

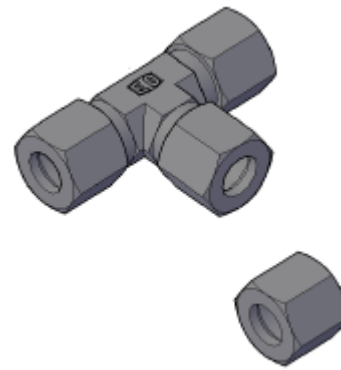
AutoCAD Plant 3D:

Erweiterungen programmieren

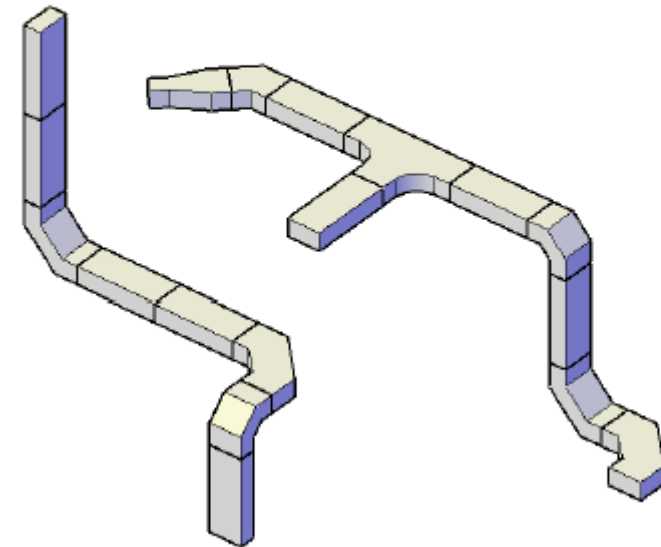
Python-Skripte -
Variantenprogrammierung

API –
Application Programming
Interface

APS –
Autodesk Plattform Services



EO T-Stutzen T08LL71



Python-Skripte

Definition

Python-Skripte können in AutoCAD Plant 3D verwendet werden, um die Variantenkonstruktion zu automatisieren und zu optimieren. Hier sind einige Schritte und Möglichkeiten, wie du Python-Skripte einsetzen kannst:

1. Erstellung von benutzerdefinierten Komponenten: Du kannst Python-Skripte schreiben, um benutzerdefinierte Komponenten zu erstellen und diese in deinen Katalogen zu speichern. Dies ermöglicht eine schnelle Anpassung und Wiederverwendung von Komponenten.

2. Automatisierung von Aufgaben: Python kann verwendet werden, um wiederkehrende Aufgaben zu automatisieren, wie z.B. das Platzieren von Rohrleitungen oder das Anpassen von Bauteilen. Dies spart Zeit und reduziert Fehler.

3. Integration mit anderen Systemen: Python-Skripte können dazu verwendet werden, AutoCAD Plant 3D mit anderen Softwarelösungen zu integrieren, um Daten auszutauschen und Arbeitsabläufe zu optimieren.

4. Erweiterung der Funktionalität: Mit Python kannst du die Funktionalität von AutoCAD Plant 3D erweitern, indem du neue Befehle und Werkzeuge erstellst, die speziell auf deine Bedürfnisse zugeschnitten sind.

Python-Skripte

Beispiel allgemein

```
import acad
from acad import Application

# Verbindung zu AutoCAD Plant 3D herstellen
app = Application()

# Neue Komponente erstellen
component =
app.create_component("CustomComponent")

# Parameter der Komponente festlegen
component.set_parameter("Height", 100)
component.set_parameter("Width", 50)

# Komponente in den Katalog einfügen
app.catalog.add_component(component)
```

Python-Skripte

Beispiel:

rechteckiges Becken

```
import acad from acad import Application
# Verbindung zu AutoCAD Plant 3D herstellen
app = Application()

# Funktion zur Erstellung eines rechteckigen Beckens
def create_rectangular_tank(length, width, height, wall_thickness):
    # Neue Komponente erstellen
    tank = app.create_component("RectangularTank")

    # Parameter der Komponente festlegen
    tank.set_parameter("Length", length) tank.set_parameter("Width", width)
    tank.set_parameter("Height", height)
    tank.set_parameter("WallThickness", wall_thickness)

    # Komponente in den Katalog einfügen
    app.catalog.add_component(tank)
    return tank

# Beispielwerte für die Parameter
length = 5000 # Länge in mm
width = 3000 # Breite in mm
height = 2000 # Höhe in mm
wall_thickness = 100 # Wandstärke in mm

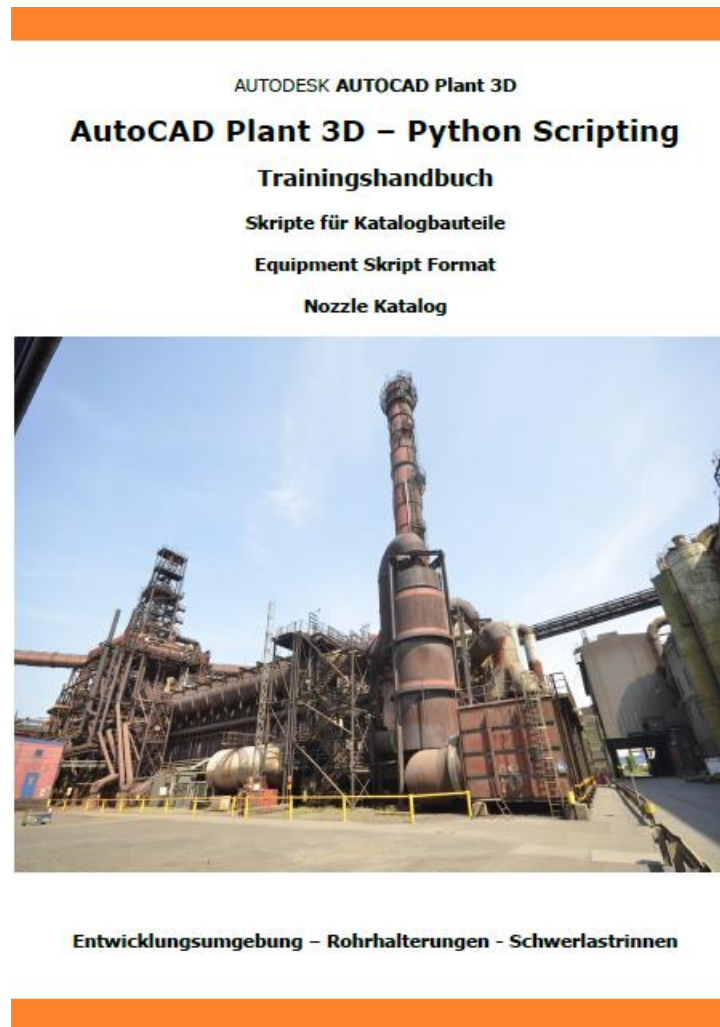
# Rechteckiges Becken erstellen
tank = create_rectangular_tank(length, width, height, wall_thickness)

print(f"Rechteckiges Becken erstellt mit Länge: {length} mm, Breite: {width} mm,
Höhe: {height} mm, Wandstärke: {wall_thickness} mm")
```

Python-Skripte

Variantenprogrammierung

Handbuch



Workshop Plant 3D	Python Scripting für Plant 3D
Benutzerdefinierte Skripte in Plant 3D	
Einführung.....	1
Einfaches Skript für Katalogbauteile.....	3
Skript registrieren	
Skript testen	
Vorschäubilder erstellen	
Anwendung im Spec – Editor	
Die Funktion setpoint.....	6
Skript: PHCT_PIPE_REC_01.py	
Die Funktion translate und subtractForm.....	8
Skript: PHCT_PIPE_UAA_01.py	
Bearbeitungsmodus aktivieren	
Systembauteile verwenden.....	12
Skript: nozflage.py	
Python Skript für Nozzle Katalog.....	14
Plant 3D Anbindung.....	16
TESTASCRIP / TESTASCRIP1	
Struktur der Skripte.....	18
Equipment Skript Format.....	21
Skript: SimpleVesselSkirt.py	
EquipmentType.xml	
Vorschäubilder (200x200 Pixel)	
Kategorien (Categories)	
Kombination Aktuator – Ventil.....	32
Entwicklungsumgebung.....	33
Python Decompiler	
Texteditor	
XML – Editor	
Bildbearbeitungsprogramme	
Dateien packen und entpacken	
GUID – Generator	
Python	
Virtuelle Maschine	
Python Werkzeugpalette in Plant 3D	

Python-Skripte

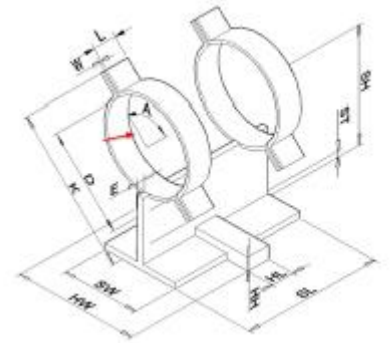
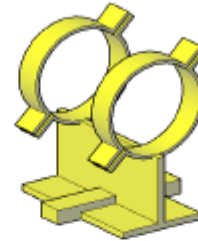
Variantenprogrammierung

Handbuch

Beispiel Rohrhalterung

Rohrhalterungen – gedrehte Schelle

In unserer Anlage verwenden wir Halterungen mit geneigter Schelle. Die Drehung der Schelle ist bei den Systemhalterungen nicht vorgesehen.



Skript: CSGC006.py

Zuerst erstellen wir ein Skript mit gedrehter Schelle:

```
import aqs.math
import math
from varmain.primitive import *
from varmain.var_basic import *
from varmain.custom import *

@activate(Group="Support", Ports="1", TooltipShort="Rohrschelle 45 Grad verdreht",
          TooltipLong="Rohrschelle 45 Grad verdreht", LengthUnit="mm")
@group("MainDimensions")
@param(D=LENGTH, TooltipShort="Inside diameter of pipe support.",
        TooltipLong="Inside diameter of pipe support. (Matches the Pipe diameter)")
@param(L=LENGTH, TooltipShort="Length of the clamp", TooltipLong="Length of the clamp")
@param(W=LENGTH, TooltipShort="Thickness of the clamp", TooltipLong="Thickness of the clamp")
@param(K=LENGTH, TooltipShort="Overall width of the clamp.", TooltipLong="Overall width of the clamp. If set to 0 we use D+(L*2)")
@param(A=ANGLE, TooltipShort="rotation of the clamp", TooltipLong="rotation of the clamp")
@group(Name="meaningless enum")
@enum(1, "align X")
@enum(2, "align Y")
@enum(3, "align Z")

def CSGC006(s, D = 116.0, L = 30.0, W = 4.0, K = 0.0, A = 45.0, ID = 'CSGC006',
**kw):
    R1 = D / 2.0
    R2 = R1 + W
    if K <= 0.0:
        K = D + L * 2.0
```

Python-Skripte

Varianten

Autodesk

Appstore

AUTODESK App Store

	Python for Tri-clamp and coupling Betriebssystem: Win64 This catalog includes the 3 most common types of clamps: 2-segments, 3-segments,...	★★★★★ 3	USD 25,00
	PVC-U Valves Fittings Catalog Betriebssystem: Win64 This package includes pipe, fittings and valves used in Water treatment, Marine, an...	★★★★★ 3	USD 79,00
	ISO 2037 fittings valves Python catalog Betriebssystem: Win64 This catalog includes all common fittings, instruments, and valves used in Food and...	★★★★★ 4	USD 79,00
	ISO 1127 fittings valves Python catalog Betriebssystem: Win64 This catalog includes all common fittings, instruments, and valves used in Food and...	☆☆☆☆☆ 0	USD 79,00
	ISCO HDPE Fittings Valves Python Content Pack Betriebssystem: Win64 This package is a fully Python script and includes pipes, valves, and connectors....	☆☆☆☆☆ 0	USD 149,00
	Instruments Python Catalog Betriebssystem: Win64 This package includes Fittings and Instruments adapted from Endress Hauser...	★★★★★ 6	USD 149,00
	GF Python Valves Fittings Aquasystem PPR Betriebssystem: Win64 All items are parametric Python parts. Fittings, valves, and instruments mostly adap...	☆☆☆☆☆ 0	USD 149,00

API

Application Programming
Interface

Autodesk bietet verschiedene APIs an, darunter:

- **ObjectARX**: Eine C++-basierte API für die Erstellung von benutzerdefinierten Befehlen und die Manipulation von Entitäten.
- **AutoLISP**: Eine spezifische Programmiersprache für AutoCAD zur Automatisierung von Aufgaben und Erstellung benutzerdefinierter Befehle.
- **Managed .NET API**: Ermöglicht die Programmierung in .NET-Sprachen wie C# zur Manipulation von Zeichnungsdateien und Integration mit anderen Windows-Anwendungen.

AutoLisp

DOSLib

DOSLib ist eine Bibliothek von LISP-fähigen Funktionen, die zusätzliche Funktionalitäten für CAD-Anwendungen wie AutoCAD, BricsCAD und ZWCAD bereitstellt

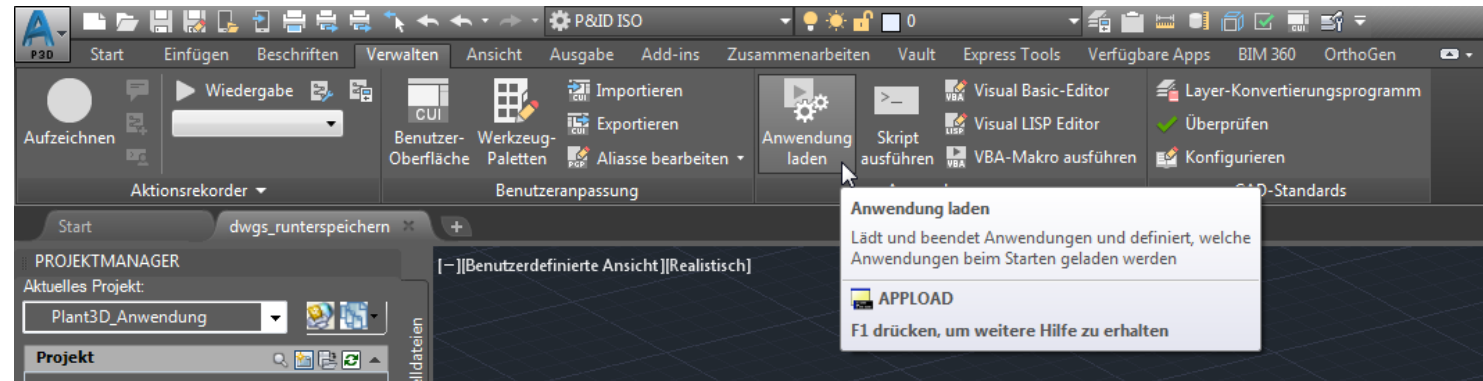
DOSLib kann man von [GitHub](#) herunterladen

The following tools are required to build DOSLib:

- [Microsoft Visual Studio 2022](#)
- [ObjectARX](#) SDK for AutoCAD

AutoLisp

Anwenden

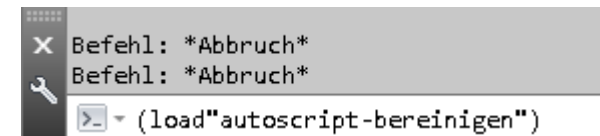


 Autoscript-bereinigen.lsp
Typ: AutoLISP-Anwendungsquelle

 Autoscript-erzeugen.lsp
Typ: AutoLISP-Anwendungsquelle

 Autoscript-export.lsp
Typ: AutoLISP-Anwendungsquelle

 Autoscript-plot.lsp
Typ: AutoLISP-Anwendungsquelle



AutoLisp

Beispiel

Automatisieren von Export nach AutoCAD

```
(defun script_erzeugen(uebergabe)

;#####
;Achtung, DOSLIB muss geladen sein!
;#####

; Beispielaufruf
; -----
; (setq uebergabe '())
; (setq uebergabe(cons "DWG" uebergabe))
; (setq uebergabe(cons "C:\\Bloecke\\" uebergabe))
; (setq uebergabe(cons "DWG(*.dwg):|*.dwg" uebergabe))
; (setq uebergabe(cons "c:\\temp\\" uebergabe))
; (setq uebergabe(cons (strcat "(load" "\" "punktkill" "\" " ")") uebergabe))
; (setq uebergabe(reverse uebergabe))
; (script_erzeugen uebergabe)

(mapcar 'set '(ueberschrift suchpfad filter scriptpfad befehl) uebergabe)

(setq dateiliste(dos_getfilem ueberschrift suchpfad filter))
(setq pfad(nth 0 dateiliste))
(setq dateiname_liste(member (nth 1 dateiliste) dateiliste))

(if (not scriptpfad)
    (setq scriptpfad pfad))

(setq scriptname(dos_getfilenav "Script-Datei" scriptpfad "scr" 1))
(setq scriptdatei(open scriptname "w"))
(foreach dateiname dateiname_liste

    (setq zeile(strcat ".open " "\"pfad dateiname \""))
    (write-line zeile scriptdatei)

    (write-line befehl scriptdatei)

    (write-line ".qsave" scriptdatei)
    (write-line ".close" scriptdatei)

);Ende foreach
(close scriptdatei)
);Ende defun

;-----
(ARXLOAD "DOSLib22x64.arx")

(setq uebergabe '())
(setq uebergabe(cons "DWG" uebergabe))
(setq uebergabe(cons "S:\\PID DWG\\" uebergabe))
(setq uebergabe(cons "DWG(*.dwg):|*.dwg" uebergabe))
(setq uebergabe(cons nil uebergabe))
(setq uebergabe(cons (strcat "(load" "\" "export" "\" " ")") uebergabe))
(setq uebergabe(reverse uebergabe))
(script_erzeugen uebergabe)
```



Autoscript-export.lsp

Typ: AutoLISP-Anwendungsquelle



CONTELOS-SEITZ
CONSULTING GMBH

APS

Autodesk Platform Services

APS steht für **Autodesk Platform Services** (früher bekannt als Forge). APS bietet eine Reihe von APIs und Diensten, die Entwicklern helfen, maßgeschneiderte Geschäftslösungen zu erstellen und die Funktionalität von Autodesk-Produkten zu erweitern

Mit APS kannst du:

- **2D- und 3D-Modelle visualisieren:** Modelle aus Anwendungen wie AutoCAD, Fusion 360 und Revit im Web anzeigen.
- **Automatisierungsskripte in der Cloud ausführen:** Aufgaben, die du auf deinem Desktop automatisieren kannst, lassen sich nun auch in großem Maßstab in der Cloud automatisieren.
- **Apps für BIM erstellen:** Die BIM 360 APIs ermöglichen die Integration und Erweiterung der Funktionen der Autodesk BIM 360 Plattform.

APS bietet Entwicklern die Werkzeuge, um Daten in Dashboards zu verwandeln, digitale Zwillinge zu erstellen, Workflows anzupassen und vieles mehr.

Ende

Tak for din interesse

Danke für Ihr Interesse

