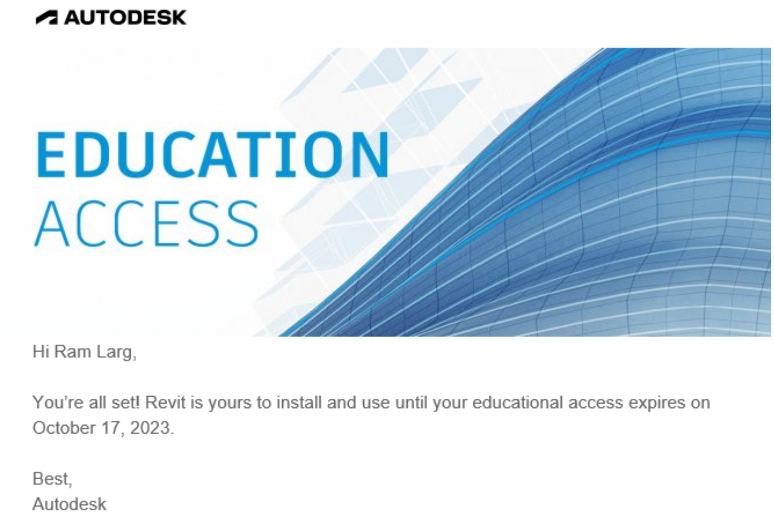
What is Revit?
Revit® BIM software helps architecture, engineering, and construction (AEC) teams create high-quality buildings and infrastructure. Use Revit to:

- **Model shapes, structures, and systems in 3D** with parametric accuracy, precision, and ease
- **Streamline documentation work**, with instant revisions to plans, elevations, schedules, and sections as projects change
- **Empower multidisciplinary teams with specialty toolsets** and a **unified project environment**

What is pyRevit?
pyRevit helps you quickly sketch out your **automation and add-on ideas inside the Revit environment using its APIs.**

- A set of free tools for Revit workflows.
- Sketch out automation ideas quickly in python.
- Write Revit add ons for Revit using python (IronPython or CPython).
- Distribute your tools (pyRevit, Dynamo, or Grasshopper) easily to your teams using a unified interface.

We are able to download Revit free with our student login. Both the installer and download files from here



AUTODESK

EDUCATION
ACCESS

Hi Ram Larg,

You're all set! Revit is yours to install and use until your educational access expires on October 17, 2023.

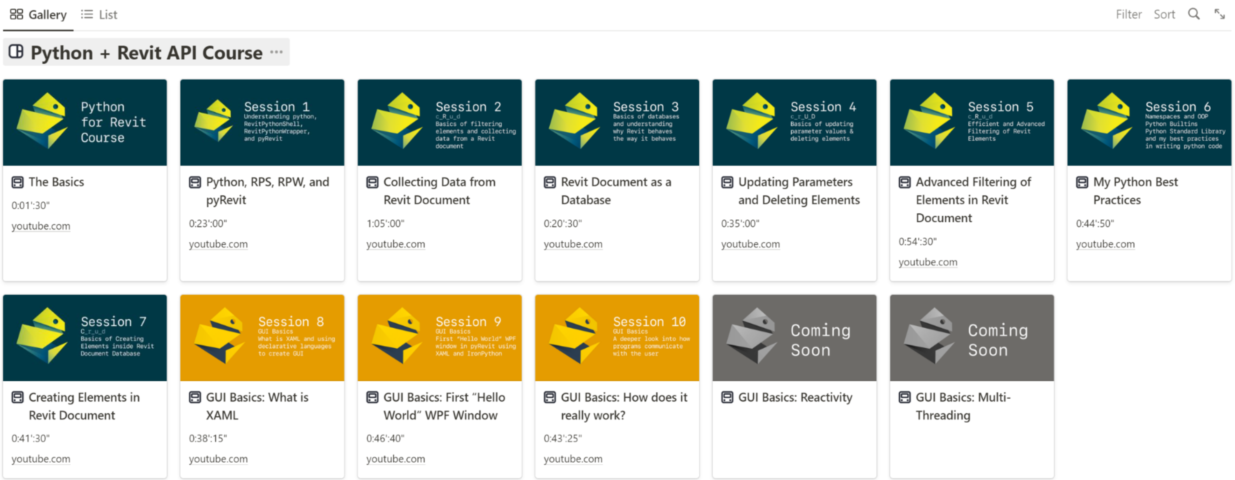
Best,
Autodesk

What we want to do is be able to access and edit data of components within a 3D Revit environment. Preferably with some sort of GUI.

- **pyRevit Installation**
- Configure pyRevit
- Install Extensions

Guide:

- [Python + Revit API Course]


Gallery
List
Filter
Sort
Python for Revit Course
Session 1
Understanding python, RevitPythonShell, RevitPythonWrapper, and pyRevit
Session 2
c_R_u_d
Basics of filtering elements and collecting data from a Revit document
Session 3
Basics of databases and understanding why Revit behaves the way it behaves
Session 4
c_r_U_D
Basics of updating parameter values & deleting elements
Session 5
c_R_u_d
Efficient and Advanced Filtering of Revit Elements
Session 6
Namespaces and OOP Python Builtins Python Standard Library and my best practices in writing python code
Session 7
C_r_u_d
Basics of Creating Elements inside Revit Document Database
Session 8
GUI Basics
What is XAML and using declarative languages to create GUI
Session 9
GUI Basics
First "Hello World" WPF window in pyRevit using XAML and IronPython
Session 10
GUI Basics
A deeper look into how programs communicate with the user
Coming Soon
Coming Soon

Python + Revit API Course

The Basics
0:01':30"
youtube.com

Python, RPS, RPW, and pyRevit
0:23':00"
youtube.com

Collecting Data from Revit Document
1:05':00"
youtube.com

Revit Document as a Database
0:20':30"
youtube.com

Updating Parameters and Deleting Elements
0:35':00"
youtube.com

Advanced Filtering of Elements in Revit Document
0:54':30"
youtube.com

My Python Best Practices
0:44':50"
youtube.com

Creating Elements in Revit Document
0:41':30"
youtube.com

GUI Basics: What is XAML
0:38':15"
youtube.com

GUI Basics: First "Hello World" WPF Window
0:46':40"
youtube.com

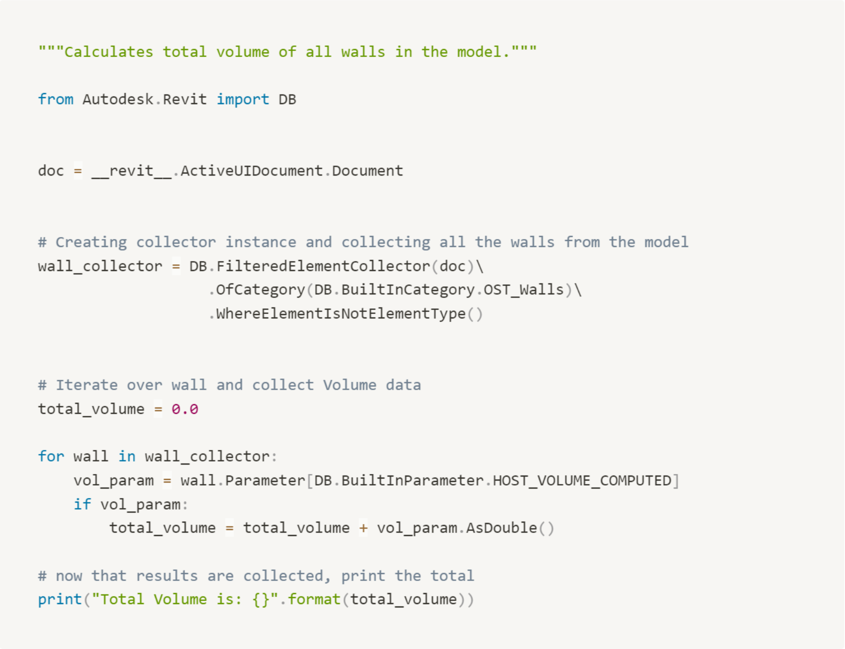
GUI Basics: How does it really work?
0:43':25"
youtube.com

GUI Basics: Reactivity

GUI Basics: Multi-Threading

Creating Commands/Scripts:

- [Create Commands/Scripts]

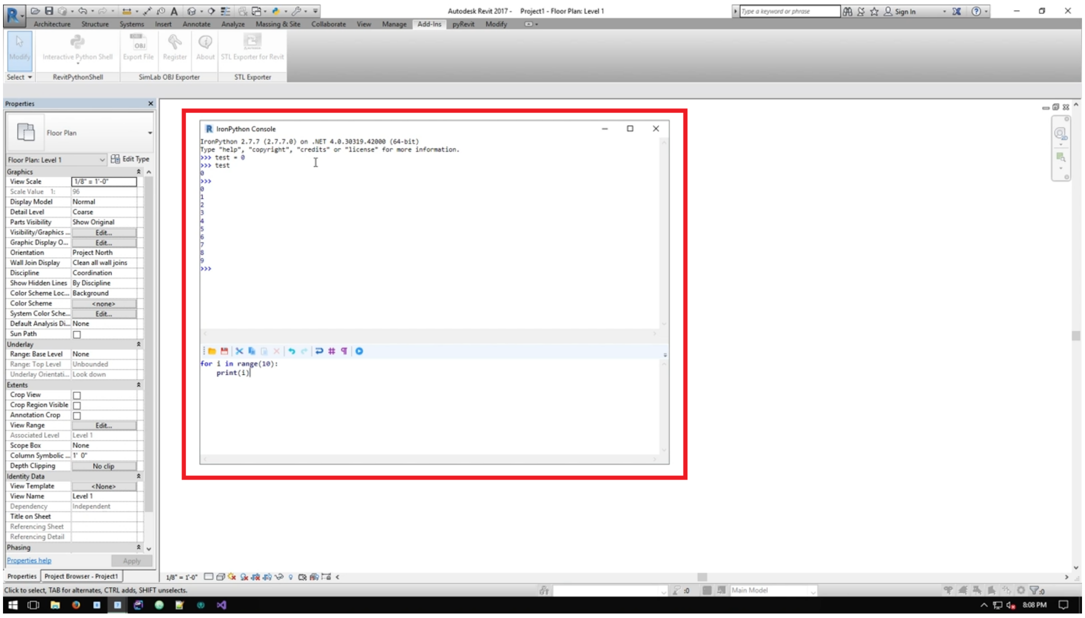


```python
"""Calculates total volume of all walls in the model."""

from Autodesk.Revit import DB


doc = __revit__.ActiveUIDocument.Document


# Creating collector instance and collecting all the walls from the model
wall_collector = DB.FilteredElementCollector(doc)\
                   .OfCategory(DB.BuiltInCategory.OST_Walls)\
                   .WhereElementIsNotElementType()


# Iterate over wall and collect Volume data
total_volume = 0.0

for wall in wall_collector:
    vol_param = wall.Parameter[DB.BuiltInParameter.HOST_VOLUME_COMPUTED]
    if vol_param:
        total_volume = total_volume + vol_param.AsDouble()

# now that results are collected, print the total
print("Total Volume is: {}".format(total_volume))
```

Python for Revit: python, RPS, RPW, and pyRevit:

- **IronPython**
  - built on python 2.7
  - For .NET platforms
    - .NET used in Autodesk Revit
  - Ironpython to connect to .NET to connect to revit to perform tasks/query information

  - Revit comes shipped with IronPython
- **RevitPythonShell (RPS) add-on**



IronPython Console

```
IronPython 2.7.7 (2.7.7.0) on .NET 4.0.30319.42000 (64-bit)
Type "help", "copyright", "credits" or "license" for more information.
>>> test = 0
>>> test
0
>>>
0
1
2
3
4
5
6
7
8
9
>>>
```

```
for i in range(10):
    print(i)
```

Floor Plan: Level 1

Graphics	
View Scale	1/8" = 1'-0"
Scale Value 1:	96
Display Model	Normal
Detail Level	Coarse
Parts Visibility	Show Original
Visibility/Graphics ...	Edit...
Graphic Display O...	Edit...
Orientation	Project North
Wall Join Display	Clean all wall joins
Discipline	Coordination
Show Hidden Lines	By Discipline
Color Scheme Loc...	Background
Color Scheme	<none>
System Color Sche...	Edit...
Default Analysis Di...	None
Sun Path	□
Underlay	
Range: Base Level	None
Range: Top Level	Unbounded
Underlay Orientati...	Look down
Extents	
Crop View	□
Crop Region Visible	□
Annotation Crop	□
View Range	Edit...
Associated Level	Level 1
Scope Box	None
Column Symbolic ...	1' 0"
Depth Clipping	No clip
Identity Data	
View Template	<None>
View Name	Level 1
Dependency	Independent
Title on Sheet	
Referencing Sheet	
Referencing Detail	
Phasing	

Click to select, TAB for alternates, CTRL adds, SHIFT unselects.

- Used to inspect into our selections
  - Test ideas before implementing
- **Revit Python Wrapper (RPW)**
  - Goal: provide more pythonic interface to Revit API
  - Simpler to able to communicate to Revit
  - Revit’s API and documentation is written in C#
    - Which may be more complex

Python for Revit: Basics of collecting data from Revit database