Development

- Gradle
  - Firebase per Gradle in Android Studio integrieren
- VBA
  - Letzte beschriebene Zeile in Excel ausgeben
  - Excel Tabelle Zeile fÃ¼r Zeile einlesen
  - Fortschrittsbalken in PowerPoint einfÃ¼gen
  - Ãœberprüfen ob Ordner / Datei vorhanden ist (VBA)
- PHP
  - Konstruktor in PHP

Gradle

Gradle

Firebase per Gradle in Android Studio integrieren

In den neuen Versionen von Android Studio generiert der Projekt Wizard die Build Skripte auf Basis von der empfohlenen Gradle Plugin DSL.

Firebase integrieren

Du musst zuerst die Firebase App erstellen. Dann integrierst du die **json** Datei in dein Projekt.

Die **build.gradle** im Root Verzeichnis sollte so aussehen:

```
plugins {
  ...
  id 'com.google.gms.google-services' version '4.3.10' apply false
}
```

Und deine **./app/build.gradle** sollte so aussehen:

```
plugins {
  ...
  id 'com.google.gms.google-services'
}
```

Und zuletzt musst du die **settings.gradle** im Root Verzeichnis verÃ¤ndern. Dort muss das **google()** ganz oben stehen.

```
pluginManagement {
  repositories {
    google()
    gradlePluginPortal()
  }
}
```

Und nun sollte Firebase geladen werden und du kannst deine App programmieren!

VBA

VBA

Letzte beschriebene Zeile in Excel ausgeben

Wenn du in Excel die letzte beschriebene Zeile benÃ¶tigst, kannst du folgenden Code verwenden. Du musst dort nur das Tabellenblatt Ã¤ndern. Wahlweise kannst du auch eine Long Variable erstellen um grÃ¶ÃŸere 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
```

ErklÃ¤rung

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

VBA

Excel Tabelle Zeile fÃ¼r Zeile einlesen

Einleitung

Wenn du in Excel grÃ¶ÃŸere Tabellen bearbeiten mÃ¶chtest, bietet es sich an diese automatisch durcharbeiten zu lassen. Dieses kannst du Ã¼ber eine For Schleife realisieren. Diese ermittelt erst die letzte Zeile und geht dann Zeile fÃ¼r Zeile durch. In der Schleife selbst kannst du dann den Code angeben den du benÃ¶tigst 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 ErklÃ¤rung

Als erstes werden 2 Variablen deklariert und 1 Objekt instanziert und Bildschirm Aktualisierungen deaktiviert. Die Variable `numberOfRows` erhÃ¤lt die Zeilen Nummer der letzten Zeile. Damit hat die Schleife ein Limit.

Im nÃ¤chsten Schritt, wird die Schleife erstellt. Die Schleife fÃ¤ngt ab Zeile 2 an, da die Tabelle eine Ãœberschrift 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 einfÃ¼gst, hast du von jedem Tabellenblatt Zugriff drauf.

VBA

Fortschrittsbalken in PowerPoint einfÃ¼gen

Einleitung

Wenn wir PrÃ¤sentationen in PowerPoint erstellen, benÃ¶tigen wir unter UmstÃ¤nden einen Fortschrittsbalken in der PrÃ¤sentation. PowerPoint hat nur leider keine eigene Funktion einen Fortschrittsbalken einzufÃ¼gen. Dazu verwenden wir ein Skript um dieses Problem zu umgehen.

Mit dem Skript wird dann ein Fortschrittsbalken generiert der sich auf den Folien langsam vergrÃ¶ÃŸert. Die GrÃ¶ÃŸe wird Prozentual anhand der erstellen Folien generiert.

Anwendung

PrÃ¤sentation 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
```

PrÃ¤sentation 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 kÃ¶nnen wir mithilfe der **RGB Farbcodes** in Zeile **9** anpassen.

VBA

ÃœberprÃ¼fen ob Ordner / Datei vorhanden ist (VBA)

Einleitung

Falls wir Ã¼berprÃ¼fen mÃ¶chten, ob ein bestimmtes Verzeichnis vorhanden ist, kÃ¶nnen wir das nachstehende **Code-Konstrukt** verwenden. Dieses gibt dann einen **boolean Wert** zurÃ¼ck, 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
```

PHP

PHP

Konstruktor in PHP

Wenn wir einen Konstruktor fÃ¼r PHP programmieren wollen, mÃ¼ssen wir innerhalb einer Klasse eine Funktion mit dem Namen **__construct** erstellen. In diese schreiben wir dann den Initialisierungscode.

```
class testClass()
{
	function __construct()
	{
		//Do something
	}
}
```