Excel - User Defined Functions mit Python

Um Python als Programmiersprache für Excel UDFs verwenden zu können muss ein Python Interpreter(z.B. WinPython) und xlwings installiert werden:

Inhalt

1. Python Interpreter installieren (WinPython)

Für Windows wird empfohlen die Python-Distribution WinPython 3.5.xx zu verwenden (Alternativ: Anaconda oder Canopy).

Nachdem WinPython runtergeladen und ausgeführt wurde gibt es den „WinPython-64bit-3.5.xx“ Ordner. **Dieser Ordner sollte nicht im Downloadordner bleiben sondern an einen anderen „sicheren“ Ort z.B. „C:\Programme“ oder „Studium\Semester1\GRM“ o.ä. verschoben werden**.

2. xlwings installieren

Ausführliche Anleitung unter http://docs.xlwings.org/en/stable/installation.html

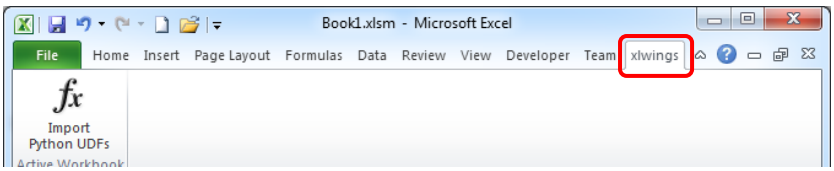
Innerhalb des WinPython Ordners muss die *WinPython Command Prompt.exe* ausgeführt werden. Es erschein eine Kommandozeile in dem der folgende Befehl eingegeben werden muss um xlwings zu installieren (Internetzugang vorausgesetzt):

```
pip install xlwings
```

Nach erfolgreicher Installation muss noch die Excel Erweiterung installiert werden. Dazu im selben Fenster den Befehl

```
xlwings addin install
```

eingeben. Beim nächsten Start von Excel sollte der Reiter xlwings erscheinen.



Excel Trust Center:

Um Excel mit xlwings nutzen zu können muss der Zugriff auf VBA-Projektobjektmodelle vertraut werden. Dazu muss unter Excel->Optionen->Trust Center->Makroeinstellungen der Haken bei „**Zugriff auf das VBA-Projektobjektmodell vertrauen**“ gesetzt werden

3. WinPython als Python Standard-Interpreter festlegen (empfohlen)

Damit Excel weiß, wo Python und damit auch xlwings auf dem Computer installiert sind, bietet es sich an, WinPython als Standard-Interpreter festzulegen. Dazu muss der WinPython Pfad zu den Windows Umgebungsvariablen hinzugefügt werden.

Der Python Interpreter befindet sich im zuvor installierten Ordner **C:\pfad_zu_WinPython\WinPython-64bit-3.5.xx\python-3.5.x.amd64.** Dieser Pfad muss zu der Umgebungsvariable *Path* hinzugefügt werden!

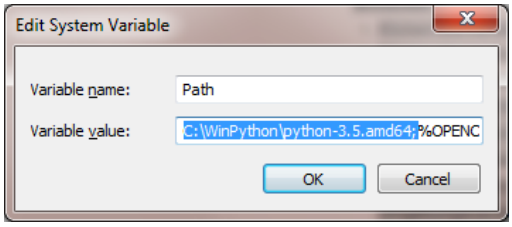
Anleitung zum Ändern der Umgebungsvariablen aus https://www.java.com/de/download/help/path.xml

Windows 10 und Windows 8

1. Suchen Sie in der Suche, und wählen Sie dann: System (Systemsteuerung)
2. Klicken Sie auf den Link **Erweiterte Systemeinstellungen**.
3. Klicken Sie auf **Umgebungsvariablen**. Suchen Sie im Abschnitt **Systemvariablen** die Umgebungsvariable PATH, und wählen Sie sie. Klicken Sie auf **Bearbeiten**. Wenn die Umgebungsvariable PATH nicht vorhanden ist, klicken Sie auf Neu.
4. Geben Sie im Fenster **Systemvariable bearbeiten** (bzw. **Neue Systemvariable**) den Pfad des WinPython Interpreters an (z.B. C:\Programme\WinPython-64bit-3.5.2.1\python-3.5.2.amd64) für die Umgebungsvariable PATH an. **Achtung: Die schon vorhandenen Umgebungsvariablen bitte nicht löschen, sondern den Pfad am Anfang hinzufügen (Taste „Pos 1“ drücken) und mit Semikolon zu den anderen Einträgen abtrennen**. Klicken Sie auf **OK**. Schließen Sie alle verbleibenden Fenster durch Klicken auf **OK**.

Windows 7

1. Klicken Sie mit der rechten Maustaste auf das **Computer**-Symbol auf dem Desktop.
2. Wählen Sie im Kontextmenü die Option **Eigenschaften**.
3. Klicken Sie auf den Link **Erweiterte Systemeinstellungen**.
4. Klicken Sie auf **Umgebungsvariablen**. Suchen Sie im Abschnitt **Systemvariablen** die Umgebungsvariable PATH, und wählen Sie sie. Klicken Sie auf **Bearbeiten**. Wenn die Umgebungsvariable PATH nicht vorhanden ist, klicken Sie auf Neu.
5. Geben Sie im Fenster **Systemvariable bearbeiten** (bzw. **Neue Systemvariable**) den Pfad des WinPython Interpreters an (z.B. C:\Programme\WinPython-64bit-3.5.2.1\python-3.5.2.amd64) für die Umgebungsvariable PATH an. **Achtung: Die schon vorhandenen Umgebungsvariablen bitte nicht löschen, sondern den Pfad am Anfang hinzufügen (Taste „Pos 1“ drücken) und mit Semikolon zu den anderen Einträgen abtrennen.** Klicken Sie auf **OK**. Schließen Sie alle verbleibenden Fenster durch Klicken auf **OK**.



Um zu testen ob WinPython der neue Standard-Interpreter ist, wird eine **neue** Eingabeaufforderung geöffnet: Öffnen Sie mit "**WINDOWS+R**" das Dialogfeld "Ausführen" und geben Sie **cmd.exe** ein.

In dem Fenster geben sie „python“ ein. Danach sollte Python in der Version 3.5.xx starten.

Alternative:

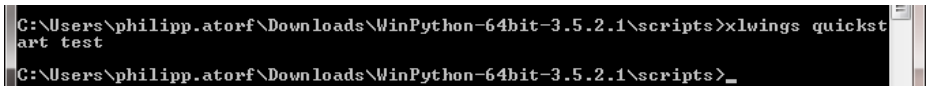
Wer WinPython nicht als Standard-Interpreter hinzufügen will, kann für jedes xlwings Excel Projekt den Interpreter manuell auswählen (siehe Falls WinPython nicht der Standard-Interpreter ist)

4. xlwings Projekt erzeugen und UDFs in Excel mit xlwings nutzen

User Defined Functions (UDF) sind vom Nutzer definierte Funktionen, die spezielle Aufgaben lösen können, die von den Standardfunktionen (z.B. Summe oder Mittelwert) nicht gelöst werden. So ist es beispielsweise möglich eine Funktion Rad2Gon() zu definieren, die Winkel von Bogenmaß nach Gon umrechnet. Excel bietet hierfür die Programmiersprache Visual Basic for Applications (VBA) an, die nicht modern, aber im Moment sehr weit verbreitet und sehr gut in professioneller Software integriert ist. Mit Hilfe von xlwings lassen sich solche Funktionen auch mit der modernen Programmiersprache Python realisieren. Da Python nicht native von Excel unterstützt wird, ist der etwas umständliche Weg über xlwings zu gehen. Um xlwings nutzen zu können, sind einige Vorbereitungen zu treffen:

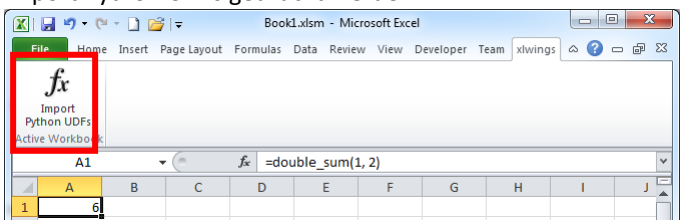
1. xlwings Projekt Erzeugen
   - Ein xlwings Projekt besteht immer aus 2 Dateien. Dabei Handelt es sich um eine Excel Datei (.xlsm) und eine Python Datei (.py). Beide Dateien müssen denselben Namen besitzen. (z.B. test.xlsm und test.py)
   - Der einfachste Weg ein xlwings Projekt zu erzeugen ist mit dem Tool xlwings quickstart. Dazu wird die „WinPython Command Prompt.exe“ im WinPython Ordner gestartet und der folgende Befehl eingegeben (test ist hierbei der Projektname und kann verändert werden):

     ```
     xlwings quickstart test
     ```

   - Damit wird ein Ordner „test“ im Ordner WinPython\scripts erzeugt. Dieser enthält die beiden Dateien test.xlsm und test.py
   - Ist es gewünscht das Projekt an einer anderen Stelle zu speichern, muss der gesamte Ordner „test“ kopiert werden.
2. xlwings Projekt Öffnen
   - In Excel müssen Markoinhalte aktiviert werden und der Zugriff auf das VBA-Projektobjektmodell vertraut werden
     - In Excel: Datei->Optionen->Sicherheitscenter->Einstellungen für das Sicherheitscenter den Haken bei „Zugriff auf das VBA-Projektobjektmodell vertrauen“ setzen
   - Die Exceldatei wird normal mit Excel geöffnet. Die Pythondatei kann mit jedem Texteditor geöffnet werden. Es bietet sich jedoch an *Spyder* zu verwenden (liegt im WinPython Ordner)
3. Erste UDF erstellen
   - Um eine erste eigene Funktion zu erstellen muss in der Python Datei der folgende Code eingefügt werden (auf das Einrücken (Tab) der Zeile „return 2 * (x+y)“ achten):

     ```
     @xw.func
     def double_sum(x, y):
         """Returns twice the sum of the two arguments"""
         return 2 * (x + y)
     ```

   - Danach die Datei speichern
   - Damit xlwings weiß das die Funktion in Excel verwendet werden soll, muss vor der Funktion der Zusatz **@xw.func** stehen
4. UDF Funktionen in Excel laden und benutzen
   - Um die Funktionen aus der Pythondatei zu importieren muss unter dem Reiter xlwings der Knopf Import Python UDFs gedrückt werden.
   - 

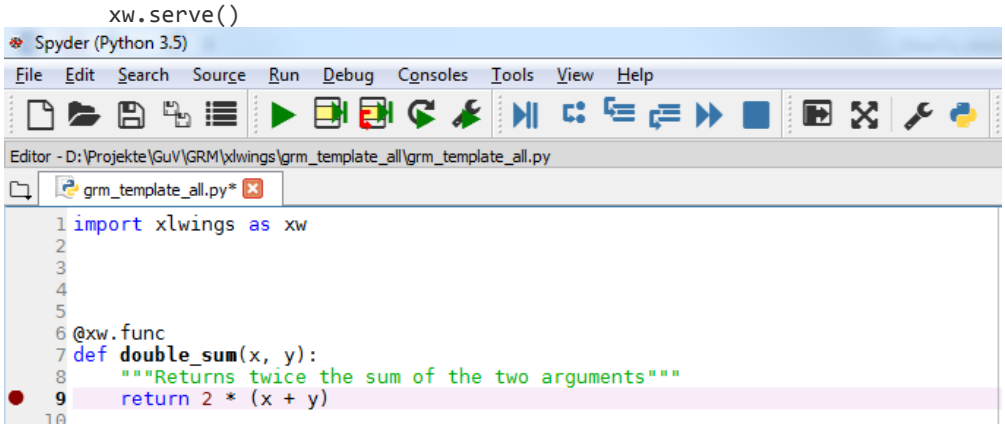
   - Danach kann die Funktion double_sum(1;2) in Excel benutzt werden (Excel trennt Funktionsparameter mit einem Semikolon).

5. Debuggen

Voraussetzung für das Debuggen ist eine Python-Entwicklungsumgebung (Spyder, VS Code, PyCharm, Eclipse o.ä.). Dabei ist das Vorgehen immer gleich:

1. Debuggen in Excel aktivieren
   - In Excel muss der VBA Editor (Alt+F11) und anschließen die Datei xlwings geöffnet werden um die xlwings Einstellungen für die Arbeitsmappe zu ändern.
   - Die Variable „UDF_DEBUG_SERVER“ muss von False auf True geändert werden
   - **Wichtig**: Nachdem das Debuggen beendet ist, muss die Variable wieder auf False gesetzt werden
2. Öffnen der Python Datei mit Spyder
   - Spyder ist eine Entwicklungsumgebung für Python und liegt im Winpython Ordner
   - Nachdem Spyder.exe gestartet ist kann die Python Datei über File->Open geöffnet werden
3. Python Datei vorbereiten
   - Am Ende der Python Datei (z.B. text.py) muss die folgende Bedingung eingefügt werden:

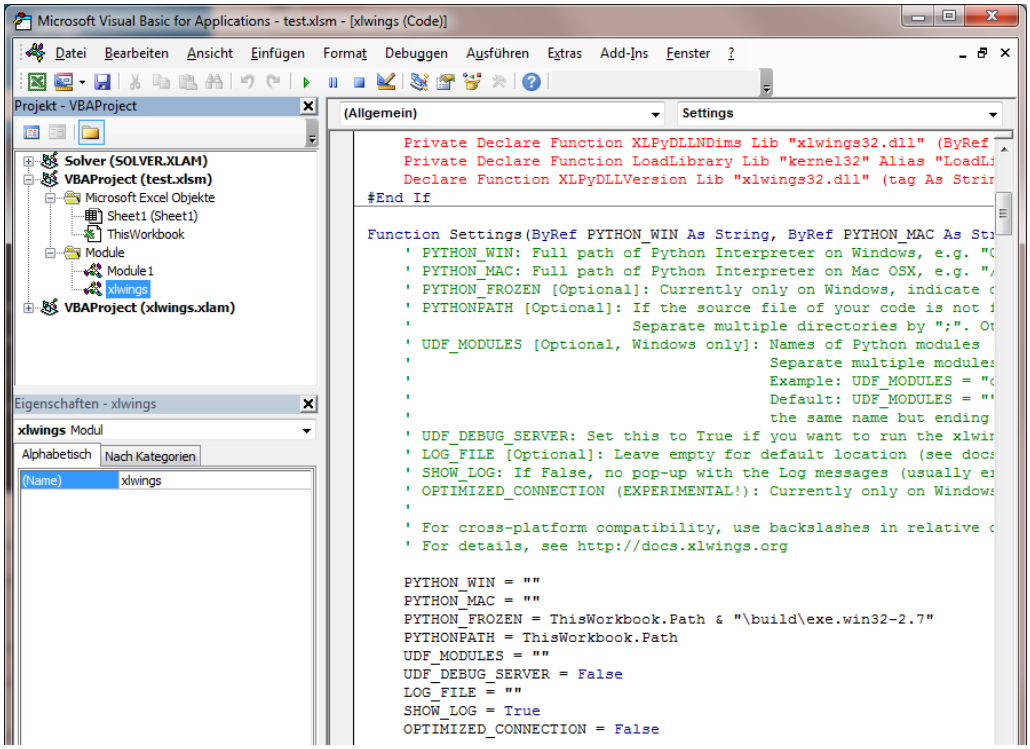
```
if __name__ == '__main__':
    xw.serve()
```



4. Breakpoint setzen
   - Ein Breakpoint dient dazu ein Programm während der Ausführung zu pausieren
   - In Spyder wird ein Breakpoint gesetzt, indem vor die Zeilennummer ein Doppelklick gemacht wird.
   - Es erscheint ein roter Punkt (zum Deaktivieren des Breakpoints ist ein Doppelklick auf diesen notwendig)
5. Debuggen in der Entwicklungsumgebung starten
   - In der Entwicklungsumgebung muss das Debuggen für die Pythondatei gestartet werden (Debug->Debug). In der Konsole sollte dann der Ausdruck
     `xlwings server running, clsid={506E67C3-…}`
     stehen.
6. Excel Funktion ausführen
   - In der Excel Tabelle kann nun die Funktion ausgeführt werden. Anschließend sollte die Funktion an dem vorher definierten Breakpoint halten
7. **Nach dem Debuggen die Variable UDF_DEBUG_SERVER wieder auf False setzen**

6. Falls WinPython nicht der Standard-Interpreter ist

Fall der Schritt „WinPython als Python Standard-Interpreter festlegen (empfohlen)“ **nicht durchgeführt** wurde, gibt es noch die Möglichkeit für jedes xlwings Projekt den Interpreter manuell festzulegen.

1. VBA Editor öffenen
   - Ist das xlwings Excel Projekt geöffnet kann mit der Tastenkombination **Alt+F11** der VBA Editor geöffnet werden.
   - 

2. xlwings Einstellungen öffen
   - Im Projekt-Explorer (open links) kann unter Module xlwings geöffnet werden.
   - Im Hauptfenster erscheint der xlwings VBA Code. In der Funktion Settings können verschiedenen Anpassungen vorgenommen werden.
3. Interpreter Pfad ändern
   - Die Variable PYTHON_WIN enthält standardmäßig keinen Eintrag (““). Die bedeutet, dass der Standard-Interpreter verwendet wird.
   - Um einen anderen Interpreter zu benutzen muss die Variable verändert werden. Für WinPython wird sie wie folgt verändert (b:
     “C:\Pfad_zu_Winpython\WinPython-64bit-3.5.2.1\python-3.5.2.amd64 \pythonw.exe”
     Dies ist der Dateipfad WinPython pythonw.exe.
     Anschließend muss die xlwings Datei im VBA Editor gespeichert werden