AUTODESK UNIVERSITY

El Poder de la Automatización: Python en Dynamo para Revit y Civil 3D

Raquel Bascones Recio

Senior Implementation Consultant, Autodesk

David Licona

Implementation Consultant, Autodesk

Ponentes

Raquel Bascones Recio
Senior Implementation Consultant

2010 – 2017: Architect & Landscape Architect
2017 – 2019: Autodesk Global Product Support
Desde 2019: Autodesk Consulting
raquel.bascones.recio@autodesk.com
LinkedIn

David Licona
Implementation Consultant

2014 – 2017: Project Engineer – Construction
2017 – 2020: Project Manager – Rail infrastructure
Desde 2020: Autodesk Consulting
david.licona@autodesk.com
LinkedIn

Objetivos de aprendizaje

- Descubrir la programación orientada a objetos
- Aprender las bases y elementos de lenguaje de Python
- Diferenciar entre Python, IronPython y Cpython
- Comprender las bases de las API de Revit, AutoCAD y Civil 3D

Python en Dynamo

¿Por qué Python?

Python en Dynamo

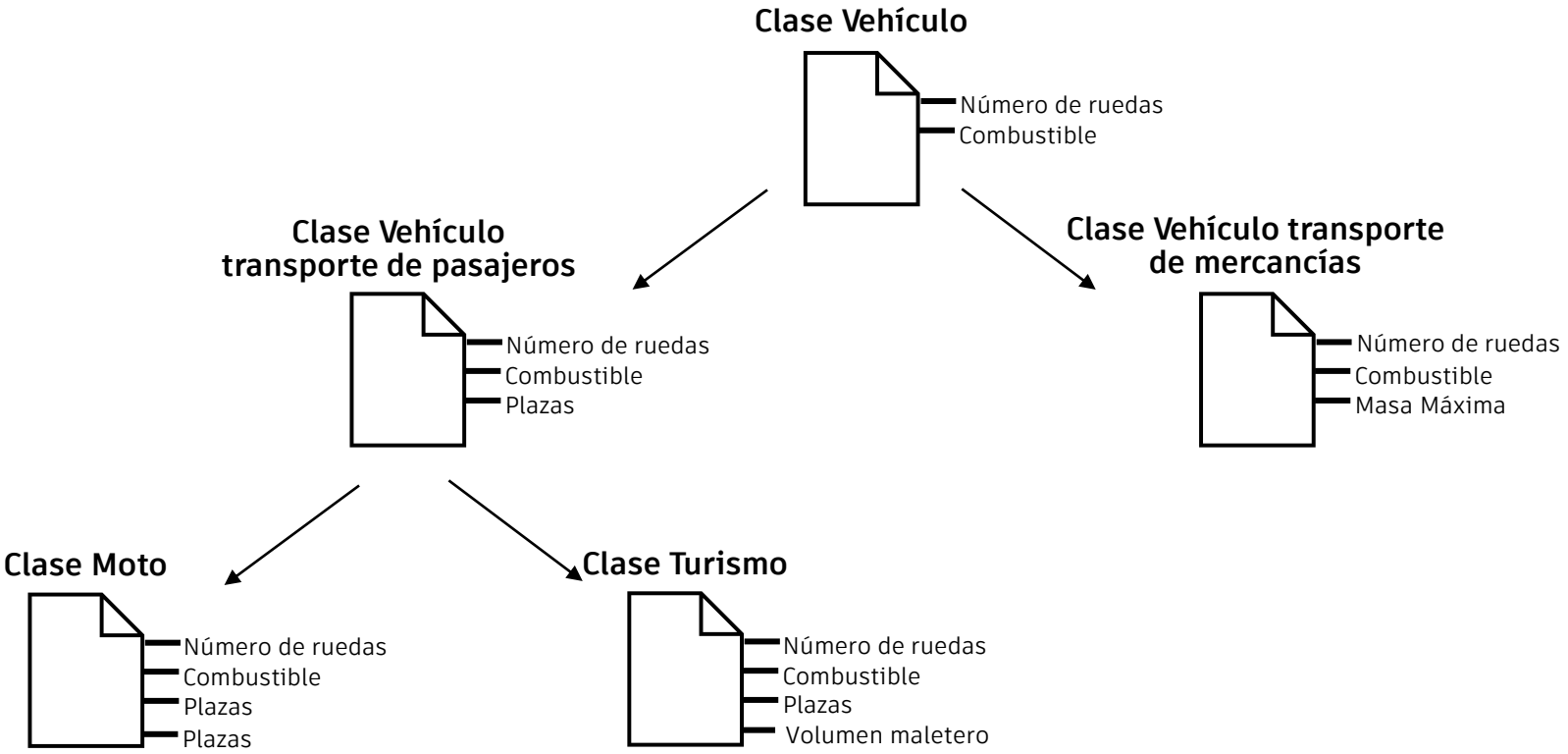
Conceptos clave

- Uno de los lenguajes de programación más usados en la actualidad
- Creado en 1980s por Guido van Rossum

Ventajas	Inconvenientes
Fácil de entender	No diseñado para ser *compiled*
Gran comunidad de usuarios y módulos	Más lento que otros lenguajes
Flexible	*Debugging* es lento y tedioso

Python en Dynamo

Programación orientada a objetos



Python en Dynamo

Implementaciones

IronPython

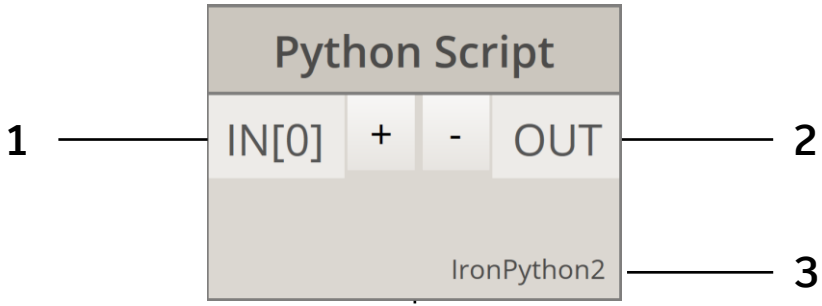
- Focalizado en .NET
- Originalmente creado por Microsoft, actualmente mantenido por voluntarios
- Utilizado por Dynamo desde sus inicios
- Usa Python 2.7, actualmente sin soporte

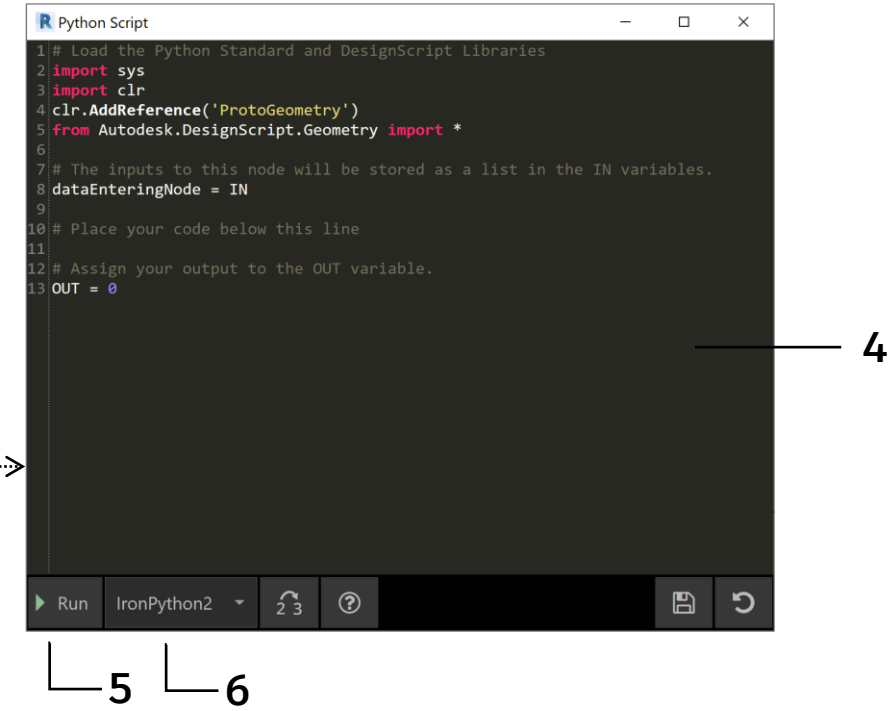
CPython

- Implementación oficial de Python
- Presente en Dynamo desde la version 2.7
- Actualizado a las últimas versiones de Python
- Permite importar módulos y librerías como NumPy, Panda, etc.

Python en Dynamo

Nodo

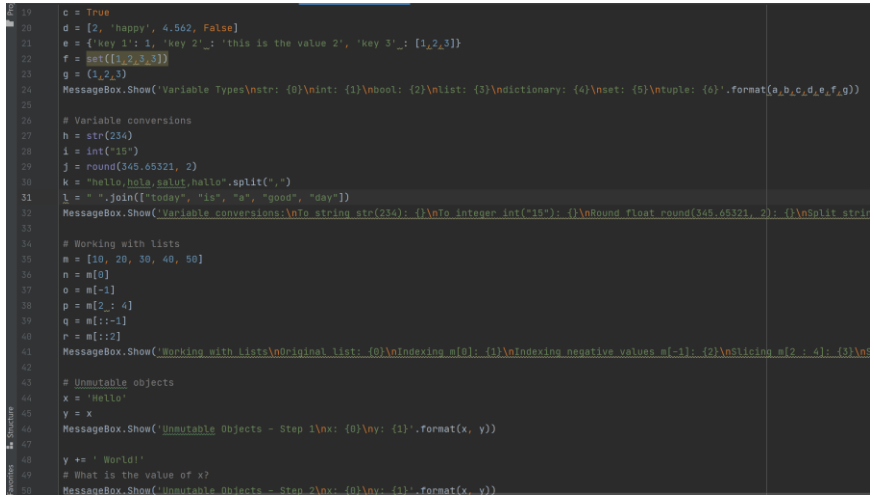




Python en Dynamo

Uso de un editor externo

- Ventajas
  - Autocompletado de funciones nativas de Python
  - Revisión de sintaxis
  - Colapsado de bloques
  - Reconocimiento de variables
- Se puede copiar y pegar al editor de Dynamo o ejecutar directamente el archivo py en Dynamo
- Multitud de editores gratuitos y de código abierto



Python en Dynamo

Cheat Sheet

Indexing lists

Working with values from lists

Value at index	*x = ["a", "b", "c"]* *x[0] = "a"*
Index counting from end	*x = ["a", "b", "c"]* *x[-1] = "c"*
Remove value	***del*** *x[2]*
Modify value	*x[1] = 25*
Length of a list	*x = ["a", "b", "c"]* ***len****(x) = 3*
Slicing lists	*list[start slice : end slice : step]* *x = [1, 2, 3, 4, 5]* *x[2::] = [3, 4, 5]* *x[1:3] = [2, 3, 4]* *x[::-1] = [5, 4, 3, 2, 1]* *x[::2] = [1, 3, 5]*

Assignment of variables

- Binding of a name with a value

Unique assignment	*x = 1*
Multiple assignments	*x, y, z = 1, 2, 3*
Assignment to the same value	*x = y = z = 0*
Increment	*x+=1 equals x = x + 1*
Decrement	*x-=1 equals x = x - 1*
Remove variable	***del*** *x*
Empty variables	*x = None* *x = ""* *x = []* *x = {}* *x =* ***set****()*

Variable conversions

- Change variable type

To integer	***int****(x)*
To float	***float****(x)*
Round float	***round****(number,#digits)*
To str	***str****(x)*
List from string	***list****("abc") > ["a", "b", "c"]* *x.****split****(separator)*
String from list	*separator.****join****(list)*
Check type	***isinstance****(x, type)*

Main operations on lists

Length of the list	***len****(x)*
Minimum / maximum value	***min****(x)* ***max****(x)*
Iterator returning (index,value)	***enumerate****(x)*
Iterator returning values at same index of all collections	***zip****(x, y, z)*
Evaluate existence/not existence in list, returns bool	*a* ***in*** *x* *a* ***not in*** *x*
Index of an item in a list	*x.****index****(a)*
Count occurrences of an item in a list	*x.****count****(a)*

Append a value at the end	*x.****append****(a)*
Add a sequence of values at the end	*x.****extend****([a,b, c])*
Insert item at index	*x.****insert****(index, a)*
Remove first item of value a	*x.****remove****(a)*
Remove item at index and return item	*x.****pop****(index)*
Sort the list in place	*x.****sort****()*
Make a sorted copy of the list	***sorted****(x)*

Lambda Forms

Anonymous functions

- Examples of use

names = ["Jo", "Mia", "Peter", "Aria", "Julian"]	
sorted*(names, key =* ***lambda*** *x:* ***len****(x))*	["Julian", "Peter", "Aria", "Mia", "Jo"]
filter*(****lambda*** *x:* ***len****(x)>3, names)*	["Peter", "Aria", "Julian"]
map*(****lambda*** *x: x + " Parker", names)*	["Jo Parker", "Mia Parker", "Peter Parker", "Aria Parker", "Julian Parker"]

Math

- Import math module

Operators	*+* *-* *** */* ***%*** *(remainder of division)* *//* *(integer value of division)* ***(pow, a^b)*
Absolute value	***abs****(x)*
Round up to integer	***math.ceil****(x)*
Round down to integer	***math.floor****(x)*

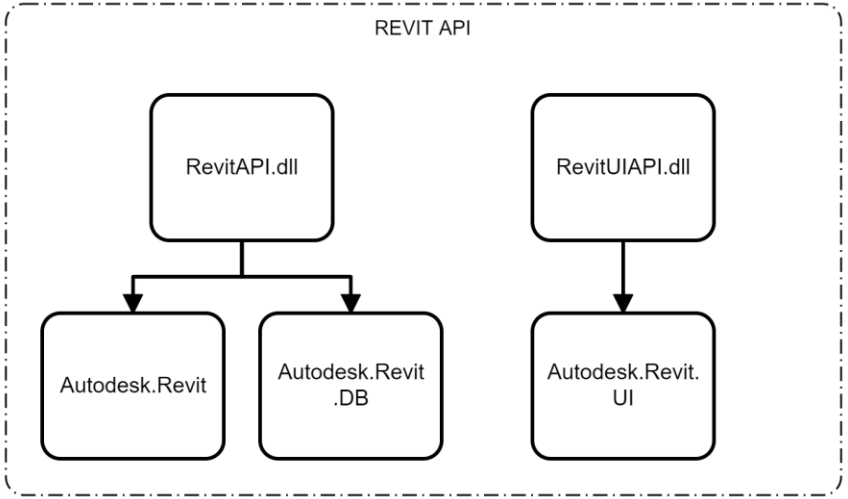
Python y Revit API

¿Qué es Revit API?

Python & Revit API

Namespaces

- Revit API se compone de dos *assemblies* (archivos .dll)
  - RevitUIAPI.dll
  - RevitAPI.dll
- Agrupan objectos e identificadores según funcionalidades
- Evita colisiones de nombres



Python & Revit API

Namespaces

- Añadir el *assembly* usando la libreria *Common Language Runtime (clr)*
  - *clr.AddReference("RevitAPI")*
  - *clr.AddReference("RevitUIAPI")*

- Importar los objectos requeridos
  - Importar todo el *namespace*
    - *from Autodesk.Revit.DB import **
    - *import Autodesk.Revit.DB*
  - Importar objectos determinados
    - *from Autodesk.Revit.DB import Wall*

```
32  # Import statements
33  import clr
34  import sys
35  # In IronPython it is possible to import references to .NET assemblies like those used to access the Revit API
36  clr.AddReference('RevitAPI')
37  clr.AddReference('RevitAPIUI')
38  clr.AddReference('RevitServices')  # this is a Dynamo for Revit assembly
```

Python & Revit API

Documentos

Document

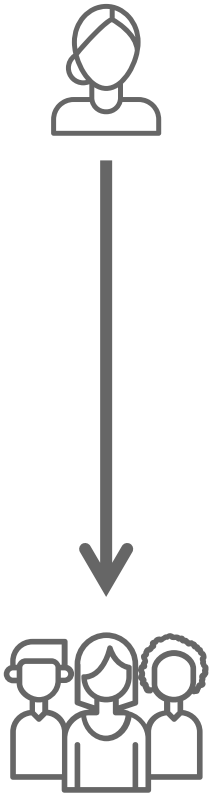
- Archivo Revit (RVT, RFA)
- Tiene propiedades que definen el archive (Title, PathName, etc)
- Métodos para modificar y crear elementos

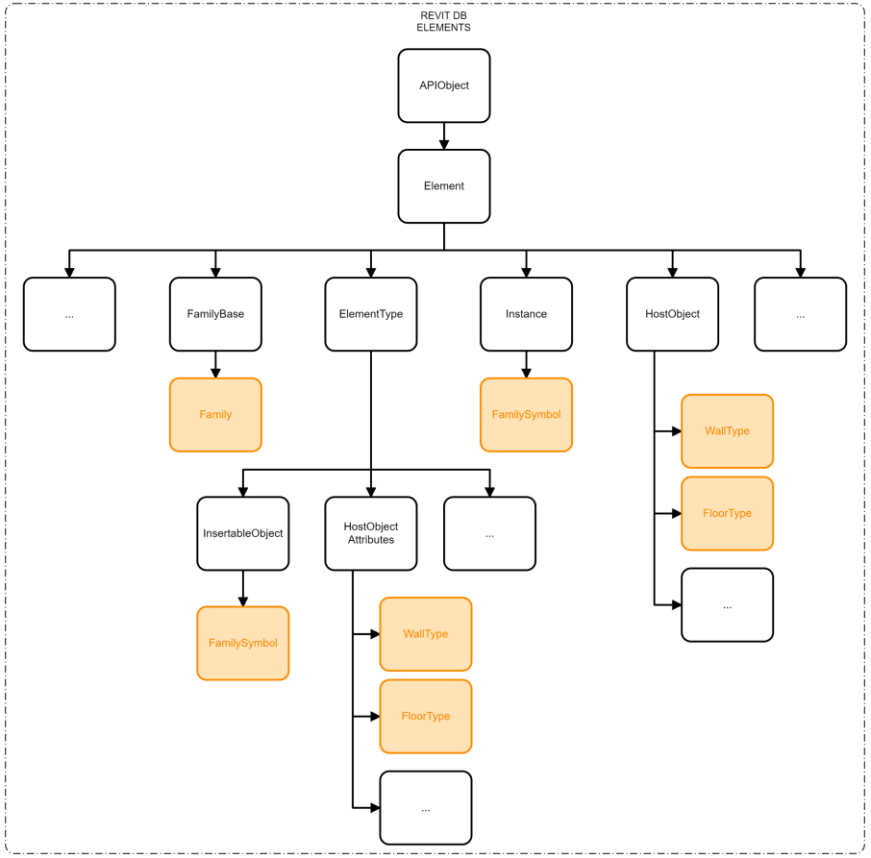
UIDocument

- Proyecto activo presentado al usuario
- Métodos para seleccionar elementos interactivamente
- Permite enviar mensajes al usuario mediante TaskDialog

Python & Revit API

DB Elements





Python & Revit API

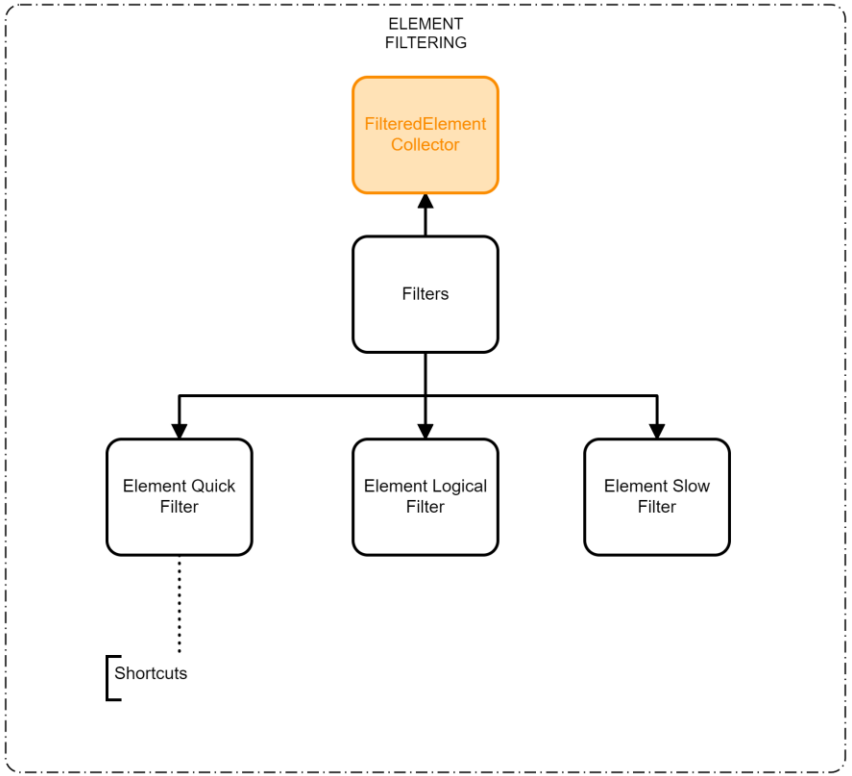
DB Elements

	Familia Sistema	Familia Cargable
Element Type	*WallType* *FloorType* ...	*FamilySymbol* & *Category - Doors, Windows...*
Instance	*Wall* *Floor* ...	*FamilyInstance* & *Category - Doors, Windows...*

Python & Revit API

Selección mediate filtros

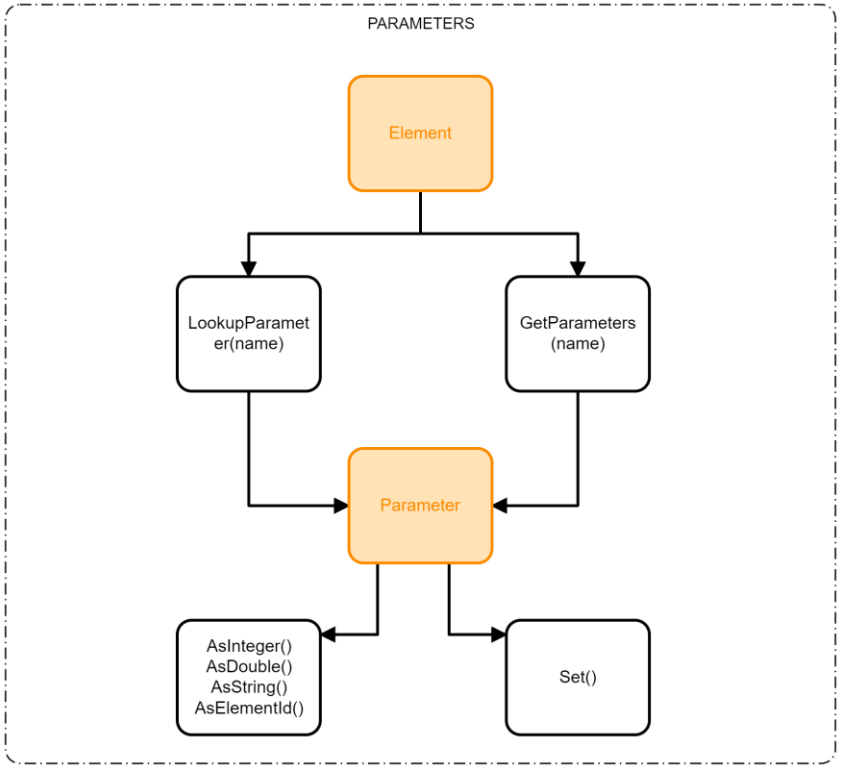
- *FilteredElementCollector(document):* colección *de* elementos
- Necesita al menos un filtro
- Shortcuts
  - OfCategory()
  - OfClass()
  - WhereElementIsNotElementType()
  - Excluding()
  - …
- No hay selección en la interfaz de usuario



Python & Revit API

Parámetros

- Todos los elementos de Revit tienen un set de propiedades
- Parámetros son **objetos** con sus propias propiedades y métodos
  - Unidades
  - Tipo de almacenamiento
  - ...
- Parámetros de tipo y de ocurrencia son el **mismo tipo de objecto**

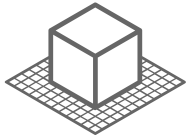


Python & Revit API

Modificación de elementos

Documento

- Métodos a nivel documento
  - Move
  - Rotate
  - Mirror
  - Array



Elemento

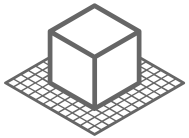
- Parámetros
- Location
- Element Type

Python & Revit API

Creación de elementos

Proyecto

- *Document.Create*
- Para la creación de
  - Group
  - FamilyInstance
  - DetailLine
  - Floor
  - ….



Familia

- *Document.FamilyCreate*
- Para la creación de
  - Sweep
  - Extrusion
  - Lofts
  - ….

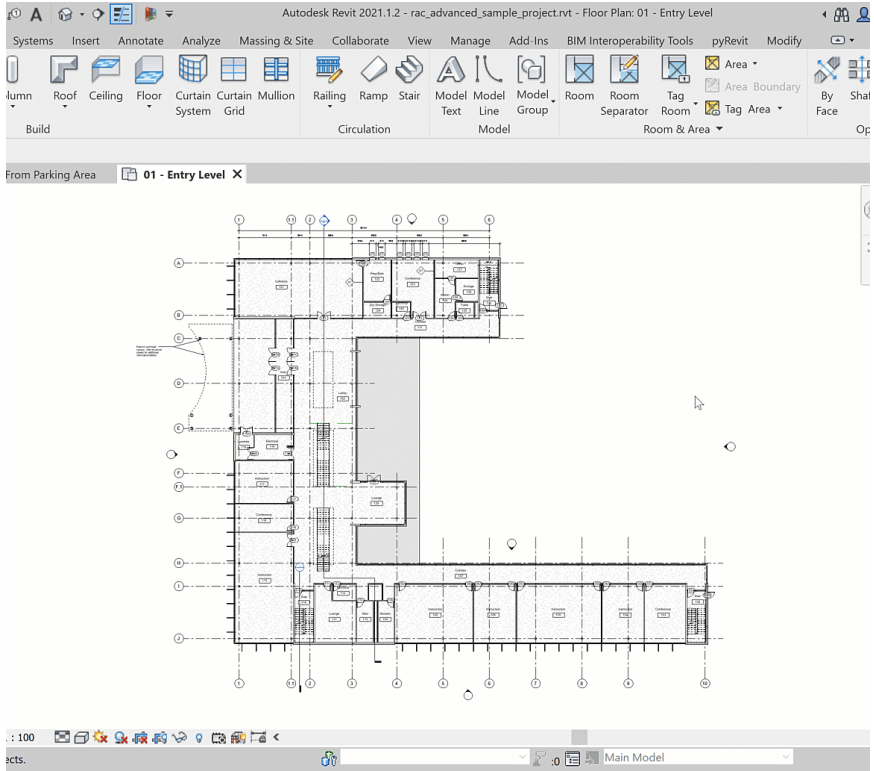
Métodos específicos

- Métodos de creación en la misma clase del objecto
- Para la creación de
  - Wall
  - AdaptiveComponent
  - …

Python & Revit API

Herramientas de ayuda

- [Revit SDK]
- [RevitApiDocs.com]
- [LookUp Tool]



Python & Revit API

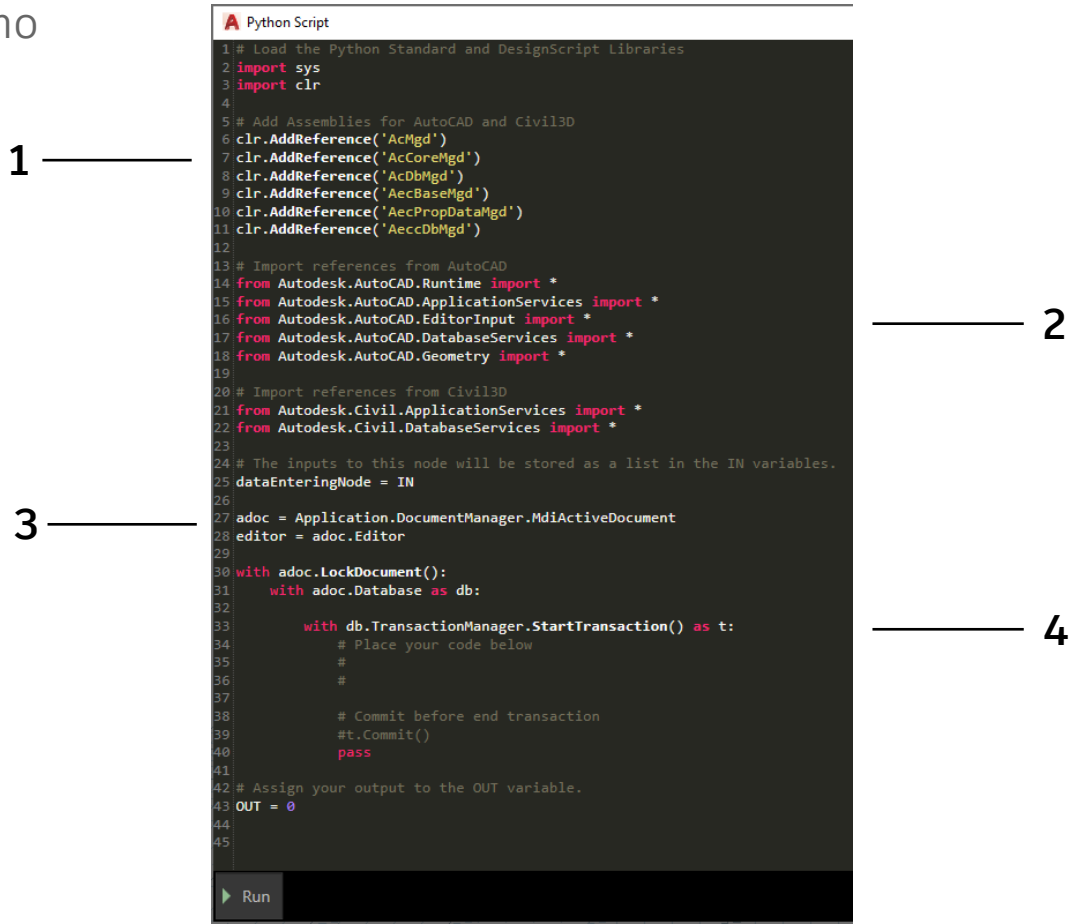
Ejemplos

- Creación de elementos
- Filtrado de elementos por nombre
- Modificación de parámetros

Python y Civil 3D API

Python & AutoCAD/Civil3D API

Plantilla en Dynamo



Python & AutoCAD/Civil3D API

Assemblies

AutoCAD

Acmgd.dll
Acdbmgd.dll
Accoremgd.dll

AutoCAD Architecture

AecBaseMgd.dll
AecPropDataMgd.dll

Civil 3D

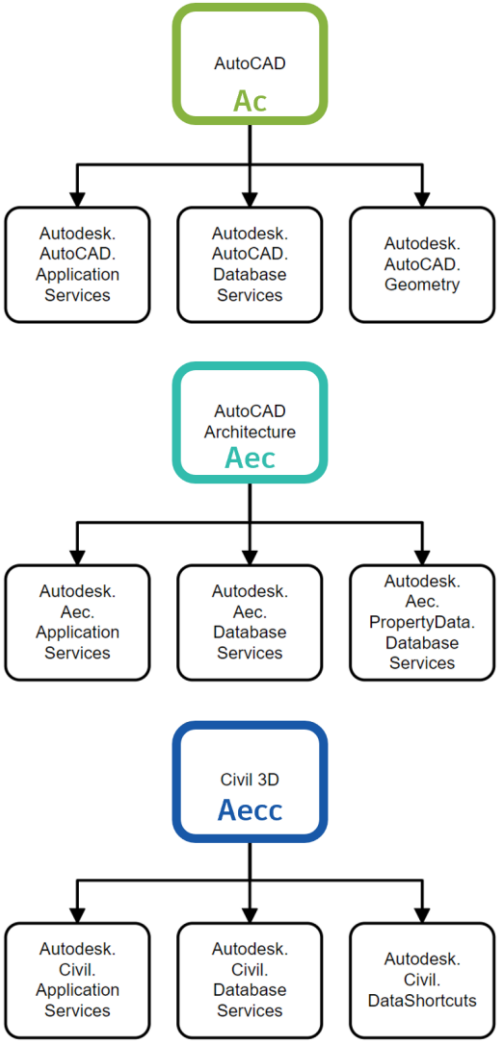
AeccDbMgd.dll
AeccPressurePipes.dll
AeccDataShortcutMgd

Python & AutoCAD/Civil3D API

Namespaces

- Añadir el *assembly* :
  - *clr.AddReference("AecBaseMgd")*
  - *clr.AddReference("AeccDbMgd")*

- Importar los objectos en el namespace:
  - *from Autodesk.AutoCAD.Geometry import **
  - *import Autodesk.AutoCAD.ApplicationServices.Application as acapp*



Python & AutoCAD/Civil 3D API

Documentos

AutoCAD Document

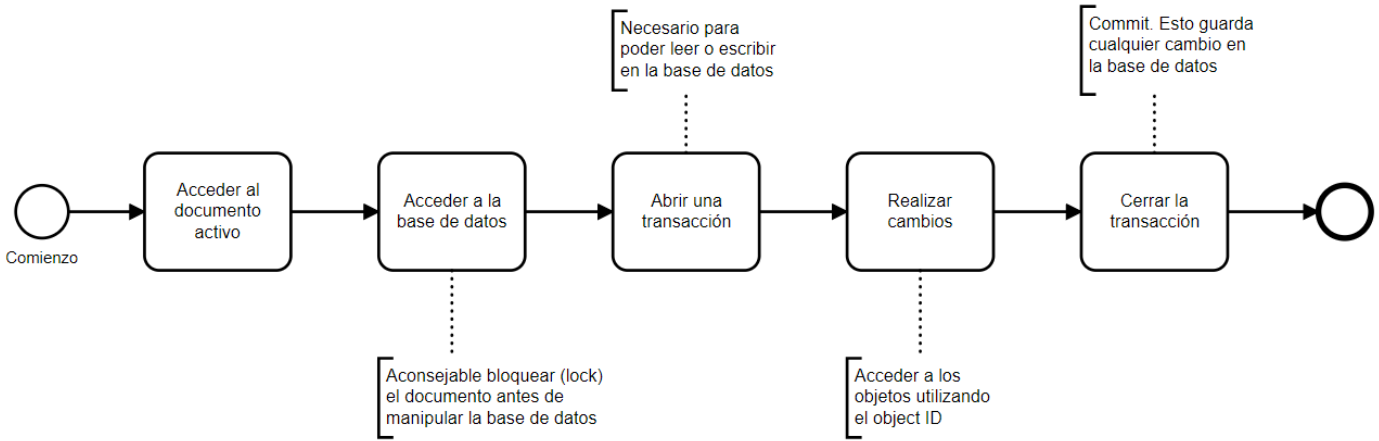
- Documento Activo
- Base de datos
- Transaction Manager
- Editor

Civil Document

- Entidades Civil 3D:
  - Superficies
  - Corredores...
- Parámetros:
  - Estilos
  - Etiquetas...

Python & AutoCAD API

Transacciones



- Photo by rupixen.com on Unsplash

Python & AutoCAD/Civil 3D API

Objetos

Object ID

- **Referencia un objeto** en la base de datos
- Entero valido solamente para la **sesión activa**
- Utilizado para **interactuar con un objeto**
- Una de sus propiedades es el Object Handle

Object ID Collection

- **Referencia a una colección (grupo) de objetos** en la base de datos
- Las entidades Civil 3D normalmente son colecciones de objetos
- Acceder al objeto dentro de la colección para interactuar con él

Handle

- **Referencia a un objeto** en la base de datos
- Hexadecimal valido para **todas las sesiones** de AutoCAD
- Utilizado para **serializar informacion**

Python & AutoCAD/Civil 3D API

Navegar el API

Developer's Guide

- Información sobre el API
- *Ejemplos (en C# generalmente)*

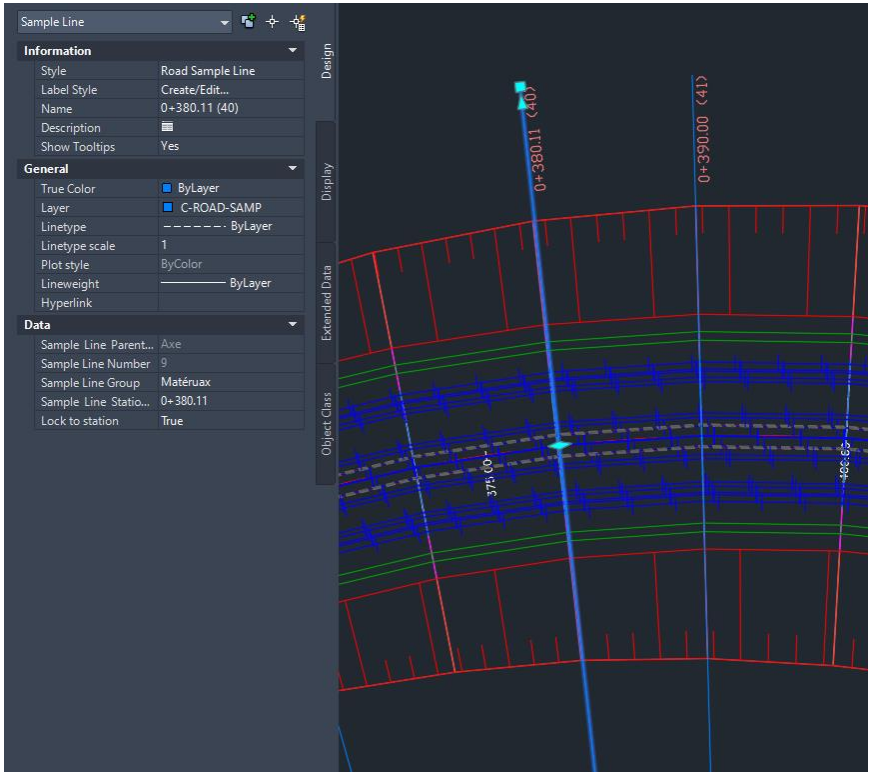
API Reference Guide

- "Diccionario" para navegar el API
- Clases y miembros de las clases :
  - Métodos
  - Propiedades

Python & AutoCAD/Civil 3D API

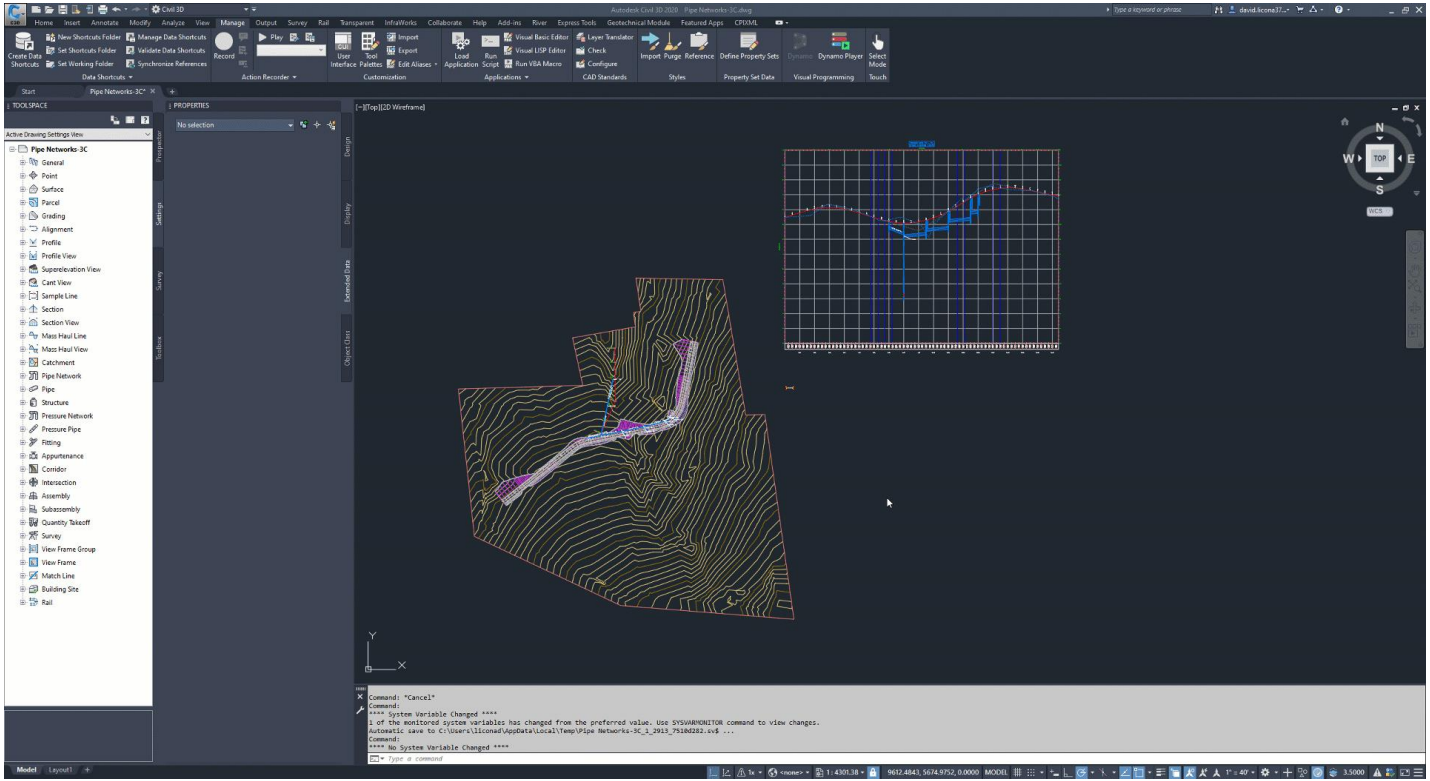
Necesario Python para interactuar con

- *Objetos :*
  - *Sample Lines*
  - *Datashortcuts*
  - *Volume surfaces*
  - *Section Views*
- *Comandos :*
  - *Audit*
  - *View*
  - *Wblock*



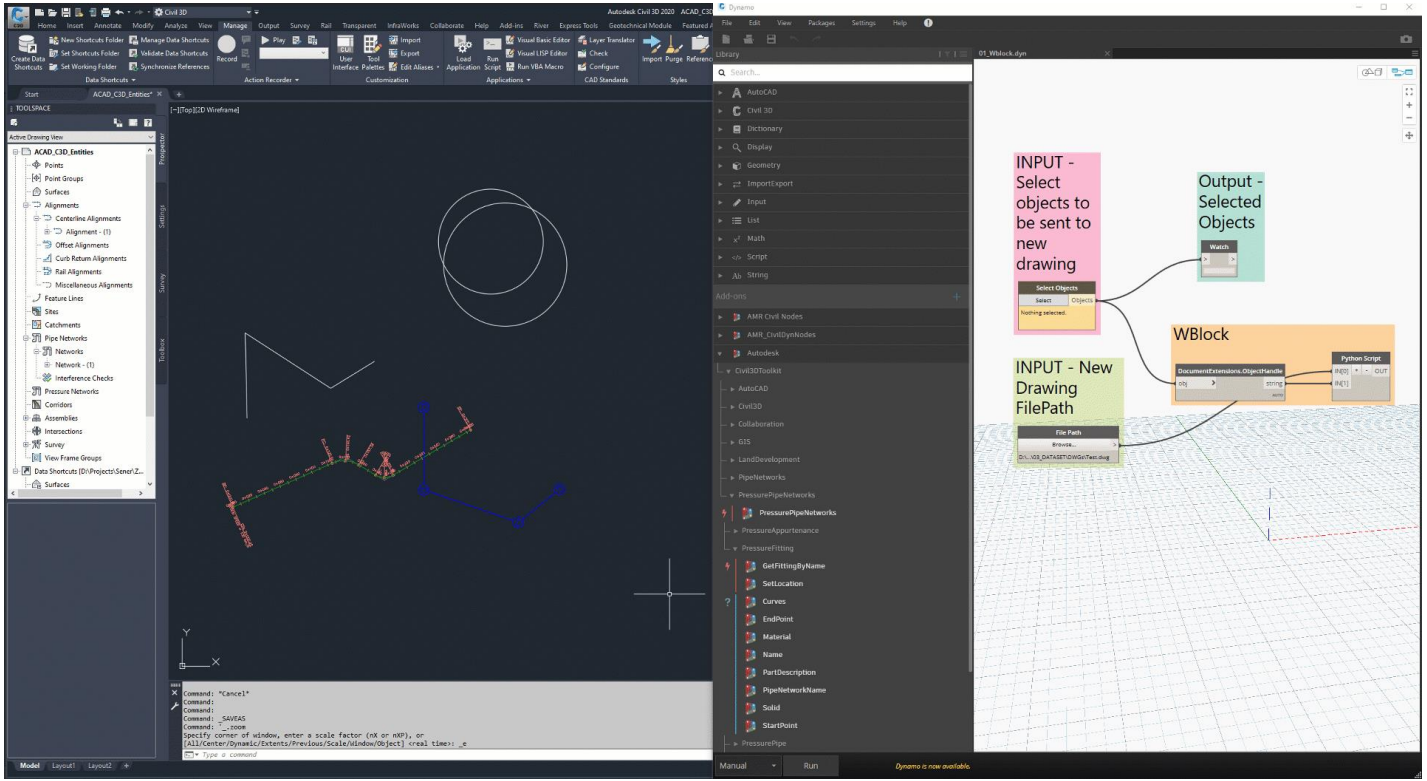
Ejemplo 1

Enviar comandos al command line de Civil 3D



Ejemplo 2

Exportar objetos a un nuevo dibujo



Recursos

Conclusiones

Objetivos de aprendizaje

- Descubrir la programación orientada a objetos
- Aprender las bases y elementos de lenguaje de Python
- Diferenciar entre Python, IronPython y Cpython
- Comprender las bases de las API de Revit, AutoCAD y Civil 3D

Primeros Pasos

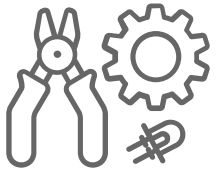
Empezando a utilizar Python para resolver problemas

Definición

Identificar problema práctico que necesite el uso de Python

Descripción

Describir el problema (BPMN)



Solución

Implementar la solución

Evaluación

Test y debug

Foro Dynamo

Subtitle

all categories ▸ | python ▸ | Latest | New (8) | Top | Categories | + New Topic

Topic	Replies	Views	Activity
Create section and rotate section in same script Revit dynamo, python	1	58	1h
Numbering every second paragraph • Developers dynamo, revit, python	2	54	16h
How can I create an Autodesk.DB.References from XYZ or DesignScript Point • dynamo, revit, python	2	37	22h
How to convert FamilyInstances to Autodesk.revit.db.element type? • dynamo, revit, python	2	52	1d
Retrieve dimension types in revit Developers revit, python, api	5	112	1d
Get PipeInsulation center reference • Developers dynamo, revit, python, geometry, mep	1	37	1d
How can I draw a viewport from a polyline? Civil3D dynamo, python, geometry	5	126	1d
Get Element from FamilyInstance Revit dynamo, python	3	202	1d
☑ Lists: Different results for build-in list-node and regex-node? • Lists-Logic python, civil3d	2	41	2d
☑ Python condition in Dynamo Revit python	2	110	2d

mzjensen Regular — lebinhx3c2012

Here's the Python script.
IN[0] = Profile View object
IN[1] = station (or list of stations)
IN[2] = elevation (or list of elevations)

The number of stations should match the number of elevations.

```
import sys
import clr

clr.AddReference('AcMgd')
clr.AddReference('AcDbMgd')
clr.AddReference('AeccDbMgd')
clr.AddReference('ProtoGeometry')

from Autodesk.AutoCAD.ApplicationServices import *
from Autodesk.AutoCAD.DatabaseServices import *

from Autodesk.Civil.DatabaseServices import *

adoc = Application.DocumentManager.MdiActiveDocument

from Autodesk.DesignScript.Geometry import *

def get_profile_view_points(profileView, station, elevation):

    global adoc

    output = []
```

Solution 4 ♡ ••• Reply

AUTODESK UNIVERSITY

Section break

Section break

Section break

Sample blank slide – title (sentence case)

Sample blank slide – subtitle (sentence case)

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

- Consectetuer adipiscing elit
- Maecenas porttitor congue massa
- Fusce posuere, magna sed pulvinar ultricies

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

- Consectetuer adipiscing elit
- Maecenas porttitor congue massa
- Fusce posuere, magna sed pulvinar ultricies

Sample 2-column content slide

Subtitle

Lorem ipsum dolor sit amet

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut

- Consectetuer adipiscing elit
- Maecenas porttitor congue massa
- Fusce posuere, magna sed pulvinar

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut

Lorem ipsum dolor sit amet

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut

- Consectetuer adipiscing elit
- Maecenas porttitor congue massa
- Fusce posuere, magna sed pulvinar

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut

Sample 3-column slide

Subtitle

Lorem ipsum dolor sit amet

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex

Stats & figures sample slide

Subtitle

90

WRITE HERE

Ut enim ad minim veniam, quis nostrud

35

WRITE HERE

Ut enim ad minim veniam, quis nostrud

70

WRITE HERE

Ut enim ad minim veniam, quis nostrud

90

WRITE HERE

Ut enim ad minim veniam, quis nostrud

Text keyword


Image keyword

Photo slide

Subtitle

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Two Photo slide

Subtitle

AUTODESK UNIVERSITY