

# Excel - eMail Adressen mit Hilfe von Regulären Ausdrücken auf Gültigkeit überprüfen

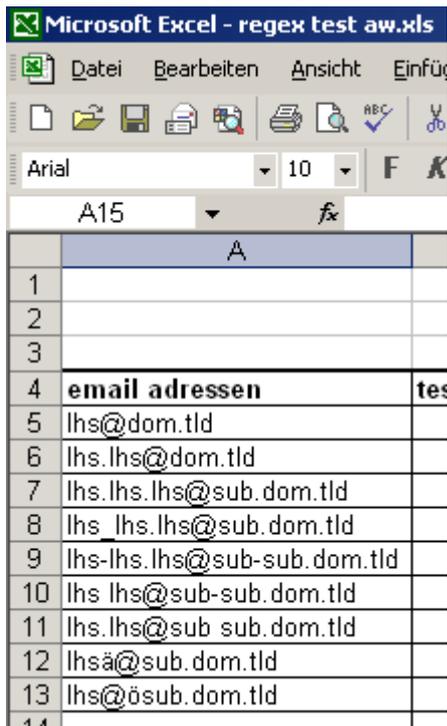
email adressen	eMail Adresse gültig?
lhs@dom.tld	WAHR
lhs.lhs@dom.tld	WAHR
lhs.lhs.lhs@sub.dom.tld	WAHR
lhs_lhs.lhs@sub.dom.tld	WAHR
lhs-lhs.lhs@sub-sub.dom.tld	WAHR
lhs lhs@sub-sub.dom.tld	FALSCH
lhs.lhs@sub sub.dom.tld	FALSCH
lhsä@sub.dom.tld	FALSCH
lhs@ösub.dom.tld	FALSCH
	FALSCH
	FALSCH
	FALSCH
	FALSCH
a@b.de	WAHR
	FALSCH
	FALSCH

Mit ein wenig Vorbereitung kann man auch unter Excel mit Regulären Ausdrücken in Formeln arbeiten. z.B. um Eingabe-Formate zu überprüfen, Ersetzungen vorzunehmen oder einfach nur um Zeichenfolgen aufzufinden. Diese Dokumentation zeigt wie man sich sein Excel so einrichten kann damit es uns dabei unterstützt fehlerhafte EMail Adressen in einer Liste leichter ausfindig machen zu können.

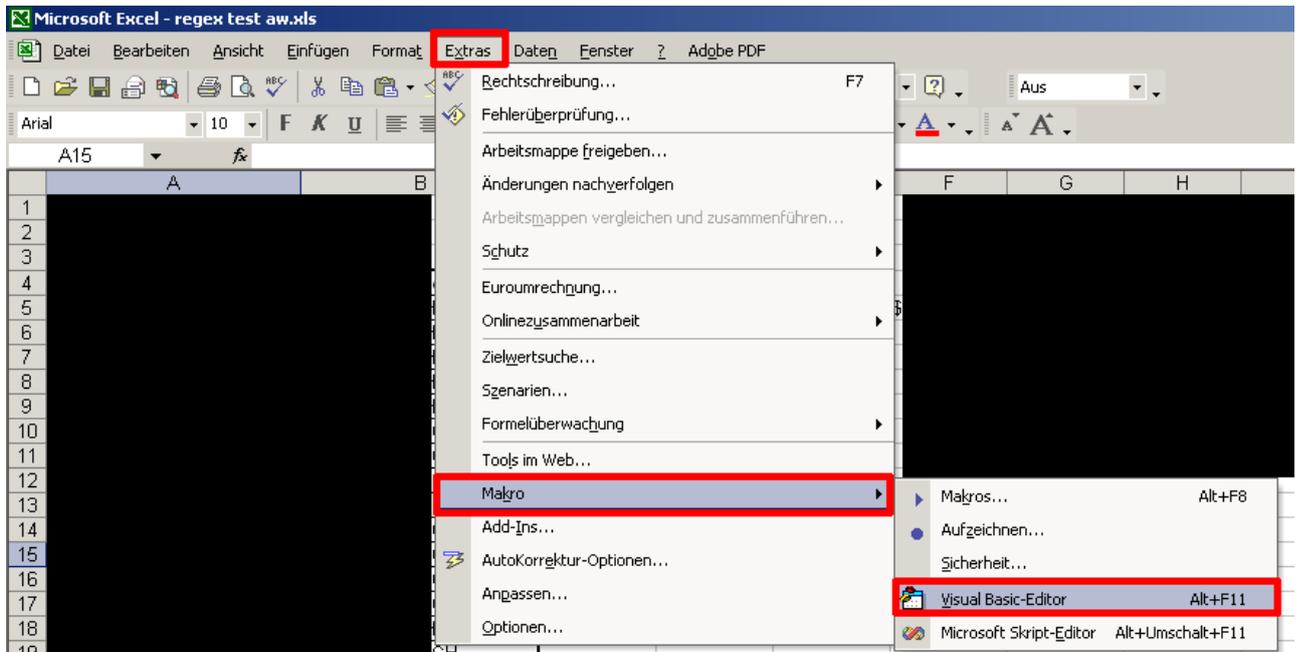
## 1. Vorbereiten der betroffenen Excel Datei

An dieser Stelle fügen wir in unser Excel Dokument einen VBA Codeblock ein welcher uns neue "benutzerdefinierte" Excel Funktionen zur Verfügung stellt.

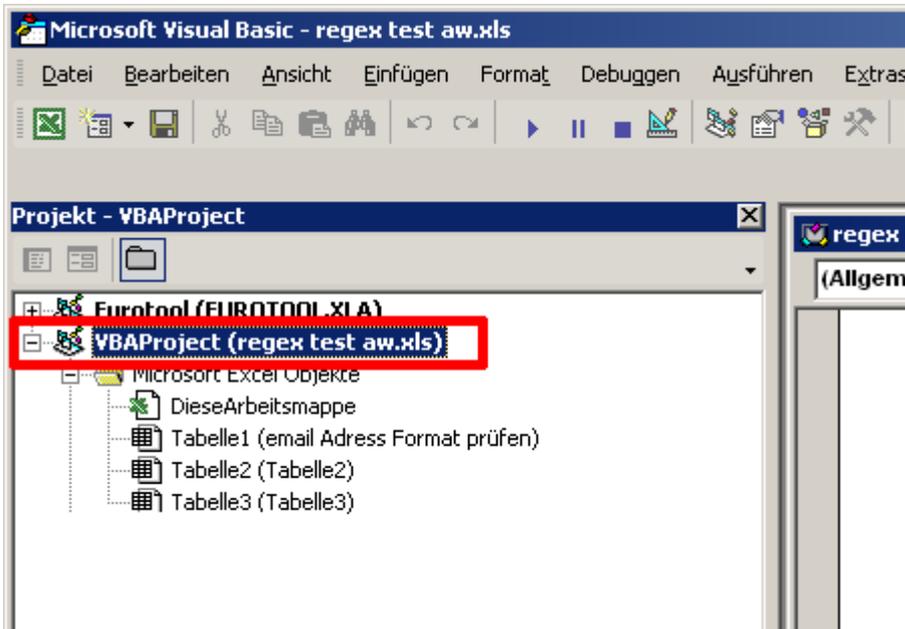
1. Öffnen Sie Ihre Excel Datei



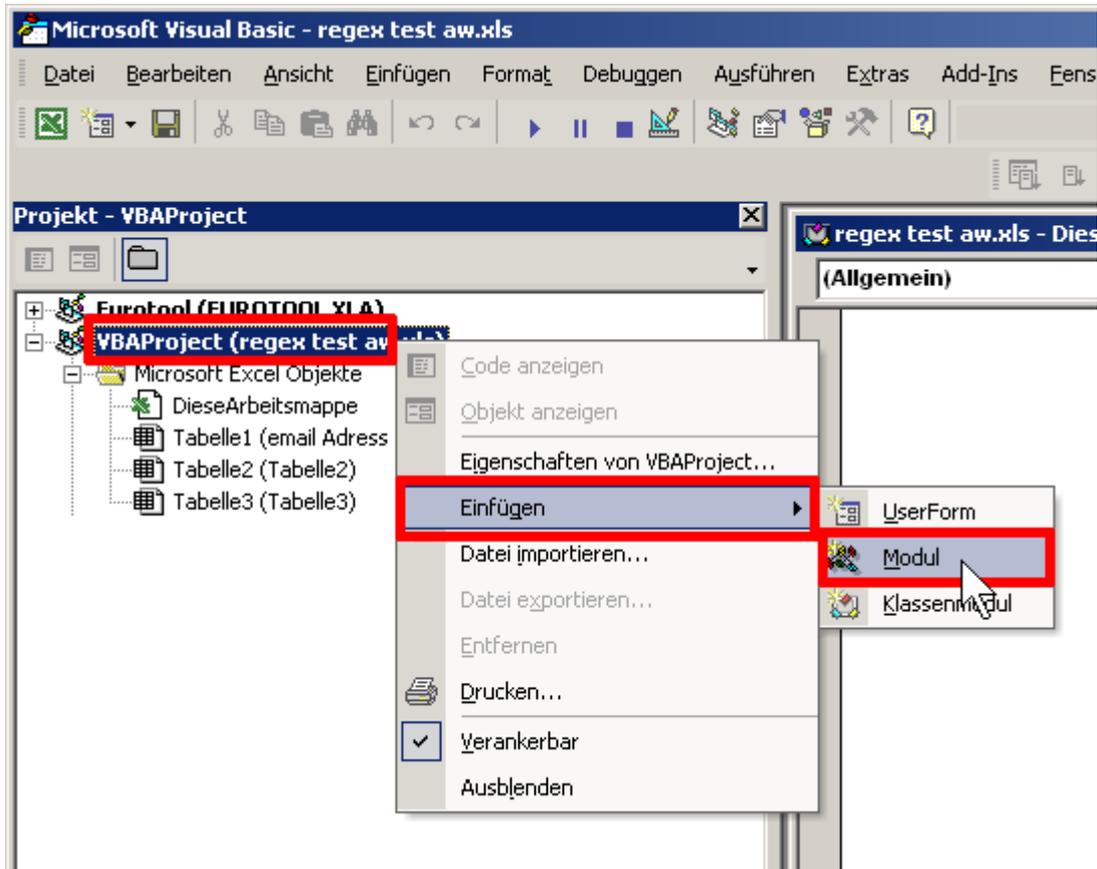
2. Öffnen Sie im Menü EXTRAS / MAKRO den Punkt "Visual Basic Editor"



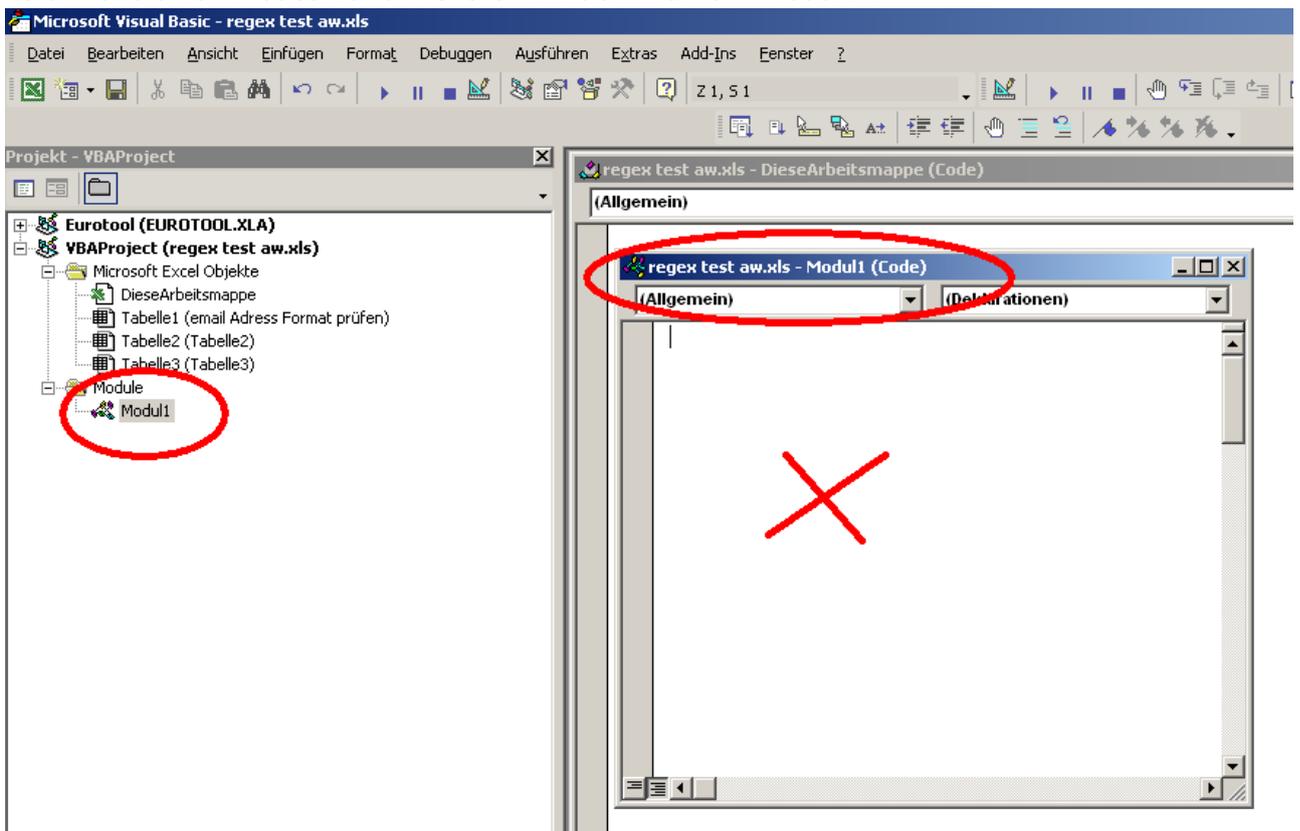
3. Suchen Sie nun in der LINKEN BILDSCHIRMHÄLFTE den Namen Ihres Excel Dokumentes



4. Erzeugen Sie an dieser Stelle ein neues "Code Modul" (Modul1) in dem Sie in der LINKEN SPALTE auf dem Namen Ihres Excel Dokumentes die RECHTE Maustaste klicken und aus dem Kontextmenü EINFÜGEN / MODUL auswählen



5. Es öffnet sich ein neues Editor-Fenster mit dem Namen "Modul1".



6. Geben Sie bei dem ROTEN X den nachfolgend gezeigten Codeblock ein:

```
Option Explicit
#Const LateBind = True
```

```
Public Function RegExMatch(FindIn, FindWhat As String, Optional IgnoreCase As Boolean = True) As Boolean
    #If Not LateBind Then
        Dim RE As RegExp, allMatches As MatchCollection, aMatch As Match
    Set RE = New RegExp
    #Else
        Dim RE As Object, allMatches As Object, aMatch As Object
        Set RE = CreateObject("vbscript.regexp")
    #End If
    RE.Pattern = FindWhat
    RE.IgnoreCase = IgnoreCase
    RE.Global = True
    RegExMatch = RE.Test(FindIn)
End Function
```

```
Function RegExSubstitute(ReplaceIn, ReplaceWhat As String, ReplaceWith As String)
    #If Not LateBind Then
        Dim RE As RegExp
        Set RE = New RegExp
    #Else
        Dim RE As Object
        Set RE = CreateObject("vbscript.regexp")
    #End If
    RE.Pattern = ReplaceWhat
    RE.Global = True
    RegExSubstitute = RE.Replace(ReplaceIn, ReplaceWith)
End Function
```

```
Function RegExFind(FindIn, FindWhat As String, _
    Optional IgnoreCase As Boolean = False)
    Dim i As Long
    #If Not LateBind Then
        Dim RE As RegExp, allMatches As MatchCollection, aMatch As Match
    Set RE = New RegExp
    #Else
        Dim RE As Object, allMatches As Object, aMatch As Object
        Set RE = CreateObject("vbscript.regexp")
    #End If
    RE.Pattern = FindWhat
    RE.IgnoreCase = IgnoreCase
    RE.Global = True
    Set allMatches = RE.Execute(FindIn)
    ReDim rslt(0 To allMatches.Count - 1)
    For i = 0 To allMatches.Count - 1
        rslt(i) = allMatches(i).Value
    Next i
    RegExFind = rslt
End Function
```

## End Function

7. Klicken Sie anschließend auf das SPEICHERN Symbol (Diskette) und schließen Sie den Visual-Basic Editor. Sie befinden sich nun wieder in Ihrem Excel Dokument. Die Vorbereitungen sind damit abgeschlossen und Sie können nun die neuen "Benutzerdefinierten Excel Funktionen" nutzen.

## 2. Anwenden der neuen Excel-Funktionen:

In unserem Beispiel haben wir ja bereits eine Excel Tabelle mit einer Spalte voller eMail Adressen. Nicht jede eMail Adresse wurde korrekt eingegeben und enthält womöglich illegale Zeichen oder hat ein illegales Format. Die uns nun zur Verfügung stehende neue Excel Funktion "RegExMatch(Zelle;RegExString)" wird uns hierbei helfen. Die Funktion vergleicht die "Zelle" mit dem Regulären Ausdruck "RegExString" und gibt bei einem Treffer ein WAHR, andernfalls ein FALSCH zurück.

Zur Überprüfung von gängigen eMail Adressen hat sich folgender Reguläre Ausdruck bewährt:

```
^[a-z0-9_\. \- ]+\@\(([a-z0-9\ - ]+\.\. )+([a-z ]+)\)$
```

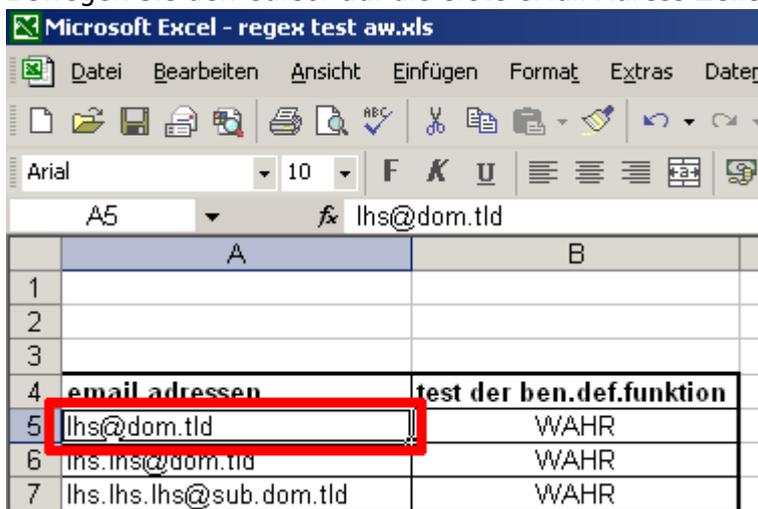
Um also eine eMail Adresse zu überprüfen könnten wir z.B. in einer Nebenspalte folgende Formel hinterlegen:

```
=RegExMatch(A5;"^[a-z0-9_\. \- ]+\@\(([a-z0-9\ - ]+\.\. )+([a-z ]+)\)$")
```

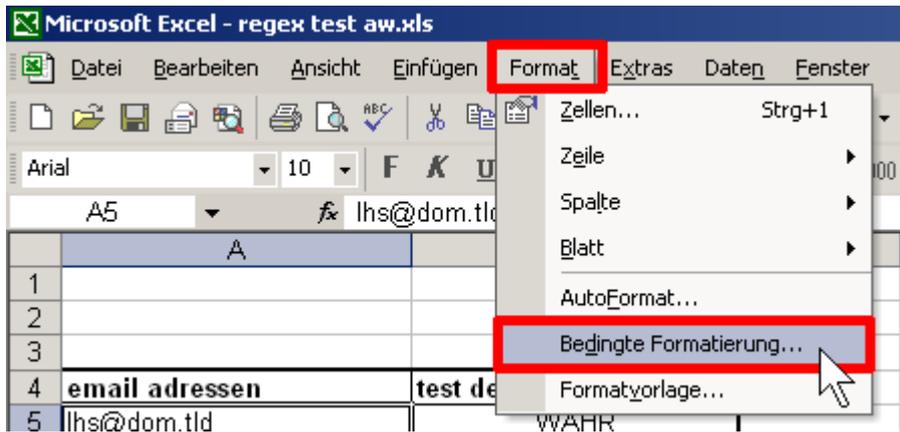
Entspricht die Adresse in Zelle "A5" einem gültigen Format erhalten wir an der aktuellen Stelle eine "WAHR" Ausgabe. Dies können wir also leicht nutzen um die Spalte der eMail Adressen mit Hilfe von "bedingter Formatierung" einzufärben wann immer eine eMail Adresse korrekt eingegeben wurde oder auf Wunsch auch umgekehrt, die Zelle soll sich verfärben wenn die Adresse nicht korrekt ist.

Dazu gehen Sie wie folgt vor:

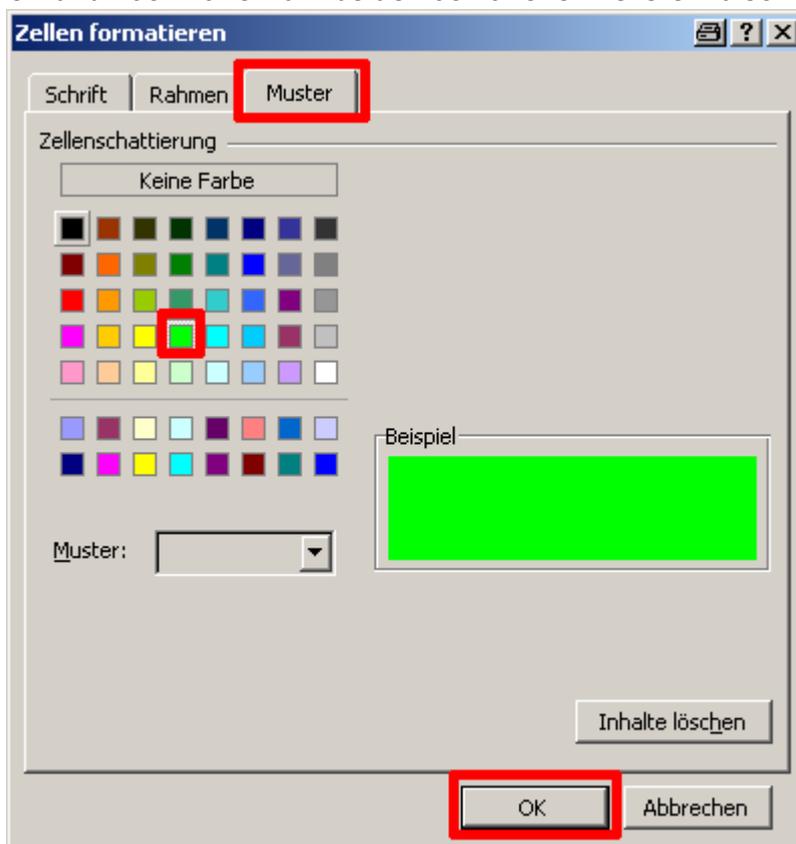
1. Bewegen Sie den Cursor auf die erste eMail-Adresse-Zelle Ihrer Liste



2. Wählen Sie aus dem Menü FORMAT den Punkt "BEDINGTE FORMATIERUNG...". Es öffnet sich ein neues Fenster



3. Stellen Sie das Fenster wie gezeigt ein, geben Sie die oben gezeigte Formel ein und klicken Sie anschließend auf den Button "FORMAT" um die Darstellung (Farben) der Zelle zu ändern.
4. Klicken Sie auf die Registerkarte MUSTER und wählen Sie für die Zelle eine gewünschte Farbe aus, für den Fall das die eingegebene eMail Adresse KORREKT ist. Klicken Sie anschließend auf OK und nochmal OK um beide noch offenen Fenster zu schließen.



5. Sie befinden sich nun wieder in Ihrem Excel Dokument und müssten jetzt sehen dass sich die Farbe der ersten eMail-Adress-Zelle geändert haben müsste, sofern dort ein gültige Adresse hinterlegt wurde. Wenn Sie die eMail Adresse an dieser Stelle nun in eine ungültige ändern, so wird sich die Zelle wieder in Ihre eigentliche "Ursprungsfarbe" verändern.

The screenshot shows an Excel spreadsheet with two columns: 'email adressen' and 'test der ben.def.funktion'. The data is as follows:

	A	B
1		
2		
3		
4	<b>email adressen</b>	<b>test der ben.def.funktion</b>
5	lhs@dom.tld	WAHR
6	lhs.lhs@dom.tld	WAHR
7	lhs.lhs.lhs@sub.dom.tld	WAHR
8	lhs.lhs.lhs@sub.dom.tld	WAHR
9	lhs.lhs.lhs@sub-sub.dom.tld	WAHR

- 6. Wenn Sie nun möchten dass die Zellen der Spalte "EMail Adressen" von ROT (illegale Eingabe) nach GRÜN (korrekte Eingabe) umspringen sollen, müssen Sie nur noch ALLE ZELLEN dieser Spalte grundsätzlich ROT einfärben (Hintergrundfarbe). Bei korrekter Eingabe einer EMail Adresse wird sich eine solche Zelle auch dann noch weiterhin GRÜN verfärben.

	A	B	C	D	E	F
1						
2						
3						
4	<b>email adressen</b>	<b>test der ben.def.funktion</b>				
5	lhs@dom.tld	WAHR				
6	lhs.lhs@dom.tld	WAHR				
7	lhs.lhs.lhs@sub.dom.tld	WAHR				
8	lhs_lhs.lhs@sub.dom.tld	WAHR				
9	lhs-lhs.lhs@sub-sub.dom.tld	WAHR				
10	lhs lhs@sub-sub.dom.tld	FALSCH				
11	lhs.lhs@sub_sub.dom.tld	FALSCH				
12	lhsâ@sub.dom.tld	FALSCH				
13	lhs@ösub.dom.tld	FALSCH				
14		FALSCH				
15		FALSCH				
16		FALSCH				
17		FALSCH				
18	a@b.de	WAHR				
19		FALSCH				
20		FALSCH				
21		FALSCH				
22		FALSCH				
23		FALSCH				
24		FALSCH				
25		FALSCH				
26		FALSCH				
27		FALSCH				
28		FALSCH				
29		FALSCH				
30		FALSCH				
31		FALSCH				
32		FALSCH				
33		FALSCH				
34		FALSCH				
35		FALSCH				
36		FALSCH				
37						
38						

7. Wie sie sehen stimmt im Augenblick in unserem Beispiel nur die "Bedingte Formatierung" der ERSTEN eMail Zelle. Alle anderen Zellen sind selbst dann noch ROT wenn eine korrekte eMail Adresse eingegeben wurde. Dies liegt daran dass wir bislang nur die erste Zelle mit der "Bedingten Formatierung" behandelt haben. Um dies auch auf die anderen eMail Zellen zu übertragen verwenden Sie das "Pinsel Symbol". Markieren Sie dazu zunächst die erste korrekt funktionierende eMail Zelle, klicken dann auf das Pinsel Symbol und anschließend markieren Sie alle weiteren eMail Zellen. Sodann wird auch die Bedingte Formatierung übernommen.

	A	Bearbeitungsleiste
1		
2		
3		
4	<b>email adressen</b>	<b>test der ben.def.funktion</b>
5	lhs@dom.tld	WAHR
6	lhs.lhs@dom.tld	WAHR
7	lhs.lhs.lhs@sub.dom.tld	WAHR
8	lhs.lhs.lhs@sub.dom.tld	WAHR
9	lhs-lhs.lhs@sub-sub.dom.tld	WAHR
10	lhs lhs@sub-sub.dom.tld	FALSCH
11	lhs.lhs@sub sub.dom.tld	FALSCH
12	lhsä@sub.dom.tld	FALSCH
13	lhs@ösub.dom.tld	FALSCH
14		FALSCH
15		FALSCH
16		FALSCH
17		FALSCH
18	a@b.de	WAHR
19		FALSCH
20		FALSCH
21		FALSCH
22		FALSCH
23		FALSCH
24		FALSCH
25		FALSCH
26		FALSCH
27		FALSCH
28		FALSCH
29		FALSCH
30		FALSCH
31		FALSCH
32		FALSCH
33		FALSCH
34		FALSCH
35		FALSCH
36		FALSCH

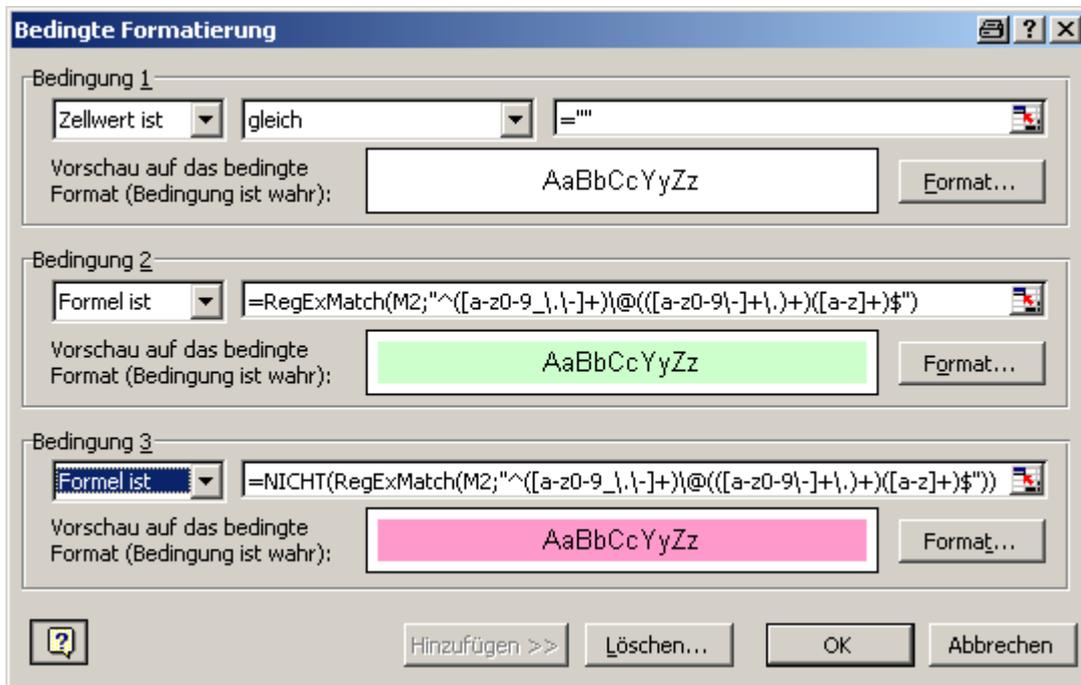
**Ein Tip:**

Wenn Sie Ihrer Formel zur bedingten Formatierung eine NICHT() Funktion hinzufügen, können Sie die Funktionsweise auch umkehren, so dass ein Farbwechsel nur bei falschen Adressen vorgenommen wird. Die Formel könnte dann wie folgt aussehen:

```
=NICHT(RegExMatch(A18;"^([a-z0-9_\. -]+)\@(([a-z0-9\ -]+\.)+)([a-z]+)$"))
```

## Noch ein Tip:

Stellt man sich die Befingte Formatierung wie folgt ein, so erhält man einen noch bequemeren Effekt. Hat man in die Zelle nichts eingegeben so bleibt die Zelle Weiß. Hat man eine gültige Adresse eingegeben so färbt sie sich Grün. Falsche Eingaben färben die Adresse nun rot. Somit fallen Fehler noch leichter auf.



— Axel Werner 2010-09-20 19:42

## Quellen und Files:

- <http://www.dailydoseofexcel.com/archives/2005/08/13/pattern-matching/>
- <http://www.tmehta.com/regex/>
- Microsoft Demo Datei: [regex-sept-1-2004.xls](#)
- Meine Demo Datei [regex-test-aw.xls](#)

— Axel Werner 2010-09-17 21:11

From: <https://awerner.myhome-server.de/> - Axel Werner's OPEN SOURCE Knowledge Base

Permanent link: <https://awerner.myhome-server.de/doku.php?id=it-artikel:windows:excel-email-adressen-mit-hilfe-von-regulaeren-ausdruecken-auf-gueltigkeit-ueberpruefen>

Last update: 2022-08-31 12:30

