



Excel VBA Funktionen

4.0 Skala Auswertung

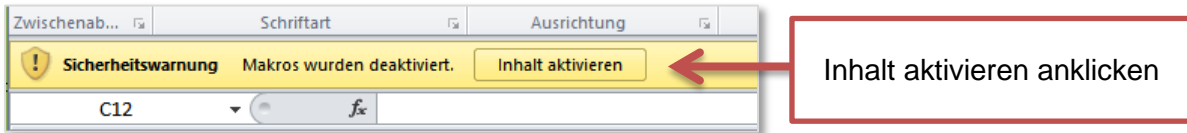
Thomas Maier, www.css4.at, 2019 CC

Inhaltsverzeichnis

Testumgebung.xlsm.....	3
Vorgaben laut dem Leitfaden der Validierung.....	4
Speicherort der Module.....	5
Prinzipien zur Programmierung	6
Variablendeklaration	6
Gemeinsamkeit aller Funktionen.....	7
=prüfe1V()	8
=prüfe2V()	9
=prüfe3V()	10
=prüfe4V()	11
=prüfe3G().....	12
=prüfe4G().....	13
=Beurteilung().....	14
=einser().....	15
=zweier().....	16
=dreier().....	17
=vierer().....	18
=Schulnote()	19
=VollerText()	20

Testumgebung.xlsm

Das Excel-Sheet beinhaltet VBA Code in einem Modul. Falls Makros deaktiviert sind, muss der Inhalt nach dem Öffnen aktiviert werden.



Es öffnet sich das Excel Sheet. In der Testumgebung müssen nur die Anzahl der Skala Ergebnisse in den Zellen C4 bis C12 eingegeben werden.

	A	B	C	D	E	F	G	H	I	J
1	Testumgebung 4.0 Skala Auswertung									
2										
3		Skala	Anzahl							
4	Nr. 1	4.0			Note	4V	Genügend in Vertiefter Allgemeinbildung			
5	Nr. 2	3.5	2		1V	Nur 0 % sind über 4.0 (50 % sind notwendig)				
6	Nr. 3	3.0	1		2V	Die Werte über 3.0 sind nur 50 % (75 % sind notwendig)				
7	Nr. 4	2.5	1		3V	Es gibt mehr als eine 1.0 bzw. 1.5 Bewertung				
8	Nr. 5	2.0			4V	OK				
9	Nr. 6	1.5	2		3G	OK				
10	Nr. 7	1.0			4G	OK				
11	Nr. 8	0.5			5G	OK				
12	Nr. 9	0.0								
13					Schulnote	3	Befriedigend			
14					Sehr Gut	Nur 0 % sind 4.0 (Mindestens 50 %)				
15					Gut	Es gibt Ergebnisse unter 2.0				
16					Befriedigend	OK				
17					Genügend	OK				
18					Nicht Genügend	OK				
19										

Nach der Eingabe der Ergebnisse (Anzahl der Skala-Werte) ermittelt die Funktion in der Zelle F1 die Benotung nach vertiefter und grundlegender Allgemeinbildung.

Die Funktion in der Zelle F13 ermittelt die Note nach dem traditionellen Schulnotensystem.

Zusätzlich wird jede Note geprüft. Wenn ein Kriterium nicht erfüllt wird, gibt die Funktion eine Begründung in roter Schrift aus. Sonst wird die Note mit einem grünen OK bestätigt.

Alle Funktionen greifen auf den Zellbereich der Ergebnisanzahl zu.

z. B: =Beurteilung(C4:C12) oder =prüfe3G(C4:C12)

Die Funktionen können also auf jeder beliebigen Stelle im Dokument aufgerufen werden, müssen aber auf den Zellbereich zugreifen.

Der Zellbereich muss alle 9 Werte umspannen.

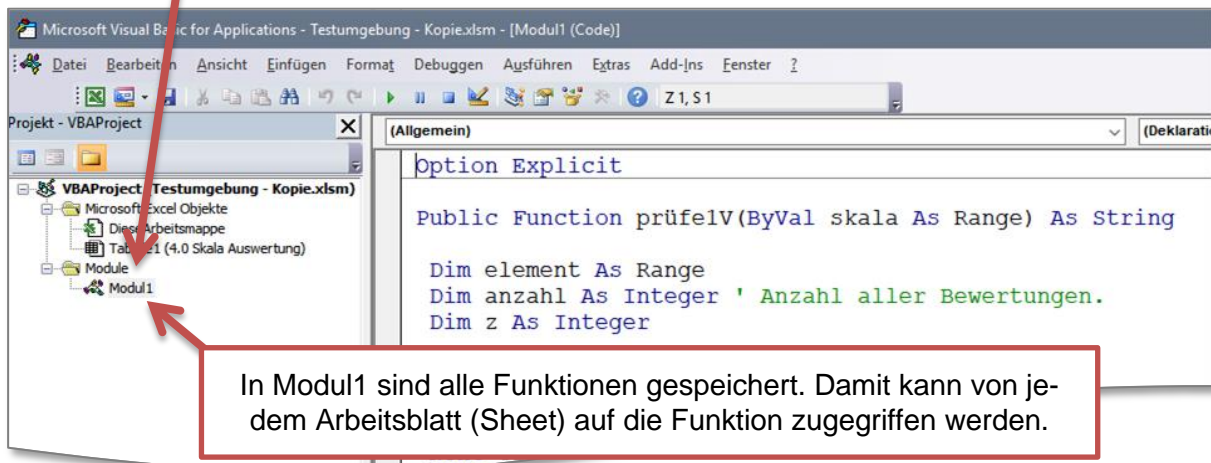
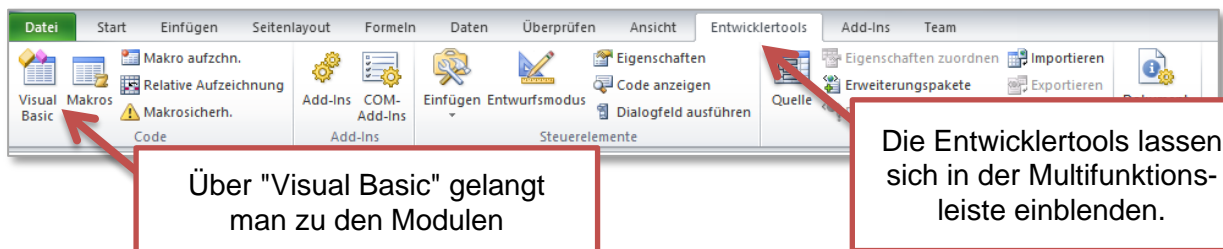
Die Reihenfolge geht immer von 4.0 bis 0.0 – also 4.0 ist der erste Wert, 3.5 der zweite usw.

Vorgaben laut dem Leitfaden der Validierung

1V	Mindestens 50% der Ergebnisse sind 4.0, die restlichen Ergebnisse sind 3.0 oder 3.5
2V	75% der Ergebnisse sind 3.0 oder höher, die restlichen Ergebnisse sind nicht weniger als 2.0
3V	Mindestens 40% der Ergebnisse sind 3.0 oder höher. Von den restlichen Ergebnissen ist maximal eines 1.0, die übrigen sind nicht weniger als 2.0.
4V	Mindestens 50% der Ergebnisse sind 2.0 oder höher. Von den restlichen Ergebnissen ist maximal eines 0.0, die übrigen sind nicht weniger als 1.0.
3G	Mindestens 25% der Ergebnisse sind 2.0 oder höher. Von den restlichen Ergebnissen ist maximal eines 0.0, die übrigen sind nicht weniger als 1.0.
4G	Mindestens 75% der Ergebnisse sind 1.0 oder höher. Von den restlichen Ergebnissen ist maximal eines 0.0, die übrigen sind 0.5.

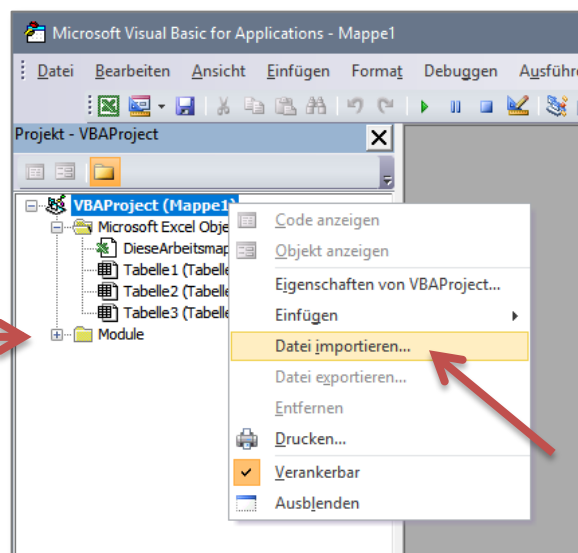
Mindestens 50 % der Ergebnisse sind 4.0, die restlichen Ergebnisse sind 3.0.	Sehr Gut
Mindestens 50% der Ergebnisse sind 3.0 oder höher, die restlichen Ergebnisse sind nicht weniger als 2.0	Gut
Mindestens 50% der Ergebnisse sind 2.0 oder höher, und die restlichen sind nicht weniger als 1.0.	Befriedigend
Maximal ein Ergebnis darf 0.0 sein, die restlichen Ergebnisse sind mindestens 1.0 oder höher.	Genügend

Speicherort der Module



Mit einem Rechtsklick auf **VBAProjekt** kann die **SkalaFunktionen.bas** importiert werden. Diese beinhaltet alle Funktionen. Nach dem Import stehen dann die Funktionen in der Excel-Mappe zur Verfügung.

Nach dem Import findet man hier die Module. Es kann sein, dass ein anderer Modul-Name als Modul1 vergeben wurde.



Prinzipien zur Programmierung

Drop-out: Die Funktionen prüfen alle Kriterien. Sobald ein Kriterium nicht erfüllt wird, gibt die Funktion eine Begründung zurück und verlässt die Funktion ohne eine weitere Prüfung der anderen Kriterien!

Variablendeklaration

Variable	Typ	Beschreibung
skala	<i>Range/Objekt</i>	Übernimmt den Zellbereich. Wirkt wie ein Array. ⇒ skala(1) = Wert von 4.0 ⇒ skala(2) = Wert von 3.5 ⇒ skala(3) = Wert von 3.0 ⇒ skala(4) = Wert von 2.5 ⇒ skala(5) = Wert von 2.0 ⇒ skala(6) = Wert von 1.5 ⇒ skala(7) = Wert von 1.0 ⇒ skala(8) = Wert von 0.5 ⇒ skala(9) = Wert von 0.0
element	<i>Range/Objekt</i>	Wird für die For-Each Schleife benötigt
anzahl	<i>Integer</i>	Summiert die Gesamtanzahl aller Bewertungen aus der Skala 4.0 bis 0.0
z	<i>Integer</i>	Zähler
krit1 krit2 krit3	<i>Single/Integer</i>	Kriterium, wird für Berechnungen verwendet um ein Kriterium zu validieren. z. B. zur Ermittlung von Prozentwerten oder um die Anzahl von bestimmten Skala-Ereignissen zu bestimmen.
n4g, n3g, n4v, n3v, n2v, n1v	<i>String</i>	Für die Funktionen Beurteilung() und Schulnote() Übernimmt den Rückgabewert der Überprüfungsfunktionen und prüft ob die Rückgabe ein "OK" ist.
ausgabe	<i>String</i>	Textausgabe für die Funktionen Beurteilung() und Schulnote()

Gemeinsamkeit aller Funktionen

1:	Public Function prüfelV (ByVal skala As Range) As String
2:	Dim element As Range
3:	Dim anzahl As Integer ' Anzahl aller Bewertungen.
4:	Dim z As Integer
5:	For Each element In skala
6:	anzahl = anzahl + element
7:	z = z + 1
8:	Next element
9:	If z <> 9 Then
10:	prüfelV = "Bitte alle 9 Werte markieren (von 4.0 bis 0.0)"
11:	Exit Function
12:	End If
13:	...

Erläuterung

1. Jede Funktion wird als Public dem gesamten Excel-Sheet bereitgestellt. Der Funktionsname (hier **prüfelV**) ist selbstverständlich in jeder Funktion anders. Der Zellbereich wird als Wert an das Range-Objekt `skala` übergeben. `skala(1)` entspricht damit dann den Wert von 4.0. Rückgabe ist ein String.
2. Ein Range-Element wird dimensioniert
3. Die Anzahl (Summe der Ergebnisse) wird dimensioniert. Diese wird später für die Prozentberechnungen verwendet.
4. Ein Zähler wird dimensioniert
5. Alle Werte aus dem Zellbereich (`skala`) werden mit einer For-Each Schleife "durchwandert".
6. Die Summe aller Ergebnisse wird ermittelt.
7. Der Zähler wird erhöht.
8. Nächstes Element in der For-Each Schleife.
9. Prüfung ob 9 Elemente gezählt wurden (die 9 Werte aus dem Zellbereich!)
10. Wurden nicht 9 Elemente ausgewählt und gezählt, gibt die Funktion die Meldung: "Bitte alle 9 Werte markieren (von 4.0 bis 0.0)" in der Zelle zurück.
11. Die Funktion wird verlassen.
12. Ende der IF-Anweisung (Prüfung auf 9 Elemente).
13. weiterer Code!

=prüfe1V()

Funktion: Prüft die Kriterien der Note 1V und gibt eine Begründung zurück, falls ein Kriterium nicht erfüllt ist. Sind alle Kriterien erfüllt, gibt die Funktion ein "OK" zurück.

Beispiel: =prüfe1V(C4:C12)

Code:

```
Public Function prüfe1V(ByVal skala As Range) As String

    Dim element As Range
    Dim anzahl As Integer ' Anzahl aller Bewertungen.
    Dim z As Integer

    For Each element In skala
        anzahl = anzahl + element
        z = z + 1
    Next element

    If z <> 9 Then
        prüfe1V = "Bitte alle 9 Werte markieren (von 4.0 bis 0.0)"
        Exit Function
    End If

    Dim krit1 As Single ' Ermittelt ob 50% der Ergebnisse 4.0 sind.
    krit1 = skala(1) / anzahl
    If krit1 < 0.5 Then
        prüfe1V = "Nur " & CInt(krit1 * 100) & _
            " % sind über 4.0 (50 % sind notwendig)"
        Exit Function
    End If

    Dim krit2 As Integer ' Ermittelt ob Ergebnisse unter 3.0 vorkommen.
    krit2 = skala(4) + skala(5) + skala(6) + skala(7) + skala(8) + skala(9)
    If krit2 > 0 Then
        prüfe1V = "Es gibt Bewertungen unter 3.0"
        Exit Function
    End If

    prüfe1V = "OK"

End Function
```

=prüfe2V()

Funktion: Prüft die Kriterien der Note 2V und gibt eine Begründung zurück, falls ein Kriterium nicht erfüllt ist. Sind alle Kriterien erfüllt, gibt die Funktion ein "OK" zurück.

Beispiel: =prüfe2V(C4:C12)

Code:

```
Public Function prüfe2V(ByVal skala As Range) As String

    Dim element As Range
    Dim anzahl As Integer ' Anzahl aller Bewertungen.
    Dim z As Integer

    For Each element In skala
        anzahl = anzahl + element
        z = z + 1
    Next element

    If z <> 9 Then
        prüfe2V = "Bitte alle 9 Werte markieren (von 4.0 bis 0.0)"
        Exit Function
    End If

    Dim krit1 As Single ' Ermittelt ob 75% der Ergebnisse 4.0 sind.
    krit1 = (skala(1) + skala(2) + skala(3)) / anzahl
    If krit1 < 0.75 Then
        prüfe2V = "Die Werte über 3.0 sind nur " & CInt(krit1 * 100) & _
            " % (75 % sind notwendig)"
        Exit Function
    End If

    Dim krit2 As Integer ' Ermittelt ob es Werte unter 2.0 gibt.
    krit2 = skala(6) + skala(7) + skala(8) + skala(9)
    If krit2 > 0 Then
        prüfe2V = "Es gibt Werte unter 2.0"
        Exit Function
    End If

    prüfe2V = "OK"

End Function
```

=prüfe3V()

Prüft die Kriterien der Note 3V und gibt eine Begründung zurück, falls ein Kriterium nicht erfüllt ist. Sind alle Kriterien erfüllt, gibt die Funktion ein "OK" zurück.

Beispiel: =prüfe3V(C4:C12)

Code:

```
Public Function prüfe3V(ByVal skala As Range) As String

    Dim element As Range
    Dim anzahl As Integer ' Anzahl aller Bewertungen.
    Dim z As Integer

    For Each element In skala
        anzahl = anzahl + element
        z = z + 1
    Next element

    If z <> 9 Then
        prüfe3V = "Bitte alle 9 Werte markieren (von 4.0 bis 0.0)"
        Exit Function
    End If

    Dim krit1 As Single ' Ermittelt ob 40% der Ergebnisse 3.0 sind.
    krit1 = (skala(1) + skala(2) + skala(3)) / anzahl
    If krit1 < 0.4 Then
        prüfe3V = "Die Werte über 3.0 sind nur " & CInt(krit1 * 100) & _
            " % (40 % sind notwendig)"
        Exit Function
    End If

    Dim krit2 As Integer ' Ermittelt ob es mehr als ein 1.0
                        ' oder 1.5 Werte gibt.
    krit2 = skala(6) + skala(7)
    If krit2 > 1 Then
        prüfe3V = "Es gibt mehr als eine 1.0 bzw. 1.5 Bewertung"
        Exit Function
    End If

    Dim krit3 As Integer ' Ermittelt die Anzahl von 0.5 und 0.0
    krit3 = skala(8) + skala(9)
    If krit3 > 0 Then
        prüfe3V = "Es gibt 0.5 bzw. 0.0 Bewertungen"
        Exit Function
    End If

    prüfe3V = "OK"

End Function
```

=prüfe4V()

Prüft die Kriterien der Note 4V und gibt eine Begründung zurück, falls ein Kriterium nicht erfüllt ist. Sind alle Kriterien erfüllt, gibt die Funktion ein "OK" zurück.

Beispiel: =prüfe4V(C4:C12)

Code:

```
Public Function prüfe4V(ByVal skala As Range) As String

    Dim element As Range
    Dim anzahl As Integer ' Anzahl aller Bewertungen.
    Dim z As Integer

    For Each element In skala
        anzahl = anzahl + element
        z = z + 1
    Next element

    If z <> 9 Then
        prüfe4V = "Bitte alle 9 Werte markieren (von 4.0 bis 0.0)"
        Exit Function
    End If

    Dim krit1 As Single ' Ermittelt ob 50% der Ergebnisse 2.0 sind.
    krit1 = (skala(1) + skala(2) + skala(3) + skala(4) + skala(5)) / anzahl
    If krit1 < 0.5 Then
        prüfe4V = "Die Ergebnisse über 2.0 liegen bei " & _
            CInt(krit1 * 100) & " % (50 % Minimum)"
        Exit Function
    End If

    Dim krit2 As Integer ' Ermittelt die Anzahl der 0.0.
    krit2 = skala(9)
    If krit2 > 1 Then
        prüfe4V = "Es gibt mehr als eine 0.0 Bewertung"
        Exit Function
    End If

    Dim krit3 As Integer ' Ermittelt ob es 0.5 und 0.0 Bewertungen gibt.
    krit3 = skala(8) + skala(9)
    If krit3 > 1 Then
        prüfe4V = "Es gibt Ergebnisse unter 1.0 (ein Ergebnis wäre erlaubt)"
        Exit Function
    End If

    prüfe4V = "OK"
End Function
```

=prüfe3G()

Prüft die Kriterien der Note 3G und gibt eine Begründung zurück, falls ein Kriterium nicht erfüllt ist. Sind alle Kriterien erfüllt, gibt die Funktion ein "OK" zurück.

Beispiel: =prüfe3G(C4:C12)

Code:

```
Public Function prüfe3G(ByVal skala As Range) As String

    Dim element As Range
    Dim anzahl As Integer ' Anzahl aller Bewertungen.
    Dim z As Integer

    For Each element In skala
        anzahl = anzahl + element
        z = z + 1
    Next element

    If z <> 9 Then
        prüfe3G = "Bitte alle 9 Werte markieren (von 4.0 bis 0.0)"
        Exit Function
    End If

    Dim krit1 As Single ' Ermittelt ob 25% der Ergebnisse 2.0 sind.
    krit1 = (skala(1) + skala(2) + skala(3) + skala(4) + skala(5)) / anzahl
    If krit1 < 0.25 Then
        prüfe3G = "Die Ergebnisse über 2.0 liegen bei " & _
            CInt(krit1 * 100) & " % (25 % Minimum)"
        Exit Function
    End If

    Dim krit2 As Integer ' Ermittelt die Anzahl der 0.0.
    krit2 = skala(9)
    If krit2 > 1 Then
        prüfe3G = "Es gibt mehr als eine 0.0 Bewertung"
        Exit Function
    End If

    Dim krit3 As Integer ' Ermittelt ob es 0.5 und 0.0 Bewertungen gibt.
    krit3 = skala(8) + skala(9)
    If krit3 > 1 Then
        prüfe3G = "Es gibt Ergebnisse unter 1.0 (ein Ergebnis wäre erlaubt)"
        Exit Function
    End If

    prüfe3G = "OK"

End Function
```

=prüfe4G()

Prüft die Kriterien der Note 4G und gibt eine Begründung zurück, falls ein Kriterium nicht erfüllt ist. Sind alle Kriterien erfüllt, gibt die Funktion ein "OK" zurück.

Beispiel: =prüfe4G(C4:C12)

Code:

```
Public Function prüfe4G(ByVal skala As Range) As String

    Dim element As Range
    Dim anzahl As Integer ' Anzahl aller Bewertungen.
    Dim z As Integer

    For Each element In skala
        anzahl = anzahl + element
        z = z + 1
    Next element

    If z <> 9 Then
        prüfe4G = "Bitte alle 9 Werte markieren (von 4.0 bis 0.0)"
        Exit Function
    End If

    Dim krit1 As Single ' Ermittelt ob 75% der Ergebnisse 1.0 sind.
    krit1 = (skala(1) + skala(2) + skala(3) + skala(4) + skala(5) + _
            skala(6) + skala(7)) / anzahl
    If krit1 < 0.75 Then
        prüfe4G = "Die Ergebnisse über 1.0 liegen bei " & _
            CInt(krit1 * 100) & " % (75 % Minimum)"
        Exit Function
    End If

    Dim krit2 As Integer ' Ermittelt die Anzahl der 0.0.
    krit2 = skala(9)
    If krit2 > 1 Then
        prüfe4G = "Es gibt mehr als eine 0.0 Bewertung"
        Exit Function
    End If

    prüfe4G = "OK"

End Function
```

=Beurteilung()

Die Funktion wird mit dem Zellbereich der Skala-Werte gestartet und prüft jede Note sequenziell von 5G bis 1V. Die höchste Note wird dann ausgegeben.

Beispiel: =Beurteilung(C4:C12)

Code:

```
Public Function Beurteilung(ByVal skala As Range) As String
    Dim ausgabe As String
    ausgabe = "5G"

    Dim n4g As String
    n4g = prüfe4G(skala)
    If n4g = "OK" Then
        ausgabe = "4G"
    End If

    Dim n3g As String
    n3g = prüfe3G(skala)
    If n3g = "OK" Then
        ausgabe = "3G"
    End If

    Dim n4v As String
    n4v = prüfe4V(skala)
    If n4v = "OK" Then
        ausgabe = "4V"
    End If

    Dim n3v As String
    n3v = prüfe3V(skala)
    If n3v = "OK" Then
        ausgabe = "3V"
    End If

    Dim n2v As String
    n2v = prüfe2V(skala)
    If n2v = "OK" Then
        ausgabe = "2V"
    End If

    Dim n1v As String
    n1v = prüfe1V(skala)
    If n1v = "OK" Then
        ausgabe = "1V"
    End If

    Beurteilung = ausgabe

End Function
```

=einser()

Die Funktion überprüft über die Ergebnisse ob die Note Sehr Gut nach dem traditionellen Schulnotensystem vergeben werden kann. Werden alle Kriterien erfüllt, gibt die Funktion ein "OK" zurück, sonst folgt eine Begründung warum das Kriterium nicht erfüllt wurde.

Beispiel: =einser(C4:C12)

Code:

```
Public Function einser(ByVal skala As Range) As String

    Dim element As Range
    Dim anzahl As Integer ' Anzahl aller Bewertungen.
    Dim z As Integer

    For Each element In skala
        anzahl = anzahl + element
        z = z + 1
    Next element

    If z <> 9 Then
        einser = "Bitte alle 9 Werte markieren (von 4.0 bis 0.0)"
        Exit Function
    End If

    Dim krit1 As Single
    krit1 = skala(1) / anzahl
    If krit1 < 0.5 Then
        einser = "Nur " & CInt(krit1 * 100) & " % sind 4.0 (Mindestens 50 %)"
        Exit Function
    End If

    Dim krit2 As Integer
    krit2 = skala(4) + skala(5) + skala(6) + skala(7) + skala(8) + skala(9)
    If krit2 > 0 Then
        einser = "Es gibt Ergebnisse unter 3.0"
        Exit Function
    End If

    einser = "OK"

End Function
```

=zweier()

Die Funktion überprüft über die Ergebnisse ob die Note "Gut" nach dem traditionellen Schulnotensystem vergeben werden kann. Werden alle Kriterien erfüllt, gibt die Funktion ein "OK" zurück, sonst folgt eine Begründung warum das Kriterium nicht erfüllt wurde.

Beispiel: =zweier(C4:C12)

Code:

```
Public Function zweier(ByVal skala As Range) As String

    Dim element As Range
    Dim anzahl As Integer ' Anzahl aller Bewertungen.
    Dim z As Integer

    For Each element In skala
        anzahl = anzahl + element
        z = z + 1
    Next element

    If z <> 9 Then
        zweier = "Bitte alle 9 Werte markieren (von 4.0 bis 0.0)"
        Exit Function
    End If

    Dim krit1 As Single
    krit1 = (skala(1) + skala(2) + skala(3)) / anzahl
    If krit1 < 0.5 Then
        zweier = "Nur " & CInt(krit1 * 100) & _
            " % sind höher als 3.0 (Mindestens 50 %)"
        Exit Function
    End If

    Dim krit2 As Integer
    krit2 = skala(6) + skala(7) + skala(8) + skala(9)
    If krit2 > 0 Then
        zweier = "Es gibt Ergebnisse unter 2.0"
        Exit Function
    End If

    zweier = "OK"

End Function
```

=dreier()

Die Funktion überprüft über die Ergebnisse ob die Note "Befriedigend" nach dem traditionellen Schulnotensystem vergeben werden kann. Werden alle Kriterien erfüllt, gibt die Funktion ein "OK" zurück, sonst folgt eine Begründung warum das Kriterium nicht erfüllt wurde.

Beispiel: =dreier(C4:C12)

Code:

```
Public Function dreier(ByVal skala As Range) As String

    Dim element As Range
    Dim anzahl As Integer ' Anzahl aller Bewertungen.
    Dim z As Integer

    For Each element In skala
        anzahl = anzahl + element
        z = z + 1
    Next element

    If z <> 9 Then
        dreier = "Bitte alle 9 Werte markieren (von 4.0 bis 0.0)"
        Exit Function
    End If

    Dim krit1 As Single
    krit1 = (skala(1) + skala(2) + skala(3) + skala(4) + skala(5)) / anzahl
    If krit1 < 0.5 Then
        dreier = "Nur " & CInt(krit1 * 100) & _
            " % sind höher als 3.0 (Mindestens 50 %)"
        Exit Function
    End If

    Dim krit2 As Integer
    krit2 = skala(8) + skala(9)
    If krit2 > 0 Then
        dreier = "Es gibt Ergebnisse unter 1.0"
        Exit Function
    End If

    dreier = "OK"

End Function
```

=vierer()

Die Funktion überprüft über die Ergebnisse ob die Note "Genügend" nach dem traditionellen Schulnotensystem vergeben werden kann. Werden alle Kriterien erfüllt, gibt die Funktion ein "OK" zurück, sonst folgt eine Begründung warum das Kriterium nicht erfüllt wurde.

Beispiel: =vierer(C4:C12)

Code:

```
Public Function vierer(ByVal skala As Range) As String

    Dim element As Range
    Dim anzahl As Integer ' Anzahl aller Bewertungen.
    Dim z As Integer

    For Each element In skala
        anzahl = anzahl + element
        z = z + 1
    Next element

    If z <> 9 Then
        vierer = "Bitte alle 9 Werte markieren (von 4.0 bis 0.0)"
        Exit Function
    End If

    Dim krit1 As Integer
    krit1 = skala(9)
    If krit1 > 1 Then
        vierer = "Es gibt mehr als ein 0.0 Ergebnis"
        Exit Function
    End If

    Dim krit2 As Integer
    krit2 = skala(8) + skala(9)
    If krit2 > 1 Then
        vierer = "Es gibt mehr als ein Ergebnis unter 1.0"
        Exit Function
    End If

    vierer = "OK"

End Function
```

=Schulnote()

Die Funktion übernimmt den Zellbereich und gibt dann die vollständige Schulnote zurück.

Beispiel: =Schulnote(C4:C12)

Code:

```
Public Function Schulnote(ByVal skala As Range) As String
    Dim ausgabe As String
    ausgabe = "Nicht genügend"

    Dim n4v As String
    n4v = vierer(skala)
    If n4v = "OK" Then
        ausgabe = "Genügend"
    End If

    Dim n3v As String
    n3v = dreier(skala)
    If n3v = "OK" Then
        ausgabe = "Befriedigend"
    End If

    Dim n2v As String
    n2v = zweier(skala)
    If n2v = "OK" Then
        ausgabe = "Gut"
    End If

    Dim n1v As String
    n1v = einser(skala)
    If n1v = "OK" Then
        ausgabe = "Sehr gut"
    End If

    Schulnote = ausgabe
End Function
```

=VollerText()

Übersetzt die Kurzform einer Note (z. B. 3G) in eine lesbare Form mit vielen Wörtern (z. B. Befriedigend in Grundlegender Allgemeinbildung)

Beispiel: =VollerText(C4) oder =Vollertext("4V")

Code:

```
Public Function VollerText(ByVal dieNote As String) As String
    VollerText = "Unbekannt"
    If dieNote = "1" Then
        VollerText = "Sehr Gut"
    End If
    If dieNote = "2" Then
        VollerText = "Gut"
    End If
    If dieNote = "3" Then
        VollerText = "Befriedigend"
    End If
    If dieNote = "4" Then
        VollerText = "Genügend"
    End If
    If dieNote = "5" Then
        VollerText = "Nicht Genügend"
    End If
    If dieNote = "1V" Or dieNote = "1v" Then
        VollerText = "Sehr gut in Vertiefter Allgemeinbildung"
    End If
    If dieNote = "2V" Or dieNote = "2v" Then
        VollerText = "Gut in Vertiefter Allgemeinbildung"
    End If
    If dieNote = "3V" Or dieNote = "3v" Then
        VollerText = "Befriedigend in Vertiefter Allgemeinbildung"
    End If
    If dieNote = "4V" Or dieNote = "4v" Then
        VollerText = "Genügend in Vertiefter Allgemeinbildung"
    End If
    If dieNote = "3G" Or dieNote = "3g" Then
        VollerText = "Befriedigend in Grundlegender Allgemeinbildung"
    End If
    If dieNote = "4G" Or dieNote = "1g" Then
        VollerText = "Genügend in Grundlegender Allgemeinbildung"
    End If
    If dieNote = "5G" Or dieNote = "5g" Then
        VollerText = "Nicht Genügend"
    End If
    If dieNote = "5V" Or dieNote = "5v" Then
        VollerText = "Nicht Genügend"
    End If
End Function
```