert
- **VisibleChanged**  
... wenn sich die Sichtbarkeit des Labels ändert

## Layout

- **LocationChanged**  
... wenn sich die Position des Labels ändert
- **SizeChanged**  
... wenn sich die Größe des Labels ändert

## Darstellung (Rendering)

- **Paint**

... wenn das Label neu gezeichnet wird (z. B. bei Aktualisierung oder Überdeckung)

## Aussehen

- **FontChanged**

... wenn sich die Schriftart des Labels ändert

- **ForeColorChanged**

... wenn sich die Textfarbe des Labels ändert

- **BackColorChanged**

... wenn sich die Hintergrundfarbe des Labels ändert

```
$label.Add_*  
    param($sender, $e)
```

- `$sender` → das Label selbst (= `$this`)
- `$e` → EventArgs (je nach Event unterschiedlich)

# Interaktionen (User Input)

## Click / DoubleClick / MouseDown

```
# Click  
$label.Add_Click({  
    param($sender, $e)  
  
    Write-Host "Auf das Label wurde geklickt"  
})  
  
# DoubleClick  
$label.Add_DoubleClick({  
    param($sender, $e)  
  
    Write-Host "Auf das Label wurde doppelt geklickt"  
})  
  
# MouseDown
```

```
$label.Add_MouseDown({
    param($sender, $e)

    Write-Host "Auf das Label wurde entweder mit links oder rechts geklickt"
})
```

## MouseEnter / MouseLeave

Mit MouseEnter und MouseLeave sind Hover-Effekte möglich

```
$label.Add_MouseEnter({
    $this.ForeColor = "Red"
})

$label.Add_MouseLeave({
    $this.ForeColor = "Black"
})
```

## MouseMove

Mausbewegung innerhalb vom Label

```
$label.Add_MouseMove({
    param($sender, $e)
})
```

## TextChanged

Wird ausgelöst, wenn sich der Text ändert

```
$label.Add_TextChanged({
    param($sender, $e)

    Write-Host "Neuer Text: $($sender.Text)"
})
```

---

# Tipps & Tricks - *Label*

---

# Typische Stolperfallen

- **Text wird abgeschnitten**
    - `AutoSize` vergessen oder `Size` zu klein
  - **Mehrzeiliger Text funktioniert nicht**
    - kein ``n` verwendet oder Größe zu klein
  - **Label wirkt "unsichtbar"**
    - `ForeColor` = `BackColor`
  - **Klick funktioniert nicht wie erwartet**
    - Label ist nicht für Interaktion gedacht
- 

# Mentales Modell

Ein `Label` ist **UI-Deko mit Bedeutung**.

Es erklärt dem Benutzer, was andere Controls tun oder zeigt Status an.

---

# Wann sinnvoll?

- Beschriftung von Eingabefeldern
  - Statusanzeigen
  - Überschriften / Struktur
- 

# Wann vermeiden?

- Wenn Interaktion erwartet wird → Button nutzen
  - Wenn komplexe Darstellung nötig ist → anderes Control prüfen
  -
- 

Revision #14

Created 2026-04-01 17:52:52 UTC by John-Andreas Borinas

Updated 2026-04-22 00:42:25 UTC by John-Andreas Borinas