

# VBA (Visual Basic for Applications)

- [Letzte beschriebene Zeile in Excel ausgeben](#)
- [Excel Tabelle Zeile für Zeile einlesen](#)
- [Fortschrittsbalken in PowerPoint einfügen](#)
- [Überprüfen ob Ordner / Datei vorhanden ist \(VBA\)](#)

# Letzte beschriebene Zeile in Excel ausgeben

Wenn du in Excel die letzte beschriebene Zeile benötigst, kannst du folgenden Code verwenden. Du musst dort nur das Tabellenblatt ändern. Wahlweise kannst du auch eine Long Variable erstellen, um größere Zahlenbereiche zu speichern.

```
Dim i As Integer  
i = Worksheets(TabellenBlatt).Cells(Rows.Count, 1).End(xlUp).Row
```

```
Dim l As Long  
l = Worksheets(TabellenBlatt).Cells(Rows.Count, 1).End(xlUp).Row
```

---

## Erklärung

Excel greift zuerst das Worksheet ab, dort verwendet er die Zellenposition der letzten Zeile in Spalte 1. Mit dem **xlUp** zählt er alle Zeilen nach oben und gibt die Zahl dann aus.

# Excel Tabelle Zeile für Zeile einlesen

## Einleitung

Wenn du in Excel größere Tabellen bearbeiten möchtest, bietet es sich an, diese automatisch durcharbeiten zu lassen. Dieses kannst du über eine **For Schleife** realisieren. Diese ermittelt erst die letzte Zeile und geht dann Zeile für Zeile durch. In der Schleife selbst kannst du dann den Code angeben, den du benötigst, um die Daten zu verarbeiten.

## Code

```
Sub FormatData()  
    Application.ScreenUpdating = False  
  
    Dim y As Long  
    Dim ws As Worksheet: Set ws = Worksheets("Data")  
    Dim numberOfRows As Long: numberOfRows = ws.Cells(Rows.Count, 1).End(xlUp).Row  
  
    For y = 2 To numberOfRows  
        Debug.Print(ws.Range("A" & y).Value)  
    Next y  
  
    Application.ScreenUpdating = True  
  
End Sub
```

## Code Erklärung

Als Erstes werden 2 Variablen deklariert und 1 Objekt instanziiert und Bildschirm Aktualisierungen deaktiviert. Die Variable `numberOfRows` erhält die Zeilennummer der letzten Zeile. Damit hat die Schleife ein Limit.

Im nächsten Schritt wird die Schleife erstellt. Die Schleife fängt ab Zeile 2 an, da die Tabelle eine Überschrift hat und diese nicht mit eingelesen werden muss.

In der Schleife selbst gibt er nur den Inhalt der aktuellen Zelle in **Spalte A** aus.

Als Letztes werden dann die Bildschirm-Aktualisierungen wieder aktiviert.

Wenn du den Code in ein Modul einfügst, hast du von jedem Tabellenblatt Zugriff drauf.

# Fortschrittsbalken in PowerPoint einfügen

## Einleitung

Wenn wir Präsentationen in PowerPoint erstellen, benötigen wir unter Umständen einen Fortschrittsbalken in der Präsentation. PowerPoint hat nur leider keine eigene Funktion, einen Fortschrittsbalken einzufügen. Dazu verwenden wir ein Skript, um dieses Problem zu umgehen.

Mit dem Skript wird dann ein Fortschrittsbalken generiert, der sich auf den Folien langsam vergrößert. Die Größe wird Prozentual anhand der erstellten Folien generiert.

## Anwendung

### Präsentation ohne Deckblatt

```
Sub CreateProgressBar()  
    On Error Resume Next  
    With ActivePresentation  
        For X = 1 To .Slides.Count  
            .Slides(X).Shapes("PB").Delete  
            Set s = .Slides(X).Shapes.AddShape(msoShapeRectangle, _  
                0, .PageSetup.SlideHeight - 8, _  
                X * .PageSetup.SlideWidth / .Slides.Count, 8)  
            s.Fill.ForeColor.RGB = RGB(0, 0, 255)  
            s.Name = "PB"  
        Next X  
    End With  
End Sub
```

### Präsentation mit Deckblatt

```
Sub CreateProgressBar()  
    On Error Resume Next  
    With ActivePresentation  
        For X = 2 To .Slides.Count
```

```
.Slides(X).Shapes("PB").Delete  
Set s = .Slides(X).Shapes.AddShape(msoShapeRectangle, _  
0, .PageSetup.SlideHeight - 8, _  
X * .PageSetup.SlideWidth / .Slides.Count, 8)  
s.Fill.ForeColor.RGB = RGB(0, 0, 255)  
s.Name = "PB"  
Next X:  
End With  
End Sub
```

Die Farben können wir mithilfe der **RGB Farbcodes** in Zeile **9** anpassen.

# Überprüfen ob Ordner / Datei vorhanden ist (VBA)

## Einleitung

Falls wir überprüfen möchten, ob ein bestimmtes Verzeichnis vorhanden ist, können wir das nachstehende **Code-Konstrukt** verwenden. Dieses gibt dann einen **boolean Wert** zurück, ob der Pfad vorhanden ist, oder nicht.

## Code Ordner

```
Sub checkPath
  Dim path = "C:\Users\test\Desktop\"
  if CreateObject("Scripting.FileSystemObject").FolderExists(path) Then
    ' Ordner ist vorhanden
  else
    ' Ordner ist nicht vorhanden!
  end if
End Sub
```

## Code Datei

```
Sub checkPath
  Dim path = "C:\Users\test\Desktop\"
  if CreateObject("Scripting.FileSystemObject").FileExists(path) Then
    ' Datei ist vorhanden
  else
    ' Datei ist nicht vorhanden!
  end if
End Sub
```