

# Willkommen

beim Excel-Stammtisch München

## Office JS & Excel





# Danke

## Euch

» You join here, not to be cool, but to make other cool\_

Satya Nadella

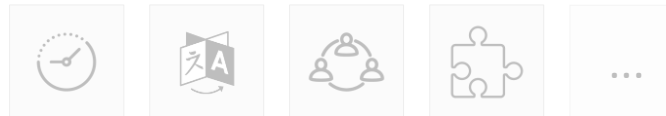
## René

» Simply Excel ;-)



## The Noun Project




» [thenounproject.com](https://thenounproject.com)








# Über mich

## Mourad Louha

-  Microsoft Most Valuable Professional (9-mal)
-  Anwendungsentwickler
-  Buchautor

## Websites

-  excel-translator.de
-  excel-ticker.de
-  excel-soccer.de



Excel-Translator  
Deutsch

Übersetzer

Funktionen v

Argumente v

Blog

Statistiken v

### Übersetzer

Die Excel-Funktionen wurden von Microsoft in verschiedenste Sprachen übersetzt. Wenn Sie eine Arbeitsmappe mit jemanden austauschen, der eine andere Sprache für Excel verwendet als Sie, werden auch die in der Mappe enthaltenen Funktionen und Formeln automatisch übersetzt. Wenn Sie jedoch eine fremdsprachliche Excel-Formel direkt in Ihre Excel-Version einfügen, wird diese in der Regel nicht erkannt. Ein solcher Fall kann beispielsweise vorkommen, wenn Sie Excel in Deutsch verwenden und die in einem Forum angegebene englische Formellösung verwenden möchten. Mit dem folgenden Online-Tool können Sie eine Excel-Formel von einer Sprache in eine andere Sprache übersetzen und die lokalisierte Formel verwenden.

Microsoft Excel ?  

Microsoft Excel 2016

Ausgangsformel ?  

Fügen Sie eine einzelne Funktion oder eine Formel in dieses Feld ein.

Ausgangssprache ?  

Deutsch

Zielsprache ?  

Englisch



# Über mich

## Tools

- Eigenes User Interface
- Mehrsprachig
- Datenbankgestützt
- ...

Excel Sharp DIP

Übersicht

Grund-Einstellungen

Spracheinstellungen

aktive Sprache: Deutsch

Datenbanken

Pfad: D:\Tools\DIP\Databases

Verzeichnisse

Pfad: D:\Tools\DIP\System

Anwendung

D:\Tools\DIP

Dokumente

D:\Tools\DIP\Documents

Export

D:\Tools\DIP\Export

Import

D:\Tools\DIP\Import

Medien

D:\Tools\DIP\Media

Datensätze

Aktuell

Kartenliste

0 Einträge

Artikel

0 Einträge

Aktualisieren...

Datenbank: D:\Tools\DIP\Databases\Kartens.200.mdb

Benutzer: Mourad Louha

Disponent: Alle

Import: Importvorgang durchführen

Importdaten: Bestehende Einträge überschreiben

Importdateien: 01: Importdatei.xlsx (Datei nicht gefunden), 02: Bestand.xlsx (Datei nicht gefunden), 03: Lagerliste ACS.xlsx (Datei nicht gefunden), 04: Lagerliste ACS.xlsx (Datei nicht gefunden), 05: Lagerliste RGS.xlsx (Datei nicht gefunden), 06: Lagerliste ACS.xlsx (Datei nicht gefunden)

Zellen: Überschrift (Zeile 1), Daten (Zeile 2)

Spalten: 01: Artikelnummer (Spalte A), 21: Artikelbezeichnung (Spalte B), 22: Teilgröße (Spalte C), 24: Stückprogramm (Spalte D), 25: Behandlungsprogramm (Spalte E), 26: Farbe (Spalte F)

Berechtigungen:

Excel Sharp PRL

KVCU » Price list

Mourad Louha

#	Page	Articles	Images	Texts	Template	Hierarchy
0171	Designation	Articles	Images	Texts	Designation	Level 1
0001	Cover	0	1	2	Cover	
0002	TEST	0	0	0		
0003	Products and Factories	0	0	0		
0004	Product 0004	0	0	0		
0005	Product 0005	28	0	0		
0006	Product 0006	25	0	0		
0007	Product 0007	24	0	0		
0008	Product 0008	24	0	0		
0009	Product 0009	23	0	0		
0010	Product 0010	27	0	0		
0011	Product 0011	24	0	0		
0012	Product 0012	23	0	0		
0013	Product 0013	23	0	0		
0014	Product 0014	22	0	0		
0015	Product 0015	0	0	0		

Publish...

Filter: Hierarchy

Valid from: September 2020

Customer: General (Gross prices)

Price list pages: 01: TEST, 02: Products and Factories, 03: Product 0004, 04: Product 0005, 05: Product 0006, 06: Product 0007, 07: Product 0008, 08: Product 0009, 09: Product 0010, 10: Product 0011, 11: Product 0012, 12: Product 0013, 13: Product 0014, 14: Product 0015

Template: Cover, Cover-Date

Options: Do not create an Excel file, Microsoft Excel

PDF-Printer: Samsung C430W (32, 358, 176, 30)

Notifications: If you use this printer, all created pages will be saved as single pages to the export folder. The merging of these pages to one PDF document must be done manually through an external application. The PDF-Printer is available on your system. Click on the green tick to start publishing.

Markets

Version 1.05 - Build 150920

Home

Base

KPIs

Rankings

Reports

Export

Print

Previous

Next

Base » Logic

Overview

Data

Map

Logic

Information

Type: Type A

Notifications:

Audit Weights

Defaults

	A	B	C	D	E	F	G	-	Σ
	Active	Active	Active	Active	Active	Active	Active	Inactive	
Standard Weights	50	10	5	3	10	7	15	0	100
Category	A	B	C	D	E	F	G	-	Σ
Category 01	50	10	5	3	10	7	15	-	100
Category 02	50	10	5	3	10	7	15	-	100
Category 03	50	10	5	3	10	7	15	-	100
Category 04	50	10	5	3	10	7	15	-	100
Category 05	50	10	5	3	10	7	15	-	100
Category 06	50	10	5	3	10	7	15	-	100
Category 07	50	10	5	3	10	7	15	-	100



# Inhalt



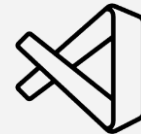
**Überblick**



**Script Lab**



**Konzepte**



**Visual Studio Code**



**Objektmodell**



**Office Scripts**



# Inhalt



**Überblick**



**Script Lab**



**Konzepte**



**Visual Studio Code**



**Objektmodell**

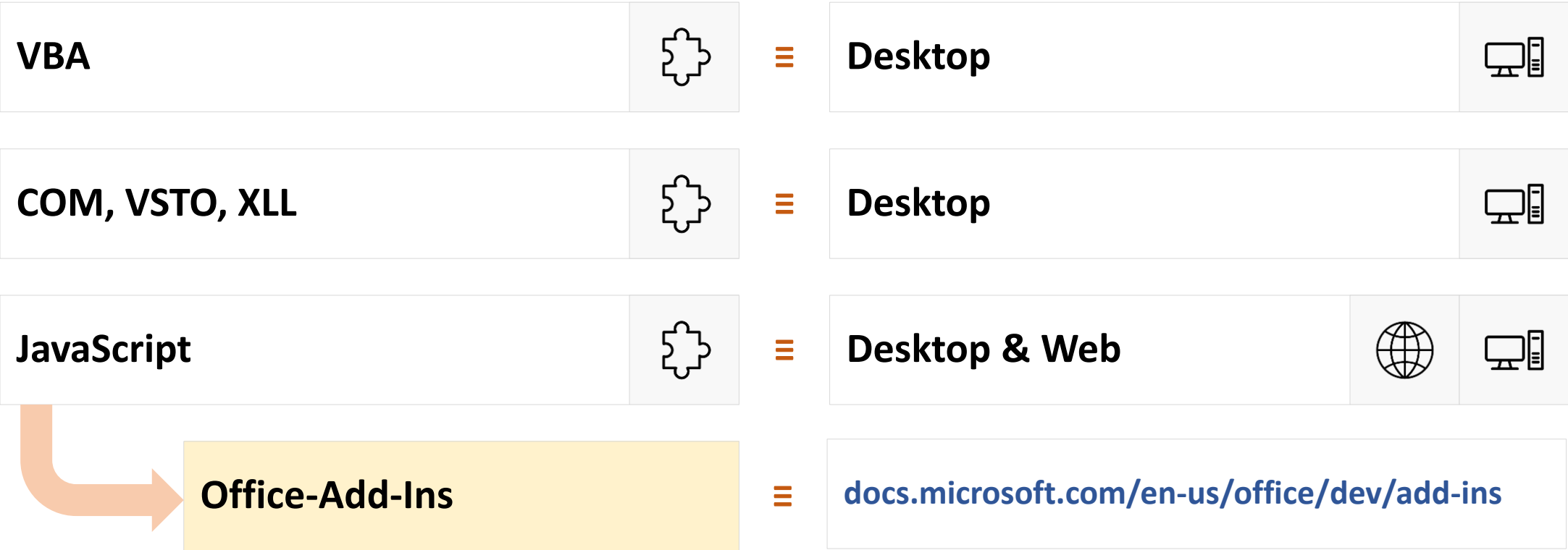


**Office Scripts**



# Überblick

## Office-Erweiterungen

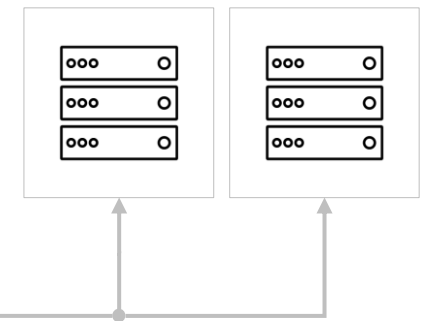
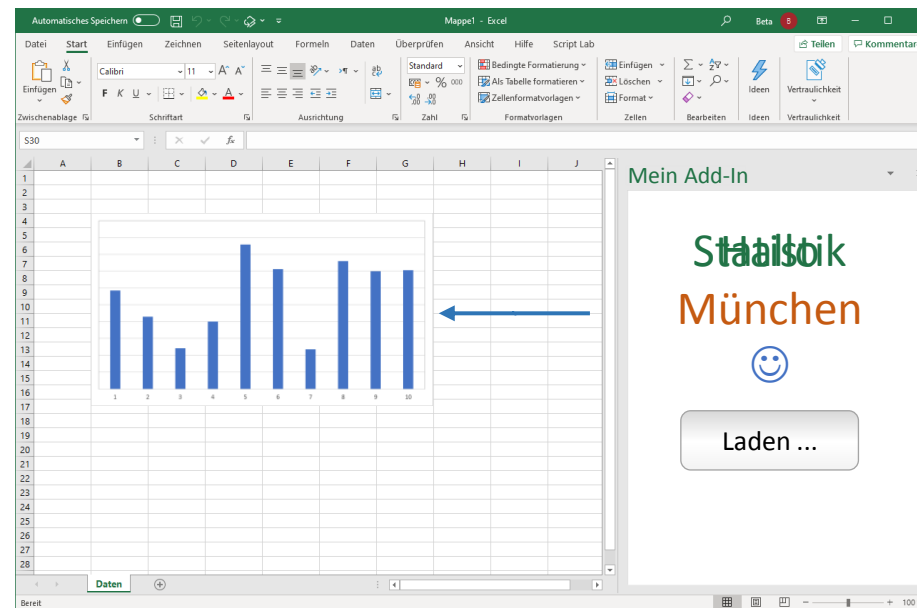




# Überblick

## Office-Add-Ins

- Plattform
- Webanwendungen
- Server
- Office JS



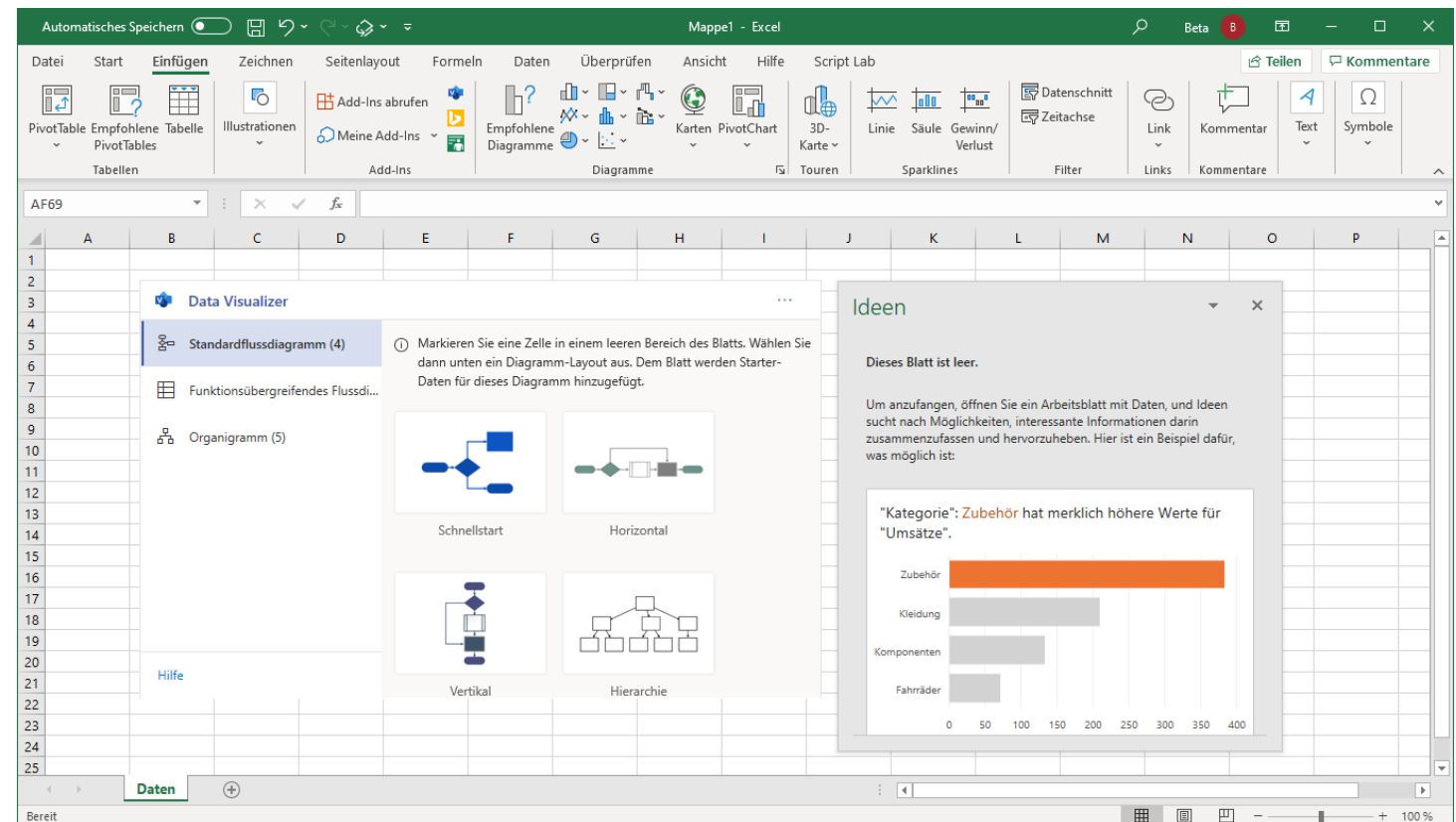




# Überblick

## Office-Add-Ins-Typen





- **Objekte**
- **Aufgabenbereiche**
- **Menüband**

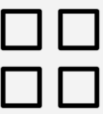




# Überblick

## Office JS API

• Allgemein		≡	Common API	
• Excel		≡	ExcelApi	≡ Version 1.12
• PowerPoint		≡	PowerPointApi	≡ Version 1.1
• Word		≡	WordApi	≡ Version 1.3
• Outlook		≡	Mailbox	≡ Version 1.8
• Visio, OneNote, Project				



# Überblick

## Office JS & VBA

VBA	
Niedrige Lernkurve	
Plattformgebunden	
Veraltete IDE	
Nur notwendige Updates	
Objektmodell vollständig	

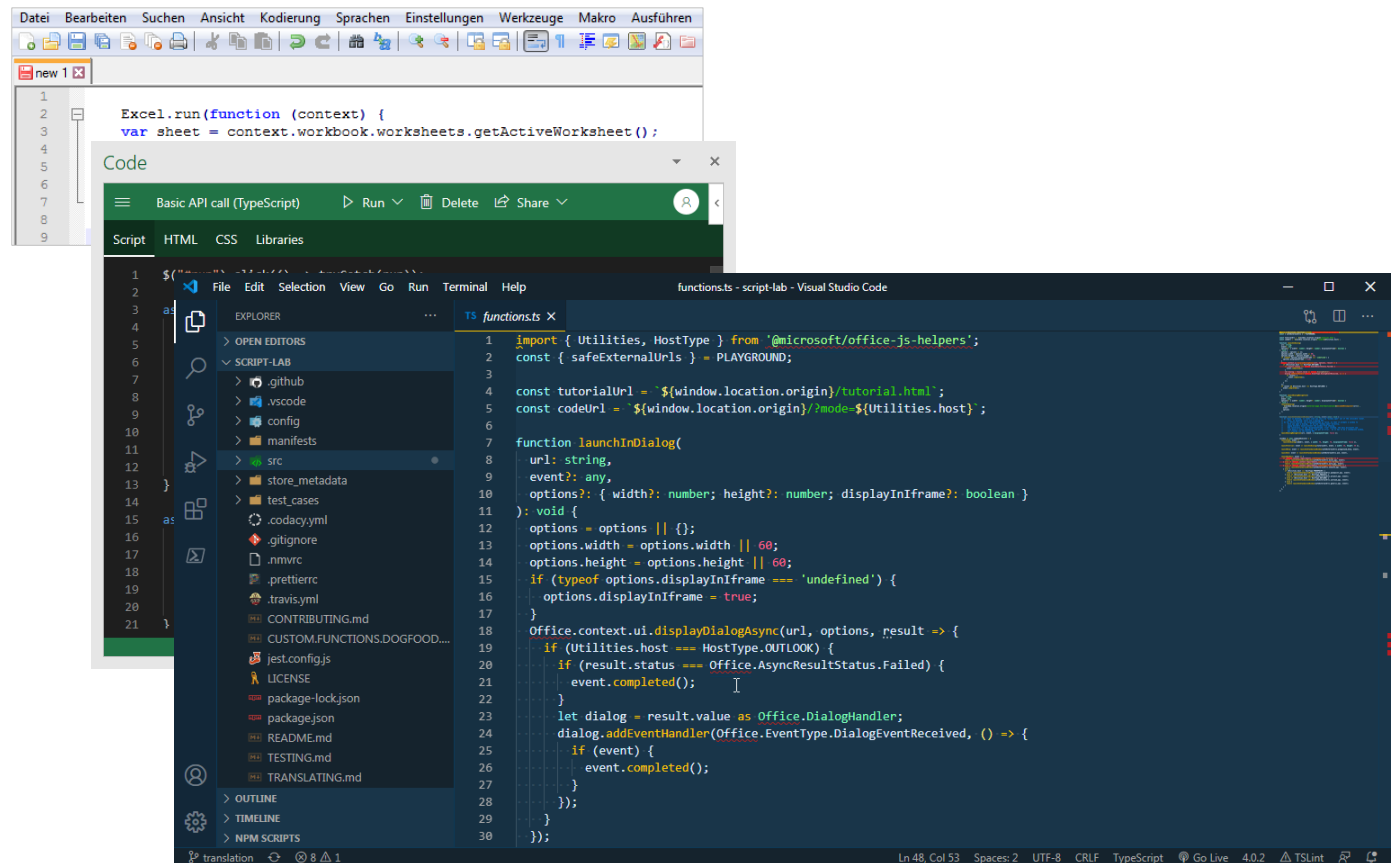
Office JS	
Hohe Lernkurve	
Plattformübergreifend	
Moderne Tools	
Konstante Weiterentwicklung	
Objektmodell <b>noch</b> unvollständig	



# Überblick

## Office JS Entwicklertools

- Notepad++
- Script Lab
- Visual Studio Code
- Visual Studio
- ...





# Inhalt



Überblick



Script Lab



Konzepte



Visual Studio Code



Objektmodell

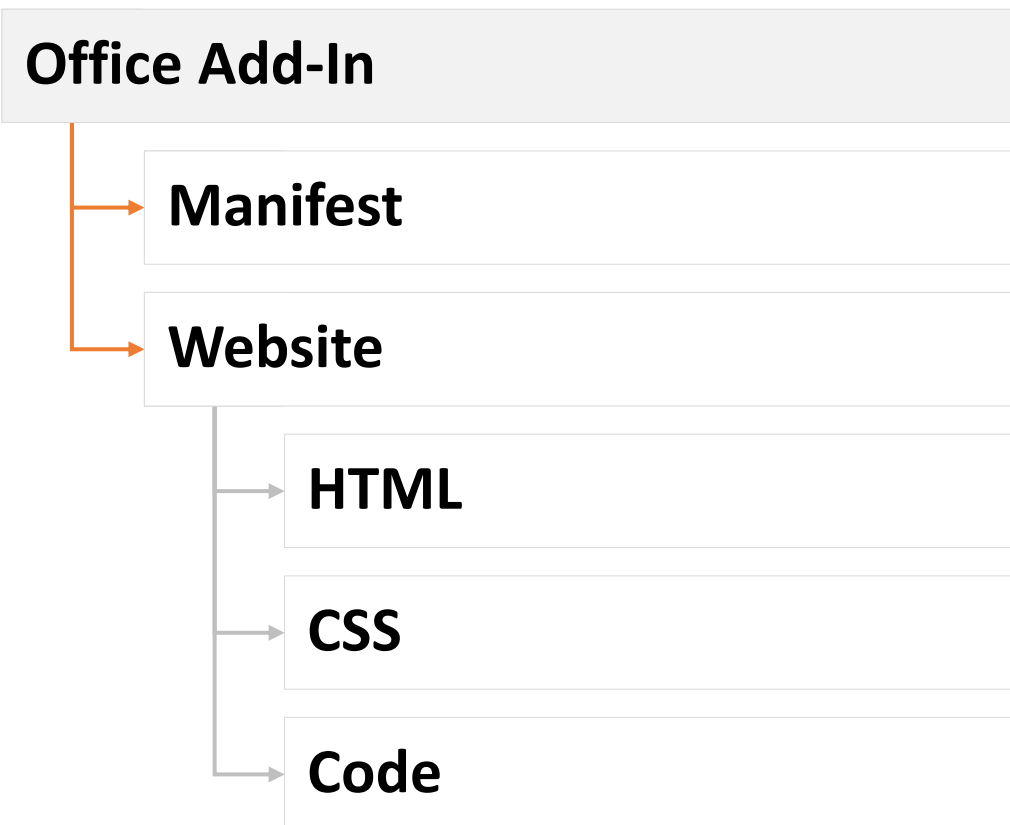


Office Scripts



# Konzepte

## Aufbau



```
▼ MUNICH-01
> .vscode
> assets
> node_modules
▼ src
  ▼ commands
    <> commands.html
    TS commands.ts
  ▼ taskpane
    # taskpane.css
    <> taskpane.html
    TS taskpane.ts
  .eslintrc.json
  CONTRIBUTING.md
  LICENSE
  manifest.xml
  package-lock.json
  package.json
  README.md
  tsconfig.json
  webpack.config.js
```



# Konzepte

## Manifest

- Typ
- Name
- Hosts
- Menüband
- ...

```
manifest copy.xml
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <OfficeApp xmlns="http://schemas.microsoft.com/office/appforoffice/1.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3           xmlns:bt="http://schemas.microsoft.com/office/officeappbasictypes/1.0"
4           xmlns:ov="http://schemas.microsoft.com/office/taskpaneappversionoverrides"
5           xsi:type="TaskPaneApp">
6   <Id>c944bb47-b09e-4b59-985f-546b8cce67e9</Id>
7   <Version>1.0.0.0</Version>
8   <ProviderName>Test</ProviderName>
9   <DefaultLocale>en-US</DefaultLocale>
10  <DisplayName DefaultValue="munich-01"/>
11  :
12  :
13  <Permissions>ReadWriteDocument</Permissions>
14  <VersionOverrides xmlns="http://schemas.microsoft.com/office/taskpaneappversionoverrides" xsi:type="VersionOverridesV1_0">
15    <Hosts>
16      <Host xsi:type="Workbook">
17        <DesktopFormFactor>
18          :
19          :
20          <FunctionFile resid="Commands.Url"/>
21          <ExtensionPoint xsi:type="PrimaryCommandSurface">
22            <OfficeTab id="TabHome">
23              <Group id="CommandsGroup">
24                <Label resid="CommandsGroup.Label"/>
25                :
26                :
27                <Control xsi:type="Button" id="TaskpaneButton">
28                  <Label resid="TaskpaneButton.Label"/>
29                  <Supertip>
```

<https://docs.microsoft.com/de-de/office/dev/add-ins/develop/add-in-manifests>



# Konzepte

## HTML

- Hypertext Markup Language
- Tags
- Attribute

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8" />
  <title>Titel des Dokuments</title>
  <script type="text/javascript" src="javascript.js"></script>
  <link rel="stylesheet" href="styles.css"/>
</head>
<body>
  <header class="Kopfzeile">
    
    <h1>Willkommen</h1>
  </header>
  <main id="app" class="Inhalt">
    <h2>Ein einfaches Add-In für Excel</h2>
    <ul>
      <li><span>Hallo München :-)</span></li>
      <li><span>Hallo Aachen :-)</span></li>
    </ul>
    <div role="button" id="run">
      <span>Ausführen</span>
    </div>
  </main>
</body>
</html>
```

<https://wiki.selfhtml.org/wiki/HTML>





# Konzepte

## CSS

- Cascading Style Sheets
- Selektoren
- Klassen & IDs
- Eigenschaften
- Funktionen
- Effekte

```
html,
body {
  width: 100%;
  height: 100%;
  margin: 0;
  padding: 0;
}

ul {
  margin: 0;
  padding: 0;
}

.Kopfzeile {
  padding: 20px;
  padding-bottom: 30px;
  padding-top: 100px;
  display: -webkit-flex;
  display: flex;
}

.Inhalt {
  align-items: center;
  padding: 10px 20px;
}
```

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8" />
  <title>Titel des Dokuments</title>
  <script type="text/javascript" src="javascript.js"></script>
  <link rel="stylesheet" href="styles.css"/>
</head>
<body>
  <header class="Kopfzeile">
    
    <h1>Willkommen</h1>
  </header>
  <main id="app" class="Inhalt">
    <h2>Ein einfaches Add-In für Excel</h2>
    <ul>
      <li><span>Hallo München :-)</span></li>
      <li><span>Hallo Aachen :-)</span></li>
    </ul>
    <div role="button" id="run">
      <span>Ausführen</span>
    </div>
  </main>
</body>
</html>
```

<https://wiki.selfhtml.org/wiki/CSS>



# Konzepte

## JavaScript

- Plattformübergreifend
- Nicht typisiert
- Standardbibliotheken
- Objektorientiert
- ECMA 2015 bzw. ES6

```
// Beispiel 1  
  
function Quadrat(Zahl)  
{  
    return Zahl * Zahl;  
}
```

```
// Prototypbasiert  
  
function Person(Vorname, Nachname) {  
    this.Vorname = Vorname;  
    this.Nachname = Nachname;  
}  
  
Person.prototype.Name = function() {  
    console.log(this.Vorname + ' ' + this.Nachname);  
}  
  
function Anwalt(Vorname, Nachname, Fachgebiet) {  
    Person.call(this, Vorname, Nachname);  
    this.Fachgebiet = Fachgebiet;  
}
```

<https://developer.mozilla.org/de/docs/Web/JavaScript/Guide>



# Konzepte

## TypeScript

- Erweitertes JavaScript
- Typisiert
- Kompilierung

```
export function bufferToHexString(buffer: ArrayBuffer | Uint8Array) {
    return Array.from(buffer instanceof Uint8Array ? buffer : new Uint8Array(buffer))
        .map(value => /* pad with, at most, two 0s */ ('00' + value.toString(16)).slice(-2))
        .join('');
}

export function hexStringToBuffer(input: string) {
    const numbers = input.match(/.{1,2}/g).map(item => parseInt(item, 16));
    const buffer = new ArrayBuffer(numbers.length);
    const uint8array = new Uint8Array(buffer);
    numbers.forEach((num, index) => (uint8array[index] = num));
    return buffer;
}

export function unicodeStringToBuffer(input: string) {
    const buffer = new ArrayBuffer(input.length * 2); // 2 bytes for each char
    const uint16arr = new Uint16Array(buffer);
    for (let i = 0; i < input.length; i++) {
        uint16arr[i] = input.charCodeAt(i);
    }
    return buffer;
}
```

<https://www.typescriptlang.org>



# Konzepte

## Unterschied let ↔ var

- Gültigkeitsbereich
- Initialisierungsverhalten
- Neudeklarationsverhalten

```
function Unterschied_1() {  
  var A = "Hello";  
  let B = "world";  
  
  console.log(A, B); // Liefert Hello world  
  
  {  
    let C = "München";  
    console.log(C); // Liefert München  
  }  
  
  console.log(C); // Liefert Fehlermeldung  
}  
  
function Unterschied_2() {  
  console.log(A); // Liefert undefined  
  console.log(B); // Liefert Fehlermeldung  
  
  var A = "Test A";  
  let B = "Test B";  
  
  console.log(A); // Liefert Test A  
  console.log(B); // Liefert Test B  
}
```



# Konzepte

## Excel-Add-In Grundgerüst

- Excel.run
- Async
- Await
- Promise

// Typescript

```
async function TS_Run() {  
    await Excel.run(async (context) => {  
        let range = context.workbook.getSelectedRange();  
        range.format.fill.color = "yellow";  
        range.load("address");  
  
        await context.sync();  
        console.log(range.address);  
    });  
}
```

```
async function TS_TryCatch(callback) {  
    try {  
        await callback();  
    } catch (error) {  
        console.error(error);  
    }  
}
```

TS\_TryCatch(TS\_Run);

// Javascript

```
function JS_Run() {  
    return Excel.run(function(context) {  
        var range = context.workbook.getSelectedRange();  
        range.format.fill.color = "yellow";  
        range.load("address");  
  
        return context.sync().then(function() {  
            console.log(range.address);  
        });  
    });  
}
```

```
function JS_TryCatch(callback) {  
    Promise.resolve()  
        .then(callback)  
        .catch(function(error) {  
            console.error(error);  
        });  
}
```

JS\_TryCatch(JS\_Run);



# Konzepte

## Bibliotheken

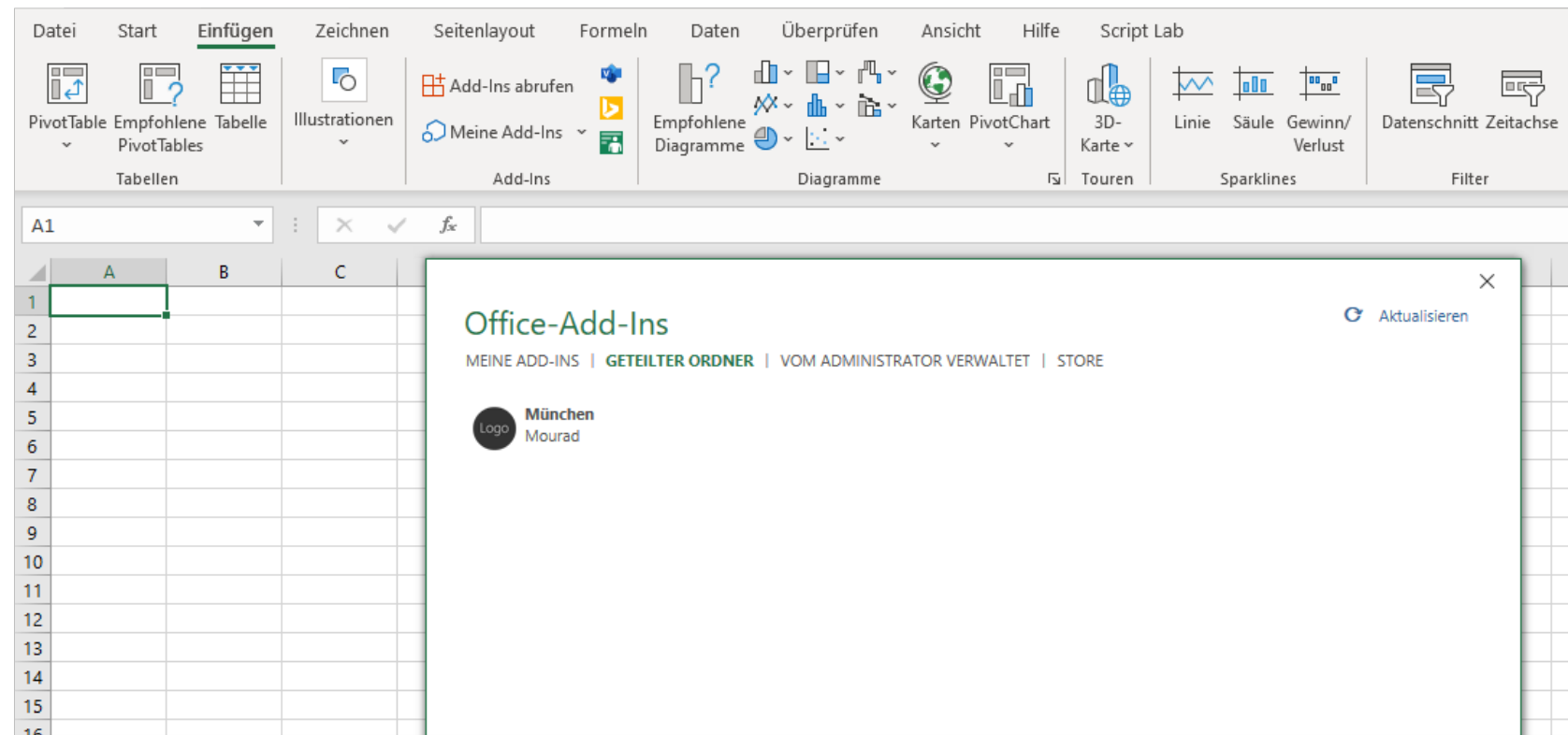
- **Office**  <https://appsforoffice.microsoft.com/lib/1/hosted/office.js>
- **Frameworks · JS** 
  - React**
  - Angular**
  - Vue.js**
  - JQuery**
  - ... ..
- **Frameworks · CSS** 
  - Fluent UI**
  - Office UI Fabric**
  - Bootstrap**
  - Foundation**
  - ... ..



# Konzepte

## Sideload

- Freigabe
- Manifest
- Katalog
- Server
- Installieren





# Inhalt



Überblick



Script Lab



Konzepte



Visual Studio Code



**Objektmodell**



Office Scripts

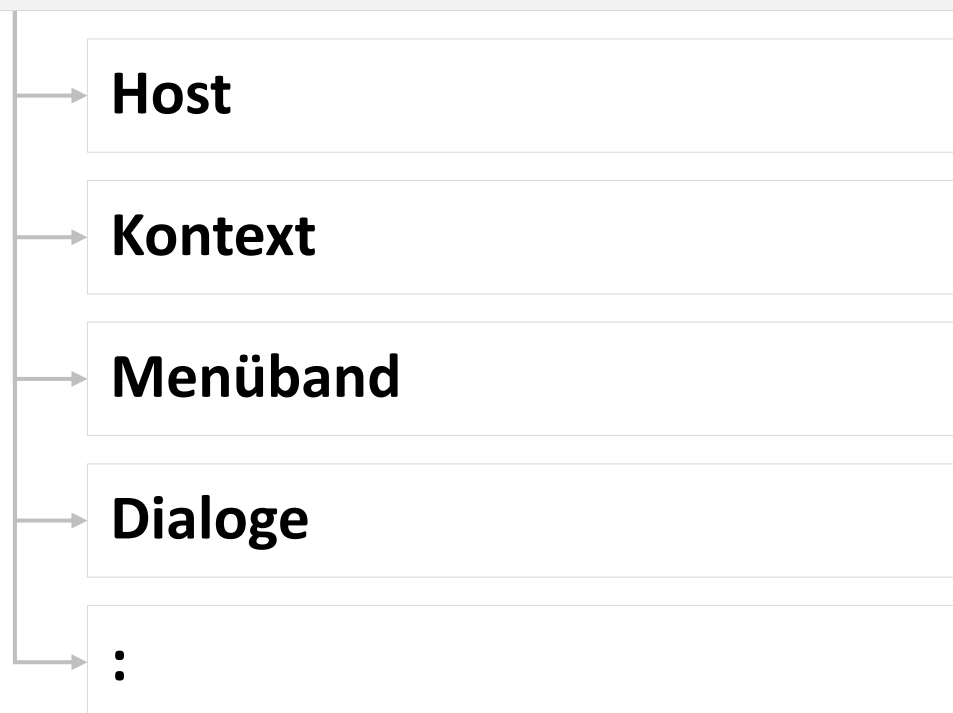




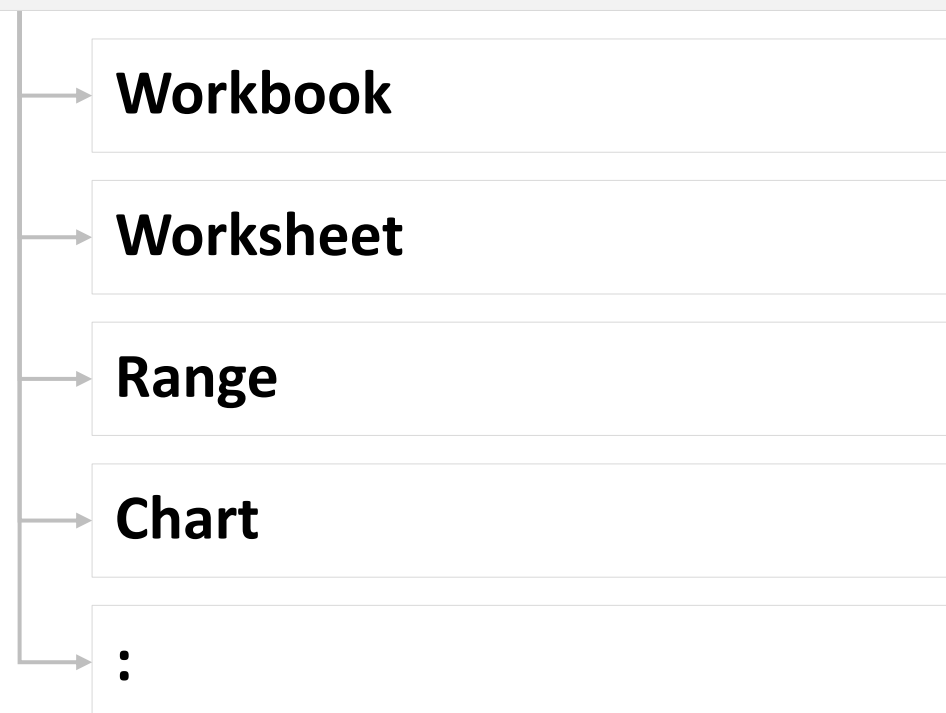
# Objektmodell

## Aufbau

### Common API



### Excel API





# Objektmodell

## Common API

### Beispiel

displayLanguage

DocumentMode

config

de-DE

```
Object {ReadOnly: "readOnly", Readwrite: "readwrite"}
```

```
Object {extendedErrorLogging: false}
```

```
async function Beispiel() {
  await Excel.run(async (context) => {

    let x = Office.context.displayLanguage;
    let y = Office.DocumentMode;
    let z = OfficeExtension.config;

    await context.sync();
    console.log(x, y, z);
  });
}

async function tryCatch(callback) {
  try {
    await callback();
  } catch (error) {
    console.error(error);
  }
}

tryCatch(Beispiel);
```



# Objektmodell

## Excel API

### Beispiel

Blattname

Zelladresse

Formel

Tabelle1

Tabelle1!A1

["=XLOOKUP("Mich",A3:A15,B3:B15,"Nope",0,1)"]

```
async function Beispiel() {
  await Excel.run(async (context) => {
    let x = context.workbook.worksheets.getActiveWorksheet();
    let y = x.getRange("A1");
    let z = y

    x.load("name");
    y.load("address, formulas");

    await context.sync();
    console.log(x.name, y.address, z.formulas[0]);
  });
}

async function tryCatch(callback) {
  try {
    await callback();
  } catch (error) {
    console.error(error);
  }
}

tryCatch(Beispiel);
```



# Objektmodell

## Excel API

### Beispiel · Ereignis

→ Ereignis registrieren

→ Excel.run beachten

→ Ereignis abfangen

B2

26.25

10.5

```
async function Beispiel()
{
    await Excel.run(async (context) =>
    {
        let x = context.workbook.worksheets.getActiveWorksheet();

        x.onSingleClicked.add((event) =>
        {
            return Excel.run((context) =>
            {
                console.log(event.address, event.offsetX, event.offsetY);
                return context.sync();
            });
        });

        await context.sync();
    });
}

async function tryCatch(callback) {
    try {await callback();} catch (error) {
        console.error(error);
    }
}

$("#run").click(() => tryCatch(Beispiel));
```



# Inhalt



Überblick



Konzepte



Objektmodell



Script Lab



Visual Studio Code













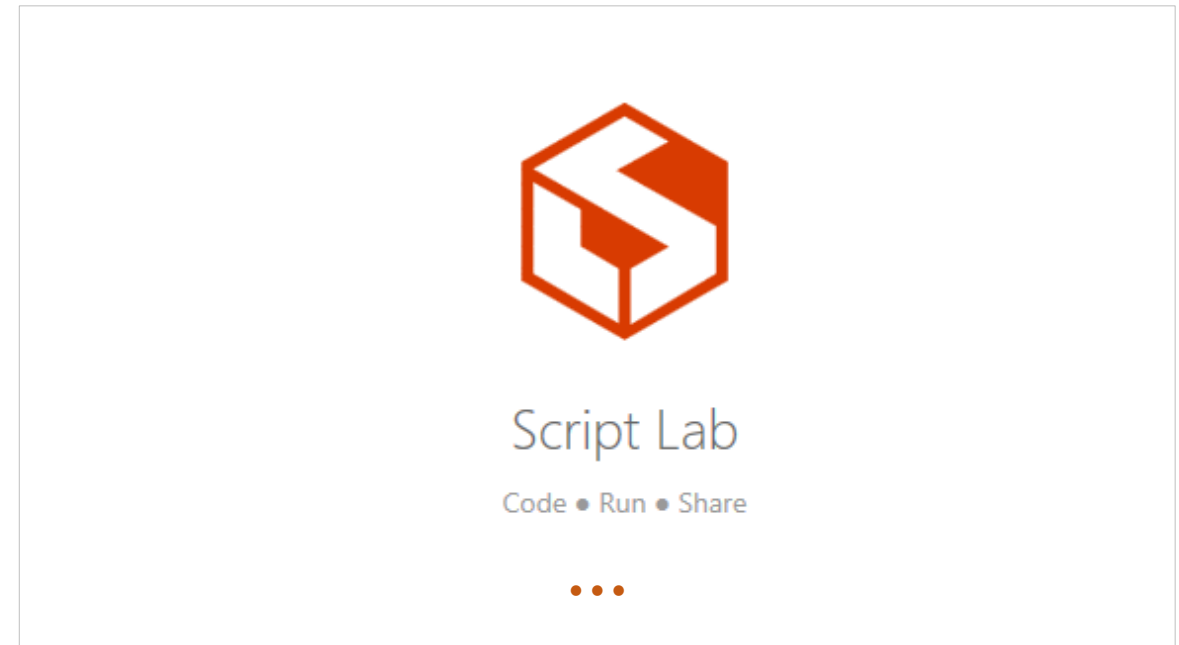
Office Scripts



# Script Lab

## Überblick

- **Add-In**     
- **Open Source**  **TypeScript**
- **Office 365**  **Empfohlen**
- **Code**  **Implementieren**
- **Run**  **Ausführen**
- **Share**  **Teilen**

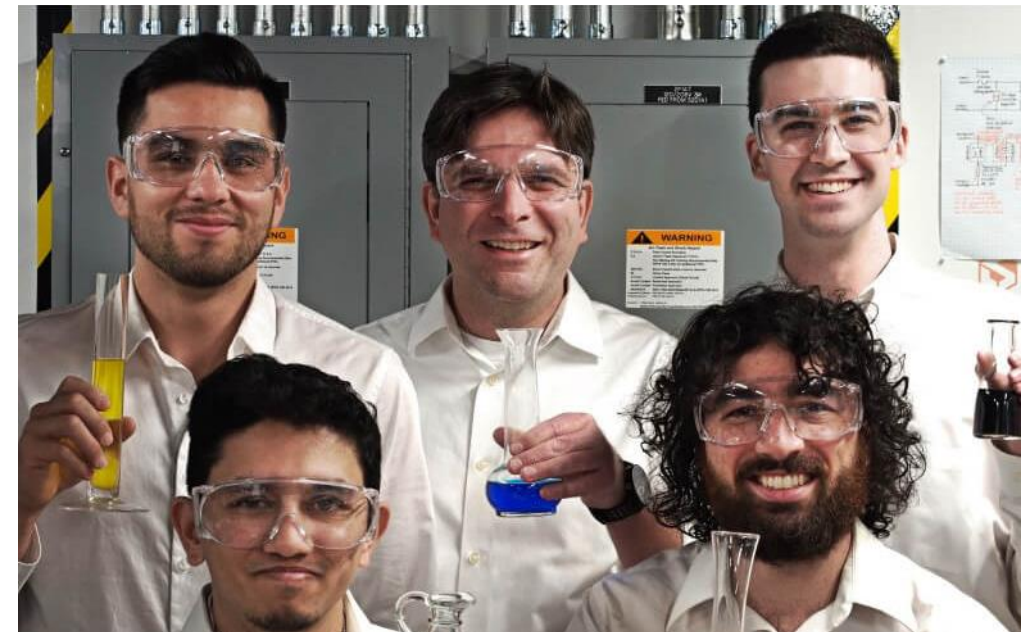




# Script Lab

## Historie

- Microsoft Garage Projekt
- Erste Version 2017
- Neuimplementierung 2018
- Script Lab für Outlook
- [github.com/OfficeDev/script-lab](https://github.com/OfficeDev/script-lab)



Daniel M. Galan, Jakob Nielsen, Michael Saunders  
Bhargav Krishna, Michael Zlatkovsky

<https://www.microsoft.com/en-us/garage/profiles/script-lab>

<https://leanpub.com/buildingofficeaddins>



# Script Lab

## Installation

- Befehl **Add-Ins abrufen**
- **Script** suchen
- **Pro Anwendung**
- **Pro User**
- **Pro Sprache**

Office-Add-Ins

MEINE ADD-INS | VOM ADMINISTRATOR VERWALTET | STORE

Add-Ins können auf persönliche Informationen und auf Dokumentinformationen zugreifen. Wenn Sie ein Add-In verwenden, stimmen Sie seinen Berechtigungen, Lizenzbedingungen und Datenschutzrichtlinien zu.

Script

Für Sie vorgeschlagen ▾

Kategorie

Alle

CRM

Datenanalyse

Finanzverwaltung


Hilfsprogramme

Produktivität

Schulungen und Lernprogramme

Vertrieb und Marketing

Visualisierung



**Script Lab, a Microsoft Garage project**  
Create, run, and share your Office Add-in code snippets from within Excel, Word, or PowerPoint.  
★★★★☆ (76)

Hinzufügen

Hinzufügen

Hinzufügen

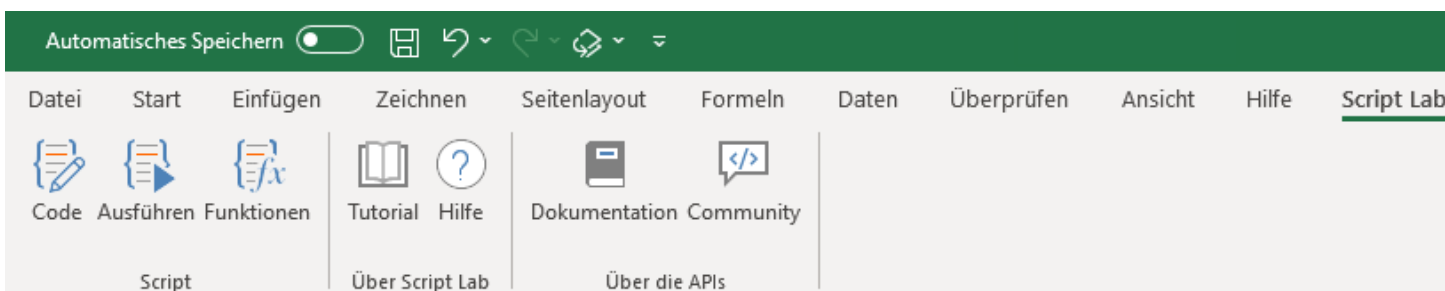




# Script Lab

## Menüband

- **Code**
- **Ausführen**
- **Funktionen**
- **Tutorial & Hilfe**
- **Dokumentation & Community**





# Script Lab

## Code

- Aufgabenbereich
- Editierfenster
- Syntax-Hervorhebung
- IntelliSense
- Optionen

```
Code
Basic API call (TypeScript) Run Delete Share
Script HTML CSS Libraries
1 $("#run").click(() => tryCatch(run));
2
3 async function run() {
4   await Excel.run(async (context) => {
5     const range = context.workbook.getSelectedRange();
6     range.format.fill.color = "yellow";
7     range.load("address");
8
9     await context.sync();
10
11     console.log(`The range address`);
12   });
13 }
14
15 /** Default helper for invoking an async function */
16 async function tryCatch(callback) {
17   try {
18     await callback();
19   } catch (error) {
20     // Note: In a production add-in, you'd want to notify the user through your add-in's UI.
21     console.error(error);
22   }
23 }
24
TypeScript Dark
```

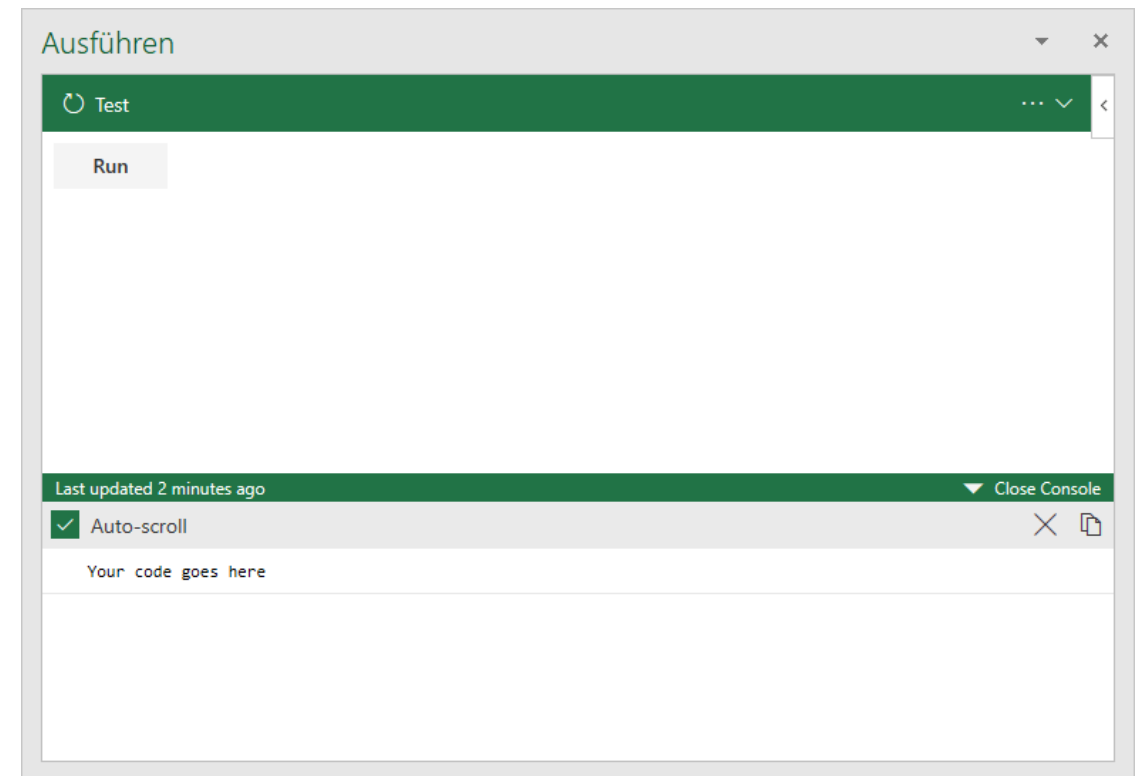
<https://microsoft.github.io/monaco-editor/index.html>



# Script Lab

## Ausführen

- **Aufgabenbereich**
- **Konsole**
- **Synchronisation**

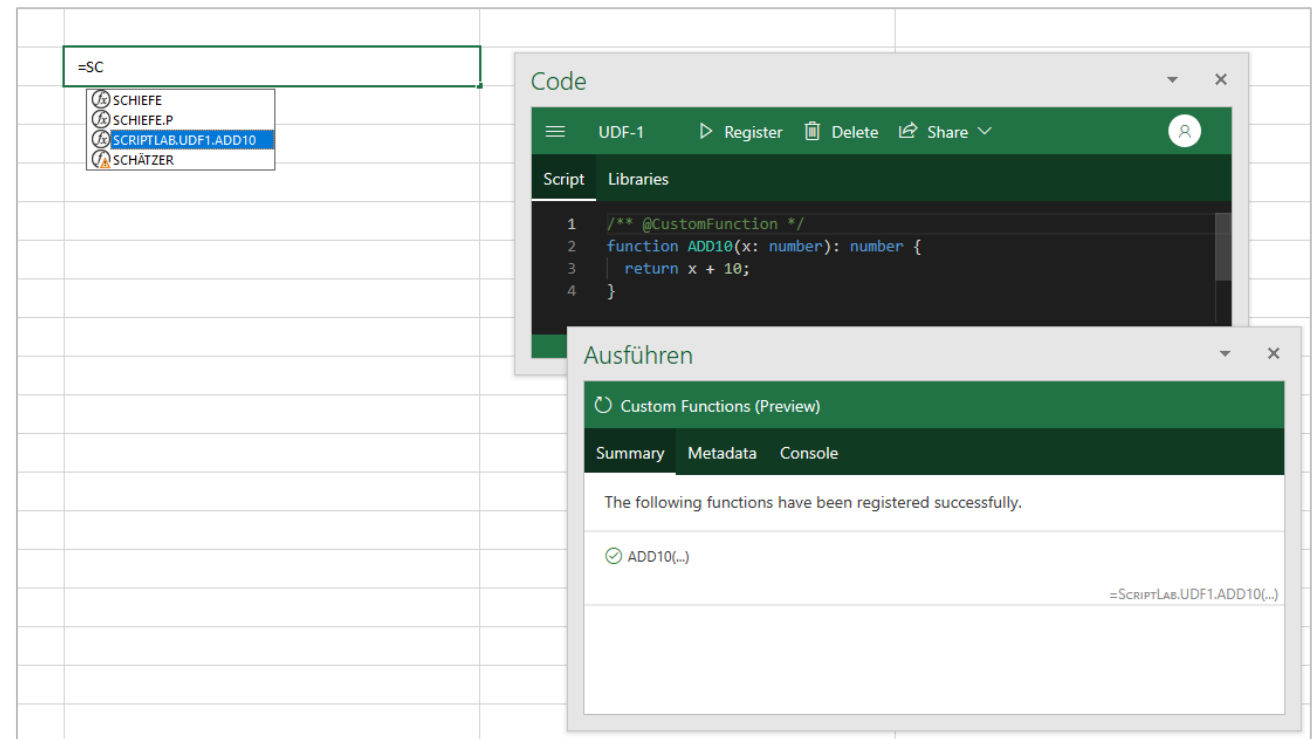




# Script Lab

## Funktionen

- Assistent
- Markierung
- Registrierung
- Metadaten
- Löschen



C:\Users\Benutzername\AppData\Local\Microsoft\Office\16.0\Wef\CustomFunctions



# Inhalt



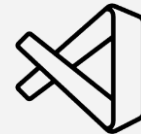
Überblick



Script Lab



Konzepte



**Visual Studio Code**



Objektmodell



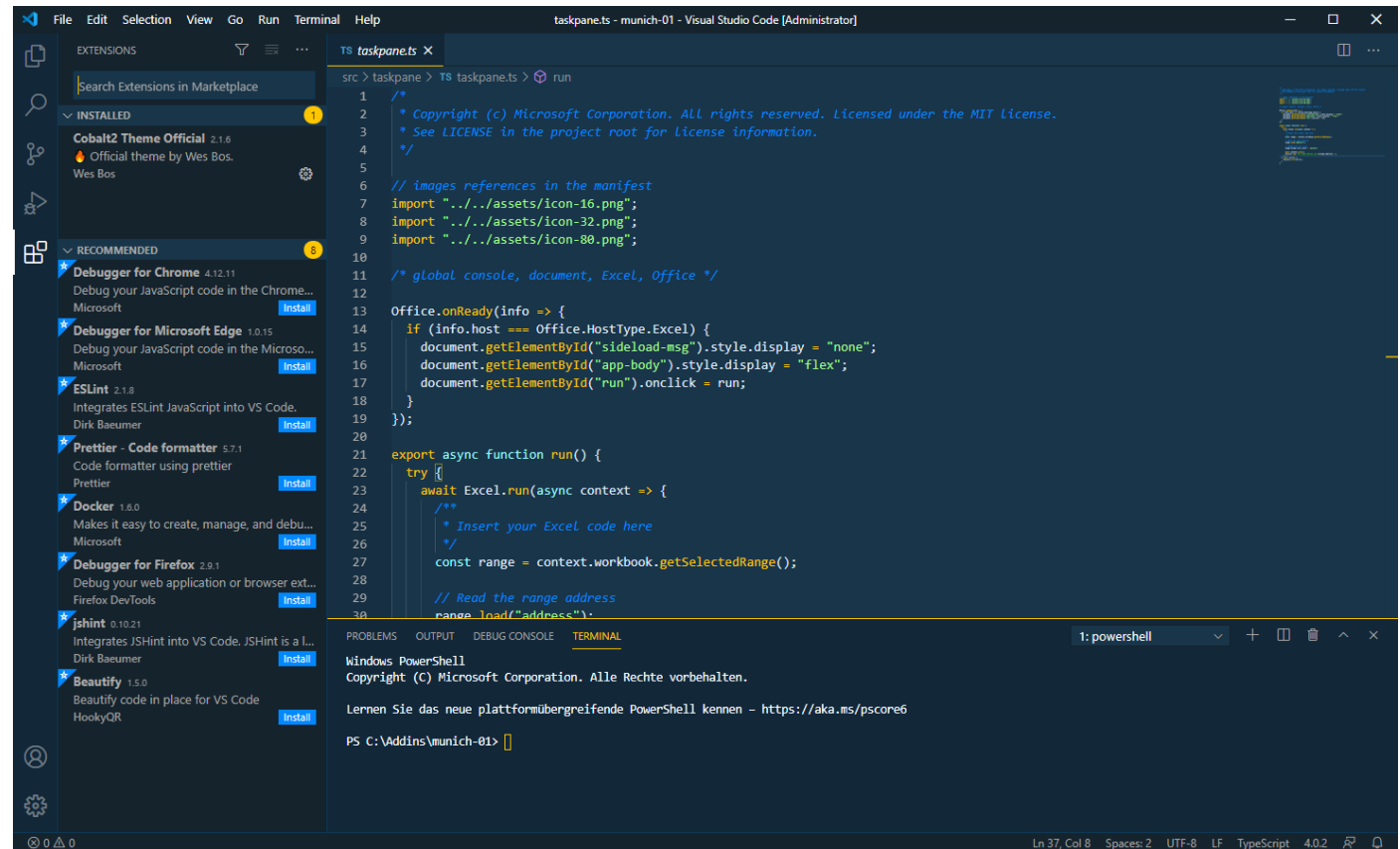
Office Scripts



# Visual Studio Code

## Überblick

- **Kostenlos**
- **Plattformübergreifend**
- **Sprachen** TS · CSS · PHP · Python ...
- **Plugins**
- **Themes**
- ...

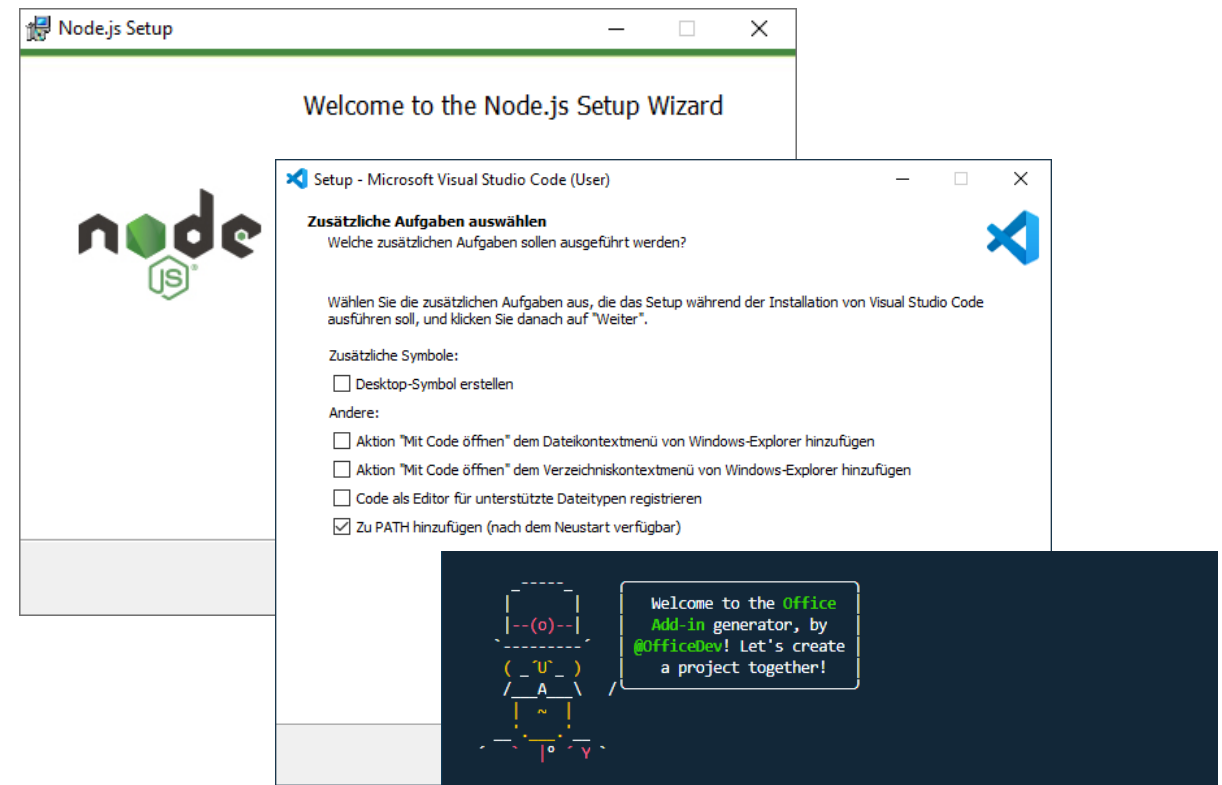




# Visual Studio Code

## VS Code & Office JS

- NodeJS
- Visual Studio Code
- Yeoman





# Visual Studio Code

## Kommandozeile

- Strg + Ö
- npm
- ...

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
1: powershell

PS C:\Users\Mourad> npm -h

Usage: npm <command>

where <command> is one of:
  access, adduser, audit, bin, bugs, c, cache, ci, cit,
  clean-install, clean-install-test, completion, config,
  create, ddp, dedupe, deprecate, dist-tag, docs, doctor,
  edit, explore, fund, get, help, help-search, hook, i, init,
  install, install-ci-test, install-test, it, link, list, ln,
  login, logout, ls, org, outdated, owner, pack, ping, prefix,
  profile, prune, publish, rb, rebuild, repo, restart, root,
  run, run-script, s, se, search, set, shrinkwrap, star,
  stars, start, stop, t, team, test, token, tst, un,
  uninstall, unpublish, unstar, up, update, v, version, view,
  whoami

npm <command> -h  quick help on <command>
npm -l           display full usage info
npm help <term>  search for help on <term>
npm help npm     involved overview

Specify configs in the ini-formatted file:
  C:\Users\Mourad\.npmrc
or on the command line via: npm <command> --key value
Config info can be viewed via: npm help config

npm@6.14.6 C:\Program Files\nodejs\node_modules\npm
PS C:\Users\Mourad>
```





# Visual Studio Code

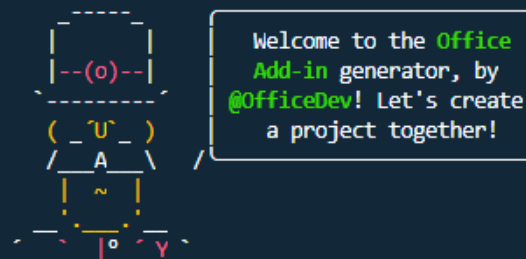
## Yeoman

- `npm install -g yo generator-office`
- `yo office`

```
PS C:\Users\Mourad> npm install -g yo generator-office
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated opn@6.0.0: The package has been renamed to `open`
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated cross-spawn-async@2.2.5: cross-spawn no longer requires a build toolchain, use it instead
[ ] - extract:mem-fs: Silly extract expand-brackets@2.1.4
```

<https://github.com/OfficeDev/generator-office>

```
PS C:\Users\Mourad> yo office
```



```
> Choose a project type: (Use arrow keys)
> Office Add-in Task Pane project
Office Add-in Task Pane project using Angular framework
Office Add-in Task Pane project using React framework
Office Add-in Task Pane project supporting single sign-on
Office Add-in project containing the manifest only
Excel Custom Functions Add-in project
```



# Visual Studio Code

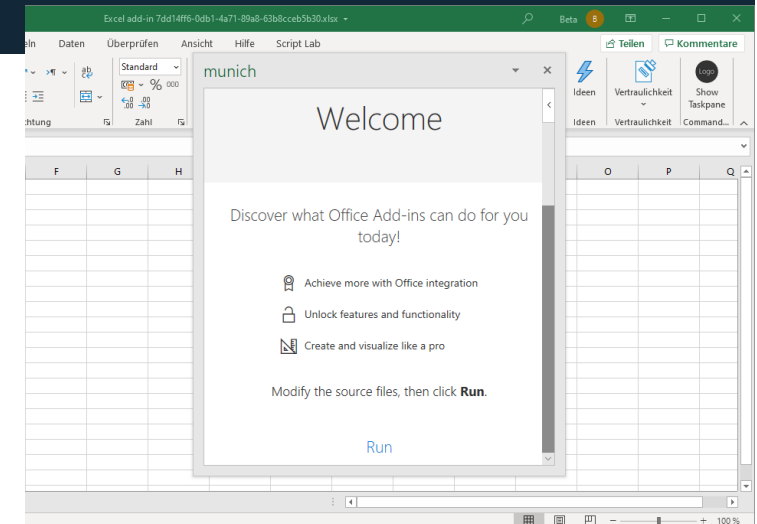
## Starten

- **npm start**
- **npm stop**
- **npm run dev-server**
- **npm run validate**
- ...

```
PS C:\Addins\munich> npm start

> office-addin-taskpane@0.0.1 start C:\Addins\munich
> office-addin-debugging start manifest.xml

Debugging is being started...
App type: desktop
Enabled debugging for add-in 7dd14ff6-0db1-4a71-89a8-63b8cceb5b30. Debug method: 0
Starting the dev server... (webpack-dev-server --mode development)
The dev server is running on port 3000. Process id: 4352
Sideload the Office Add-in...
Debugging started.
```

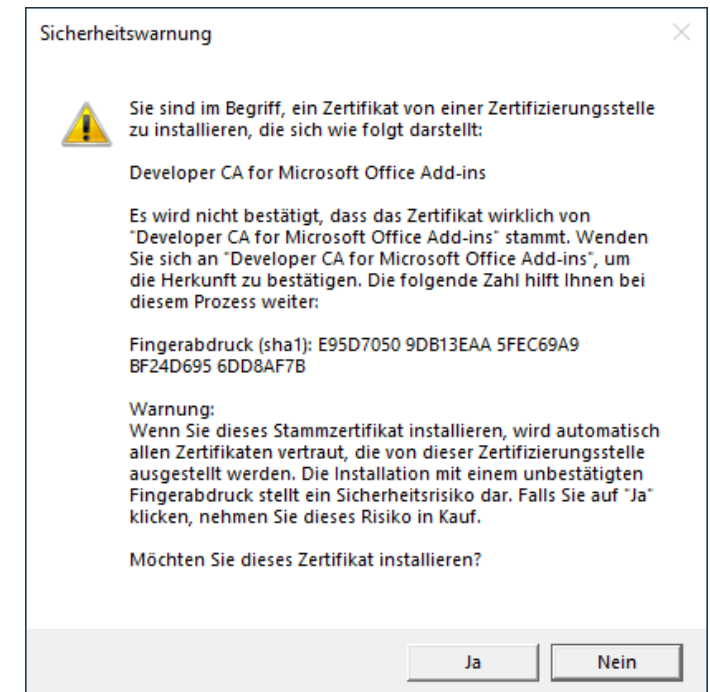




# Visual Studio Code

## Zertifikat

- `npx office-addin-dev-certs verify`
- `npx office-addin-dev-certs install`
- `npx office-addin-dev-certs uninstall`





# Inhalt



Überblick



Script Lab



Konzepte



Visual Studio Code



Objektmodell



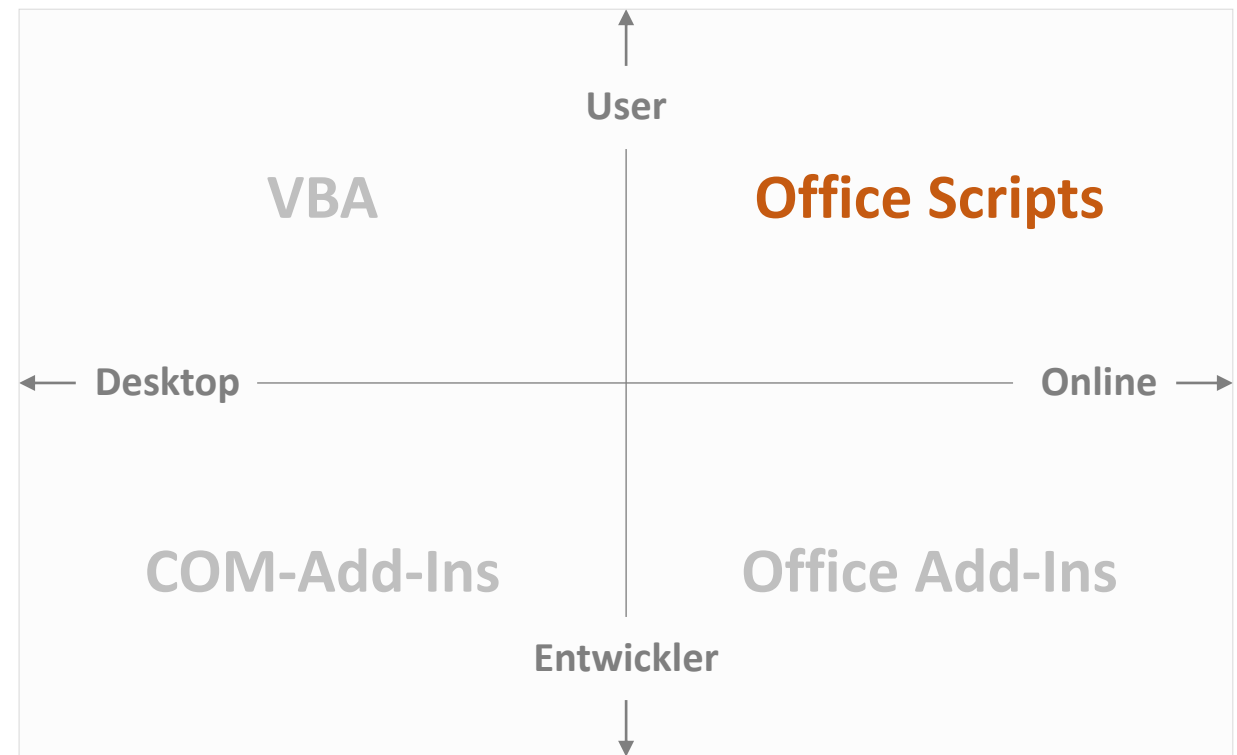
**Office Scripts**



# Office Scripts

## Überblick

- Automatisierung
- Preview
- Excel
- Spezielle API
- Plattform-Limitationen



<https://docs.microsoft.com/en-us/office/dev/scripts/overview/excel>



# Office Scripts

## Voraussetzungen

- **Microsoft 365**    ≡    **Business, ProPlus, E3, E5, A3, A5**
- **Aktivierung**    ≡    **Admin**
- **Excel**    ≡    **Online**





# Office Scripts

## Aktivierung

Microsoft 365 admin center

Suchen

Ressourcen

Abrechnung

Support

Einstellungen

Domänen

Microsoft Search

Einstellungen der Organisation

Add-Ins

Partnerbeziehungen

Setup

Berichte

Status

Admin Center

Security

Dienste

Sicherheit und Datenschutz

Organisationsprofil

Name ↑	Beschreibung
Office im Web	Benutzern das Öffnen von Dateien, die in einem Drittanbieter-Speicher...
Office Scripts	Benutzern das Automatisieren ihrer Aufgaben in Office im Web...
Office-Installationsoptionen	Wählen Sie aus, wie häufig Benutzer Funktions Updates und die...
SharePoint	Externes Teilen kontrollieren.
Sway	Freigabe- und Inhaltsquellen verwalten, die für Sway zugelassen sind.
Verzeichnissynchronisierung	Benutzer mithilfe von Azure Active Directory in die Cloud synchronisieren.
Whiteboard	Benutzern den Zugriff auf Microsoft Whiteboard sowie die Zusammenar...
Zustimmung von Benutzern zu Apps	Wählen Sie aus, ob Benutzer Apps den Zugriff auf die Daten Ihrer Organisat...

### Office Scripts

① Planen Sie bis zu 48 Stunden ein, bis Office Scripts-Einstellungsänderungen wirksam werden.

Office Scripts ermöglicht Benutzern, Ihre Aufgaben zu automatisieren, indem Sie Skripts in Excel im Web aufzeichnet, bearbeitet und ausführt. Office Scripts funktioniert mit Power Automate, und die Benutzer können Skripts für Arbeitsmappen unter Verwendung des Excel Online (Business)-Connectors ausführen. Sie sollten Ihre vorhandenen Richtlinien für die Verhinderung von Datenverlust überprüfen, damit die Daten bei Verwendung von Office Scripts geschützt bleiben.

[Weitere Informationen zu Office Scripts](#)  
[Weitere Informationen zur Auswirkungen auf vorhandene Richtlinien](#)

- ☒ Benutzern das Automatisieren ihrer Aufgaben in Excel im Web gestatten  
Aktivieren Sie Office Scripts für Ihre Organisation.  

☒ Jeder  
☐ Bestimmte Gruppe
- ☒ Zulassen, dass Benutzer mit Zugriff auf Office Scripts ihre Skripts für andere Personen in der Organisation freigeben  
Keine externe Freigabe zugelassen.  

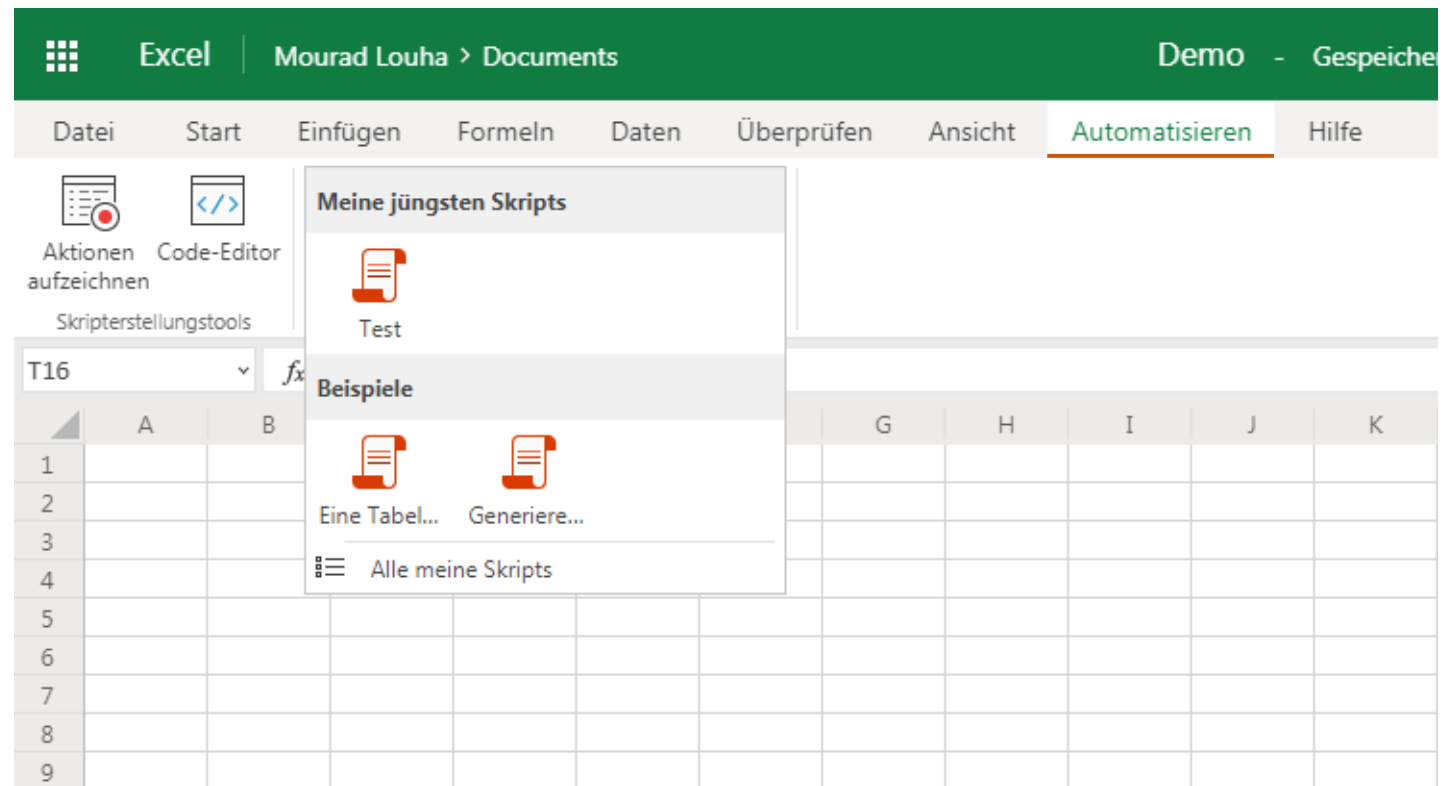
☒ Jeder  
☐ Bestimmte Gruppe



# Office Scripts

## Menüband

- Recorder
- Code
- Skripte
- Beispiele

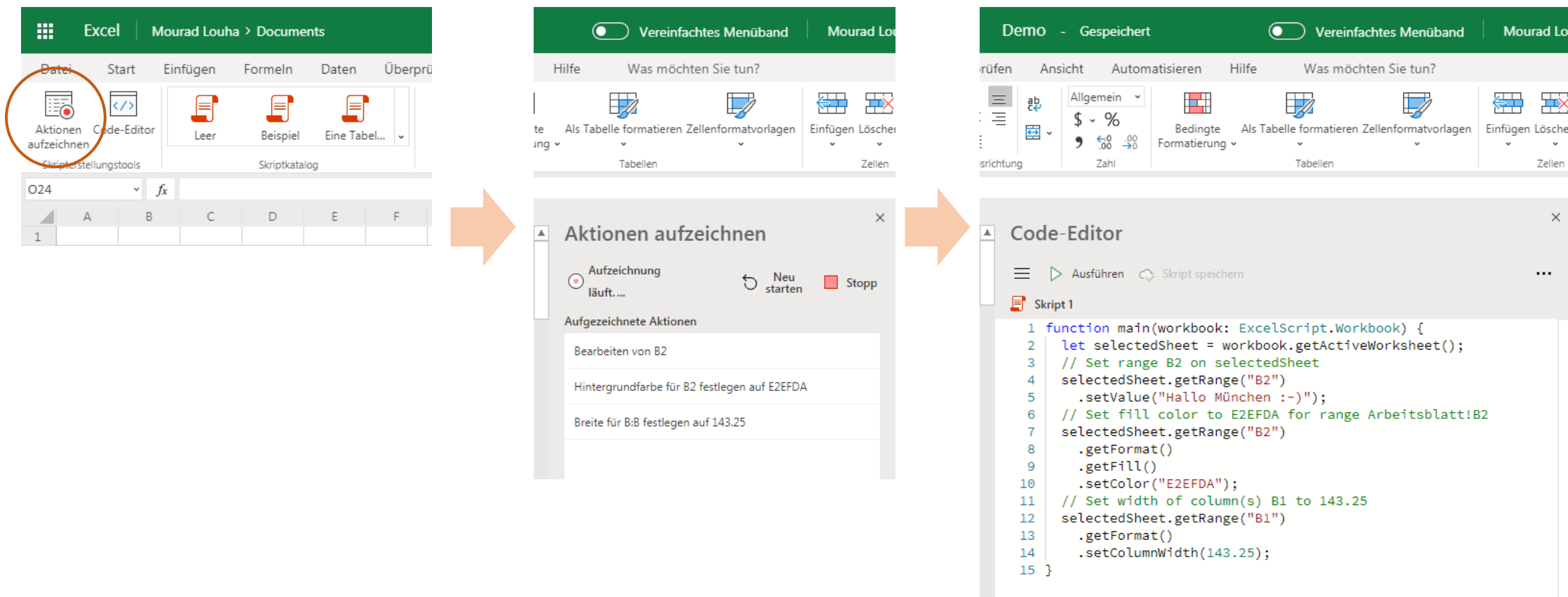






# Office Scripts

## Recorder





# Office Scripts

## Code

- Aufgabenbereich
- Editierfenster
- Syntax-Hervorhebung
- IntelliSense
- Optionen

The screenshot shows the 'Code-Editor' window with a sample script. The script is a function named 'main' that takes a 'workbook' parameter of type 'ExcelScript.Workbook'. It sets up a separate worksheet with sample data and creates a table. The code is as follows:

```
1  /**
2   * This sample script creates a sorted table.
3   * It also highlights a couple rows to demonstrate formatting.
4   * It does not affect any pre-existing data in your workbook.
5   *
6   * It is based off of this Office Scripts tutorial:
7   * https://docs.microsoft.com/office/dev/scripts/tutorials/
8   * excel-tutorial
9   */
10 function main(workbook: ExcelScript.Workbook) {
11     // This function sets up a separate worksheet with sample
12     // data.
13     let sampleSheet = setUpSample(workbook);
14
15     /*****
16      * Begin Create Table sample *
17      *****/
18
19     // Set fill color to FFC000 (the hex value for orange).
20     sampleSheet
21     .getRange("A2:C2")
22     .getRange()
23     .getPageLayout
24     .getPivotTable
25     .getPivotTables
26     .getPosition
27     .getPrevious
28     .getProtection
29     .getRange
30     .getRangeByIndexes
31     .getRanges
32     .getShape
33     .getShapes
34
35     // Create the table
36     let newTable = workbook.addTable(sampleSheet.getRange
37     ("A1:C5"), true);
38 }
```

IntelliSense suggestions are visible for the 'getRange' method call on line 20, showing various methods like 'getPageLayout', 'getPivotTable', 'getPivotTables', 'getPosition', 'getPrevious', 'getProtection', 'getRange', 'getRangeByIndexes', 'getRanges', 'getShape', and 'getShapes'.

The screenshot shows the 'Editor-Einstellungen' (Editor Settings) panel. It contains several settings that can be configured:

- Design:** A dropdown menu currently set to 'Dunkel' (Dark).
- Schriftgrad:** A dropdown menu currently set to '12'.
- Zellenbruch:** A dropdown menu currently set to 'Begrenzt' (Limited).
- Registerkartengröße:** A dropdown menu currently set to '4'.
- Minikarte:** A checkbox that is checked.
- Falten:** A checkbox that is checked.

At the bottom, there is a link: 'Weitere Informationen zu Office-Skripts' (More information about Office Scripts).



# Office Scripts

## API

- **Spezielle Version der Office JS API**

<https://docs.microsoft.com/en-us/javascript/api/office-scripts/overview?view=office-scripts>

- **Nur ein Einstiegspunkt**
- **Keine Common API**
- **Keine Ereignisse**

### Office Scripts

```
let mySheet = workbook.getActiveWorksheet();
let myRange = mySheet.getRange("B2");

console.log('Wert in B2 = ' + myRange.getValue());
```

### Office JS

```
let mySheet = context.workbook.worksheets.getActiveWorksheet();
let myRange = mySheet.getRange("B2");

myRange.load("values");
await context.sync();

console.log("Wert in B2 = " + myRange.values[0][0]);
```



# Office Scripts

## Einstiegspunkt

- **Main**
- **ExcelScript.Workbook**
- **Hilfsfunktionen möglich**

```
function main(workbook: ExcelScript.Workbook) {  
    let x = myTextLeft();  
    let y = myTextRight();  
  
    console.log(x + ' ' + y);  
}  
  
function myTextLeft() {  
    return "Hallo";  
}  
  
function myTextRight() {  
    return "München";  
}
```



# Office Scripts

## Dateien

- **OneDrive**
- **Textdateien**
- **Metadaten**

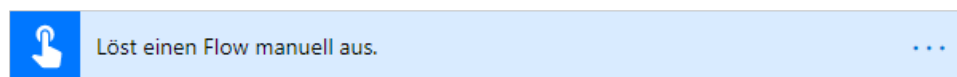
The screenshot displays the Office Scripts interface. On the left, a sidebar shows the user 'Mourad Louha' and a list of folders: 'Eigene Dateien', 'Zuletzt verwendet', 'Geteilt', 'Papierkorb', and 'Geteilte Bibliotheken'. The main area shows a file list under the path 'Dateien > Dokumente > Office-Skripts'. The list contains two files: 'Beispiel.osts' and 'Leer.osts'. A callout box highlights the 'Beispiel.osts' file, showing its metadata in JSON format:

```
{
  "version"      : "0.2.0",
  "body"         : "function main(workbook: ExcelScript.Workbook) {}",
  "description"  : "",
  "parameterInfo": "{
    \"originalParameterOrder\" : [],
    \"parameterSchema\"       : {
      \"type\"          : \"object\",
      \"default\"       : {},
      \"x-ms-visibility\" : \"internal\"
    },
    \"returnSchema\"    : {
      \"type\"          : \"object\",
      \"properties\"    : {}
    }
  }",
  "apiInfo"      : "{
    \"variant\": \"synchronous\"
  }"
}
```



# Office Scripts

## Power Automate



**Skript ausführen (Vorschau)**

\* Speicherort: OneDrive for Business

\* Dokumentbibliothek: OneDrive

\* Datei: /Demo.xlsx

\* Skript: Automate-Timestamp

	A	B	C	D
1	Datum	Uhrzeit		
2	26.09.2020	10:27:37		
3	26.09.2020	10:28:33		
4				
5				
6				



```
function main(workbook: ExcelScript.Workbook) {  
  
    let mySheet = workbook.getWorksheet("Power Automate");  
    let myRow = mySheet.getRange().getColumn(0)  
        .getUsedRange().getLastRow();  
  
    // Zellen  
  
    let myDateCell = myRow.getOffsetRange(1, 0);  
    let myTimeCell = myRow.getOffsetRange(1, 1);  
  
    // Datum...  
  
    let myTimestamp = new Date(Date.now());  
  
    // Schreiben  
  
    myDateCell.setValue(myTimestamp.toLocaleDateString());  
    myTimeCell.setValue(myTimestamp.toLocaleTimeString());  
}
```





# Zusammenfassung

- » Überblick
  - » Konzepte
  - » Objektmodell
  - » Script Lab
  - » Visual Studio Code
  - » Office Scripts
- ≡ Office Add-Ins sind Websites
  - ≡ Umdenken für VBA-Entwickler
  - ≡ An VBA angelehnt
  - ≡ Das Tool für den Einstieg
  - ≡ Professioneller Add-Ins erstellen
  - ≡ Vereinfachung der JS-API



# Kontakt

## Mourad Louha

E-Mail	:	vba@maninweb.de	
LinkedIn	:	linkedin.com/in/maninweb	
Twitter	:	twitter.com/maninweb	
Facebook	:	facebook.com/maninweb	





# Ich habe fertig mit Vortrag :-)

... und nun ...

## ... Action ...

