

## Kapitel VI

# Rechtschreibfehler für den Ausdruck formatieren

### In diesem Kapitel:

Wie die Lösung funktioniert

3

Zusammenfassung

8

Das Word-Objektmodell haben Sie bereits in Kapitel 5, und UserForms in Kapitel 15 kennen gelernt. Im vorliegenden Kapitel finden Sie nun eine Lösung, in der die folgenden Themen veranschaulicht werden. Der programmtechnische Umgang mit Rechtschreibfehlern wurde bislang nicht behandelt und ist neu.

---

**Im Blickpunkt:**

Document-Objekt

Add-Methode

SaveAs-Methode

Document-Objekt:SpellingErrors

Range-Objekt

ConvertToTable-Methode

Font

Information

Selection-Objekt

GoTo-Methode

HomeKey

Word-interne Textmarken

UserForm

Kontrollkästchen

Combobox-Steuerelement

Tag

---

Seit Word 95 können Rechtschreibfehler im Text mit wellenförmigen Unterstrichen gekennzeichnet werden. Diese werden jedoch nicht ausgedruckt. Gelegentlich besteht der Bedarf – vor allem in Lernumgebungen –, diese auszudrucken oder in einer Liste zusammenzutragen.

Der vorliegende Lösungsvorschlag zeigt, wie das Objekt-Modell von Word Rechtschreibfehler handhabt. Grammatikfehler können auf ähnliche Weise bearbeitet werden.



---

Die Beispieldatei *BspVI\_01.dot* finden Sie auf der CD-ROM zum Buch im Ordner *\Beispiele\KapVI*. Speichern Sie die Datei im *StartUp*-Ordner von Word, um das Werkzeug beim Starten von Word automatisch mitzuladen.

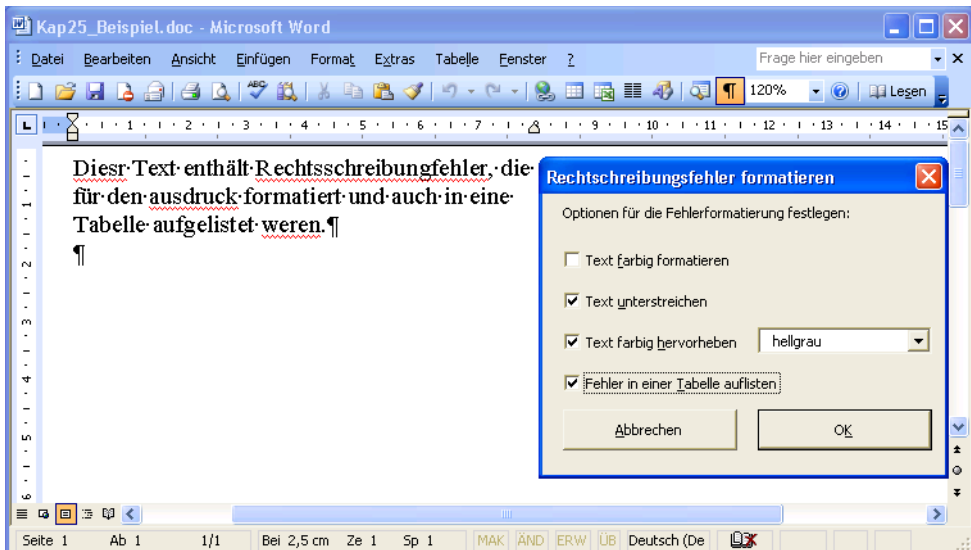
---

# Wie die Lösung funktioniert

Beim Aufruf des Werkzeugs, um Rechtschreibfehler zu formatieren, wird ein Formular wie das in Abbildung VI.1 eingeblendet. Darin stehen dem Anwender vier Optionen zur Auswahl, von denen er so viele aktivieren darf, wie er möchte.

Beim Aktivieren einer der Optionen, mit denen Text farbig formatiert oder hervorgehoben werden kann, wird eine ComboBox mit einer Liste von 16 Farben eingeblendet. Diese sind in der Reihenfolge ihres WdColorIndex-Werts aufgelistet; die Farbe wird anhand seiner Position in der Liste bestimmt.

Abbildg. VI.1 Optionen für die Formatierung der Rechtschreibfehler im Ausdruck



Der Code für das Formular steht in Listing VI.1. Die erste Prozedur – *RechtschreibfehlerOptionen* –, die sich in einem gewöhnlichen Modul befindet, blendet das Formular ein und speichert die Benutzereinstellungen nach deren Bestätigung für den weiteren Verlauf. Das Formular wird dann geschlossen und aus dem Speicher entfernt, bevor die Prozedur für die Formatierung (Listing VI.2) aufgerufen wird.

## HINWEIS

Mehr über die Arbeit mit VBA-Formularen (UserForms) finden Sie in Kapitel 15 beschrieben.

Die weiteren Prozeduren dieses Listings steuern das Formular. Diese sind:

- Zwei Ereignisprozeduren für die Kontrollkästchen für die Schriftfarbe und die Hervorhebung. Sie blenden die entsprechenden ComboBoxen ein und aus.
- Zwei Ereignisprozeduren für die Schaltflächen *Abbrechen* und *OK*. In beiden Fällen wird das Formular ausgeblendet. Der Wert des Tag-Steuerelements hält fest, welche Schaltfläche betätigt wurde.

- *UserForm\_Initialize* wird ausgeführt beim Laden des Formulars. Die Listen der ComboBoxen werden gefüllt, und es wird sichergestellt, dass diese nicht sichtbar sind.
- Die letzte Prozedur – *FarbenAuflisten* – füllt die ComboBoxen. Sie steht in einem gewöhnlichen Modul, das mit Option Private Module deklariert wurde. Diese Deklaration sorgt dafür, dass die darin enthaltenen Prozeduren und globalen Variablen außerhalb dieses Projekts nicht sichtbar sind (es handelt sich um das VBA-Gegenstück zur Deklaration Friend in der klassischen Visual Basic-Sprache).

**Listing VI.1** Prozeduren rund um das Formular

```
'Code in einem gewöhnlichen Modul
Public Sub RechtschreibfehlerOptionen()
    Dim frm As frmRSFehlerFormatieren

    Set frm = New frmRSFehlerFormatieren
    frm.Show
    If frm.Tag = "1" Then
        bFarbigFormatieren = frm.chkFarbig
        bMitUnterstrichFormatieren = frm.chkUnterstreichen
        bMitHervorhebungFormatieren = frm.chkHervorheben
        bInTabelleSchreiben = frm.chkTabelle
        lFarbeText = frm.cboFarbeText.ListIndex
        lFarbeHervorhebung = frm.cboFarbeHervorheben.ListIndex
    End If
    Unload frm
    Set frm = Nothing
    RechtschreibfehlerFuerDenAusdruckFormatieren
End Sub

'Code im Klassenmodul des UserForms
Private Sub chkFarbig_Click()
    SichtbarkeitFestlegen chkFarbig, cboFarbeText
End Sub

Private Sub chkHervorheben_Click()
    SichtbarkeitFestlegen chkHervorheben, cboFarbeHervorheben
End Sub

Private Sub cmdAbbrechen_Click()
    Me.Tag = "0"
    Me.Hide
End Sub

Private Sub cmdOK_Click()
    Me.Tag = "1"
    Me.Hide
End Sub

Private Sub UserForm_Initialize()
    FarbenAuflisten Me.cboFarbeText
    FarbenAuflisten Me.cboFarbeHervorheben
    Me.cboFarbeText.Visible = False
    Me.cboFarbeHervorheben.Visible = False
End Sub
```

## Listing VI.1 Prozeduren rund um das Formular (Fortsetzung)

```

Option Private Module
Public Sub FarbenAuflisten(cbo As MSForms.ComboBox)
    cbo.AddItem ""
    cbo.AddItem "schwarz"
    cbo.AddItem "blau"
    cbo.AddItem "türkis"
    cbo.AddItem "hellgrün"
    cbo.AddItem "rosa"
    cbo.AddItem "rot"
    cbo.AddItem "gelb"
    cbo.AddItem "weiß"
    cbo.AddItem "dunkelblau"
    cbo.AddItem "dunkelgrünblau"
    cbo.AddItem "grün"
    cbo.AddItem "violett"
    cbo.AddItem "dunkelrot"
    cbo.AddItem "dunkelgelb"
    cbo.AddItem "dunkelgrau"
    cbo.AddItem "hellgrau"
End Sub

Public Sub SichtbarkeitFestlegen(chk As MSForms.CheckBox, ctl As MSForms.Control)
    If chk.Value = True Then
        ctl.Visible = True
    Else
        ctl.Visible = False
    End If
End Sub

```

Das Betätigen der Schaltfläche *OK* löst die Prozedur *RechtschreibfehlerFuerDenAusdruckFormatieren* in Listing VI.2 aus. Zuerst wird kontrolliert, ob das aktuelle Dokument Rechtschreibfehler enthält. Wenn nicht, wird die Ausführung abgebrochen.

Sonst wird das Dokument gespeichert, dann nochmals (im gleichen Pfad) unter einen neuen Namen, der um den Text im Konstantwert *strNeueDateiEndung* erweitert wurde.

Da der Umfang der durch die Rechtschreibprüfung erkannten Fehler durch den Arbeitsspeicher beschränkt ist, empfiehlt es sich, einzelne Bereiche statt des ganzen Dokuments durcharbeiten. Deshalb wird von der ersten bis zur letzten Seite »geblättert« und der Inhalt einer Range-Variablen zugewiesen.

Jedem Rechtschreibfehler im jeweiligen Bereich wird die vom Anwender festgelegte Formatierung zugewiesen. Falls er die Fehler in einer Tabelle auflisten möchte, werden Seitenzahl, fehlerhafter Text sowie der Satz, in dem sich der Fehler befindet, einer zeichengetrennten Zeichenkette hinzugefügt. Diese wird schließlich in ein neues Dokument eingefügt und in eine Tabelle (Abbildung VI.2) konvertiert.

Abbildg. VI.2 Liste der im Dokument gefundenen Rechtschreibfehler

Seite	fehlerhafter Text	Kontext
1	Dieser	Dieser Text enthält Rechtschreibfehler, die für den Ausdruck formatiert und auch in eine Tabelle aufgelistet werden.
1	Rechtsschreibungfehler	Dieser Text enthält Rechtschreibfehler, die für den Ausdruck formatiert und auch in eine Tabelle aufgelistet werden.
1	ausdruck	Dieser Text enthält Rechtschreibfehler, die für den Ausdruck formatiert und auch in eine Tabelle aufgelistet werden.
1	werden	Dieser Text enthält Rechtschreibfehler, die für den Ausdruck formatiert und auch in eine Tabelle aufgelistet werden.

Die Abbildung VI.3 veranschaulicht die Wirkung der Optionen, fehlerhaften Text zu unterstreichen sowie hervorzuheben.

Abbildg. VI.3 Die Rechtschreibfehler wurden unterstrichen und hervorgehoben

Dieser Text enthält Rechtsschreibungfehler, die für den ausdruck formatiert und auch in eine Tabelle aufgelistet werden.

Listing VI.2 Die Prozeduren, um Rechtschreibfehler zu formatieren und in einer Tabelle auflisten

```
Option Private Module

Public Const strNeueDateiEndung As String = "_RSFehler"
Public Const strTrennzeichen As String = "|"

Public bFarbigFormatieren As Boolean '= False
Public bMitUnterstrichFormatieren As Boolean '= True
Public bMitHervorhebungFormatieren As Boolean '= False
Public bInTabelleSchreiben As Boolean '= False
Public lFarbeText As Long '= 0
Public lFarbeHervorhebung As Long '= 0

Sub RechtschreibfehlerFuerDenAusdruckFormatieren()
    Dim doc As Word.Document
    Dim strDokNeuerName As String
    Dim rng As Word.Range
    Dim lSeitenZaehler As Long
    Dim rngRSFehler As Range
    Dim strTabellenDaten As String

    Set doc = ActiveDocument
    'Die Arbeit speichern, damit keine Daten verloren gehen
    doc.Save
```

Listing VI.2 Die Prozeduren, um Rechtschreibfehler zu formatieren und in einer Tabelle auflisten (Fortsetzung)

```

' Falls keine Rechtschreibfehler vorhanden sind, abbrechen.
If doc.SpellingErrors.Count > 0 Then
    MsgBox "Keine Rechtschreibfehler gefunden."
    Exit Sub
End If

'Dokument für den formatierten Fehler unter einem anderen Namen
'im gleichen Pfad speichern wie das ursprüngliche.
strDokNeuerName = NeuerDateiName(doc, strNeueDateiEndung)
doc.SaveAs FileName:=doc.Path & "\" & strDokNeuerName, AddToRecentFiles:=False

'Auf der ersten Seite anfangen
Selection.HomeKey wdStory
Set rng = ActiveDocument.Range

'Die Rechtschreibprüfung hat Mühe in einem großen Dokument mit vielen Fehlern.
'Deshalb werden einzelne Seiten abgearbeitet.
For lSeitenZaehler = 1 To doc.Range.Information(wdNumberOfPagesInDocument)
    'Den Text einer Seite einem Bereich mit Hilfe der \Page Textmarke zuweisen
    Set rng = Selection.Bookmarks("\Page").Range

    'Die Rechtschreibfehler den gewünschten Einstellungen entsprechend formatieren.
    For Each rngRSFehler In rng.SpellingErrors
        With rngRSFehler
            'Die Eigenschaft ColorIndex ist gültig in allen Word-Versionen
            If bFarbigFormatieren = True Then .Font.ColorIndex = lFarbeText
            If bMitUnterstrichFormatieren = True Then .Font.Underline = wdUnderlineDotDash
            If bMitHervorhebungFormatieren = True Then _
                rngRSFehler.HighlightColorIndex = lFarbeHervorhebung

            ' Falls die Fehler in einer Tabelle aufgelistet werden sollen, werden sie in einer
            ' Zeichenkette gesammelt, die am Schluss in eine Tabelle konvertiert wird.
            If bInTabelleSchreiben = True Then
                strTabellenDaten = strTabellenDaten & CStr(lSeitenZaehler) & _
                    strTrennzeichen & rngRSFehler.Text & strTrennzeichen & _
                    TrimEndParas(rngRSFehler.Sentences(1).Text) & vbCrLf
            End If
        End With
    Next rngRSFehler
    Selection.GoTo What:=wdGoToPage, Which:=wdGoToNext
Next lSeitenZaehler

'Die Tabelle in einem neuen Dokument erstellen
If bInTabelleSchreiben = True Then
    Dim rngTabelle As Word.Range
    Dim tbl As Word.Table
    Dim docFehlerLog As Word.Document

    Set docFehlerLog = Documents.Add
    Set rngTabelle = docFehlerLog.Range

    'Die letzte Absatzmarke abschneiden
    strTabellenDaten = Left(strTabellenDaten, Len(strTabellenDaten) - 1)

    'Spaltenüberschriften hinzufügen und die zeichengetrennte Liste einfügen

```

**Listing VI.2** Die Prozeduren, um Rechtschreibfehler zu formatieren und in einer Tabelle auflisten (*Fortsetzung*)

```

    rngTabelle.Text = "Seite" & strTrennzeichen & "Fehlbarer Text" & _
        strTrennzeichen & "Kontext" & vbCrLf & strTabellenDaten

    '... und den Bereich in eine Tabelle umwandeln
    Set tbl = rngTabelle.ConvertToTable(Separator:=strTrennzeichen, Numcolumns:=3)

    'Die erste Zeile fett formatieren.
    tbl.Rows(1).Range.Font.Bold = True
End If
End Sub

'Die Datei-Endung ersetzen und den neuen Dateinamen zurückgeben
Function NeuerDateiName(doc As Word.Document, ext As String) As String
    Dim s As String
    Dim loc As Long

    s = doc.Name
    loc = InStr(s, ".doc")
    s = Left(s, loc - 1)
    NeuerDateiName = s & ext
End Function

'Absatzmarken am Ende einer Zeichenkette entfernen
Function TrimEndParas(s As String) As String
    Do While Asc(Right(s, 1)) = 13
        s = Left(s, Len(s) - 1)
    Loop
    TrimEndParas = s
End Function

```

## Zusammenfassung

Dieser Lösungsvorschlag stellt den Umgang mit Rechtschreibfehlern (SpellingErrors) vor. Er veranschaulicht zudem, wie Farben in einer ComboBox-Liste aufgenommen und ausgewertet werden.