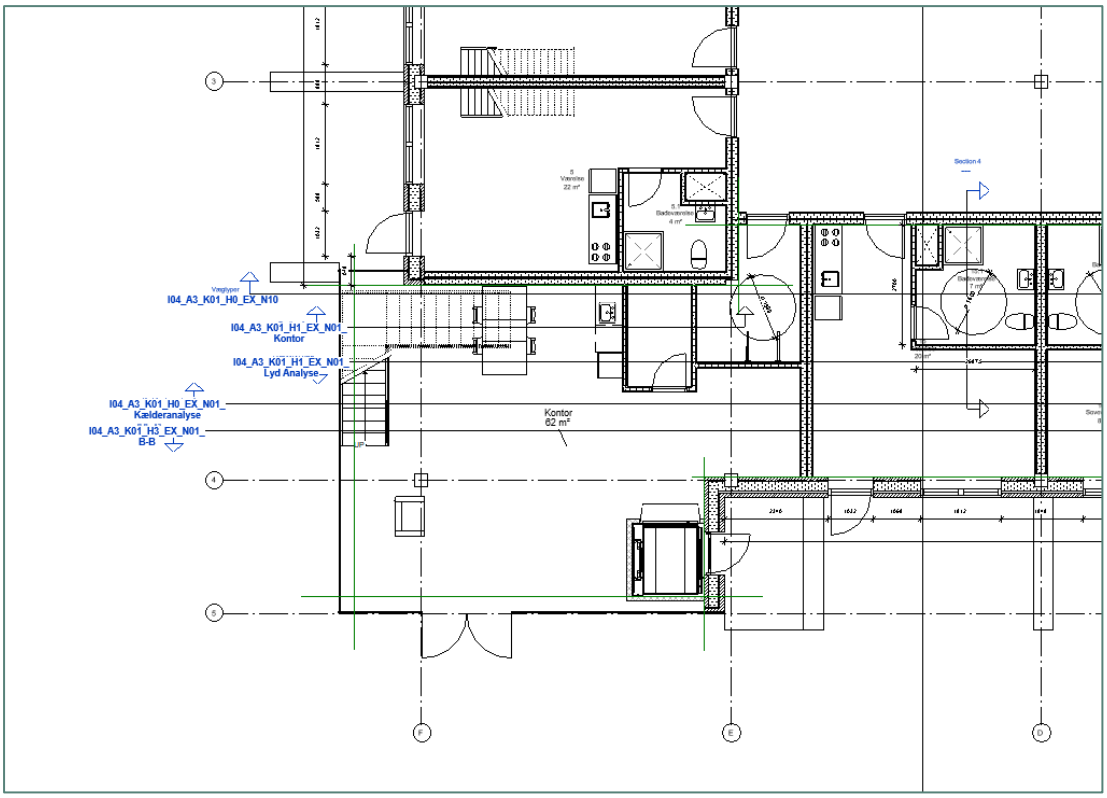
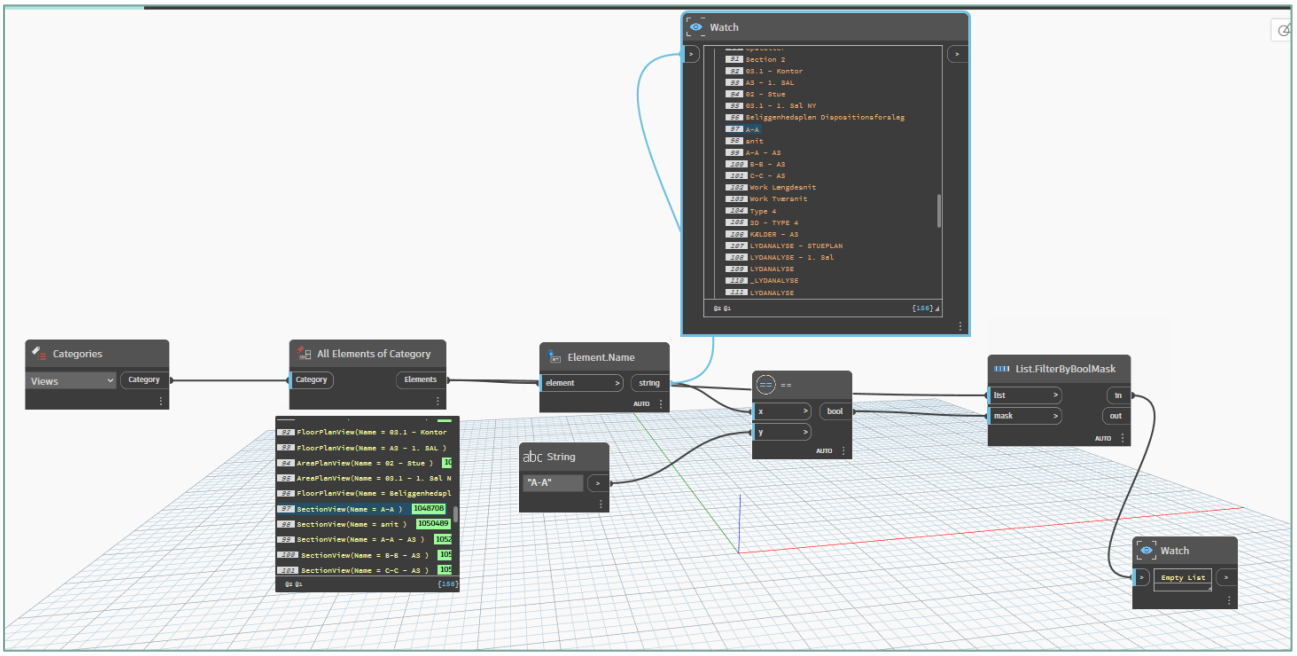
Vis bestemte snit:



Vi sidder mange i Revit filen, og bruger mange forskellige snit. I revit er der mange muligheder for at skjule dem. Hide dem manuelt, bruge view template og bruge parameter. Men dette kan tage langt tid,



derfor undersøgte jeg om det ikke kunne være muligt at gøre med Dynamo, for at gøre det noget hurtigere.

Der er prøvet med boksene, men da Sections ikke er en parameter i Revit. Samt Dynamo ikke havde en HideElements kasse. Har jeg ikke kunne få det til at virke på denne måde.

Men jeg gav ikke op så let, derfor har jeg brugt ChatGPT til at hjælpe med en Python-kode så det kunne lykkedes alligevel.

Først skal der bruges .NET-bibliotekter som Dynamo bruger til at kommunikere med Revit:

```
import clr
clr.AddReference('RevitServices')
clr.AddReference('RevitAPI')
clr.AddReference('System')
```

Her importerer vi værktøjer til at hente dokumentet (doc) og arbejde med Elementer (Element, Transaction mm)
List bruges til at oprette en .NET-liste af ElementId som kræves for HideElements():

```
from RevitServices.Persistence import DocumentManager
from Autodesk.Revit.DB import *
from System.Collections.Generic import List
```

Doc er det aktive Revit-dokument
Active_view er den visning, som Dynamo-scriptet køres i. (er du i en plantegning køres scriptet der, er du section køres den der osv.)

```
doc = DocumentManager.Instance.CurrentDBDocument
active_view = DocumentManager.Instance.
CurrentUIApplication.ActiveUIDocument.ActiveView
```

Her skal vi indsætte hvilke snit vi vil have der skal være fremme.

```
tilladte = ["A-A", "B-B", "C-C", "Work Længdesnit", "Work Tværsnit"]
```

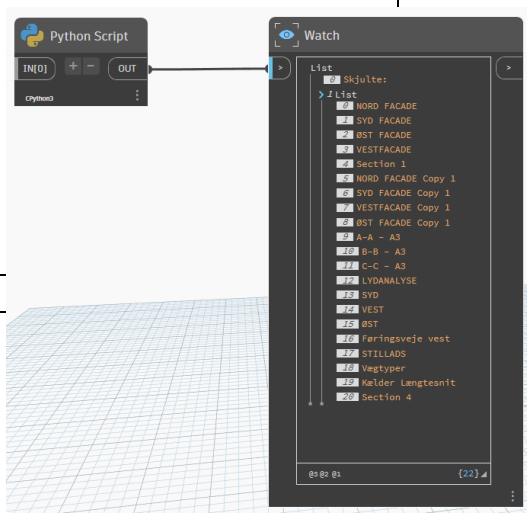
Her finder alle elementer der ikke er skjult.

```
collector = FilteredElementCollector(doc,
active_view.Id).WhereElementIsNotElementType()
```

Så skal der udvælges hvilke snit der skal skjules:
Koden filtrerer snitlinjer i kategorien Views
Kun snit der **ikke matcher de tilladte navne** bliver skjult
De tilføjes til to lister: skjul_ids (som er ElementID'er der bliver skjult) og Skjulte_navne (De snit der skal vises)

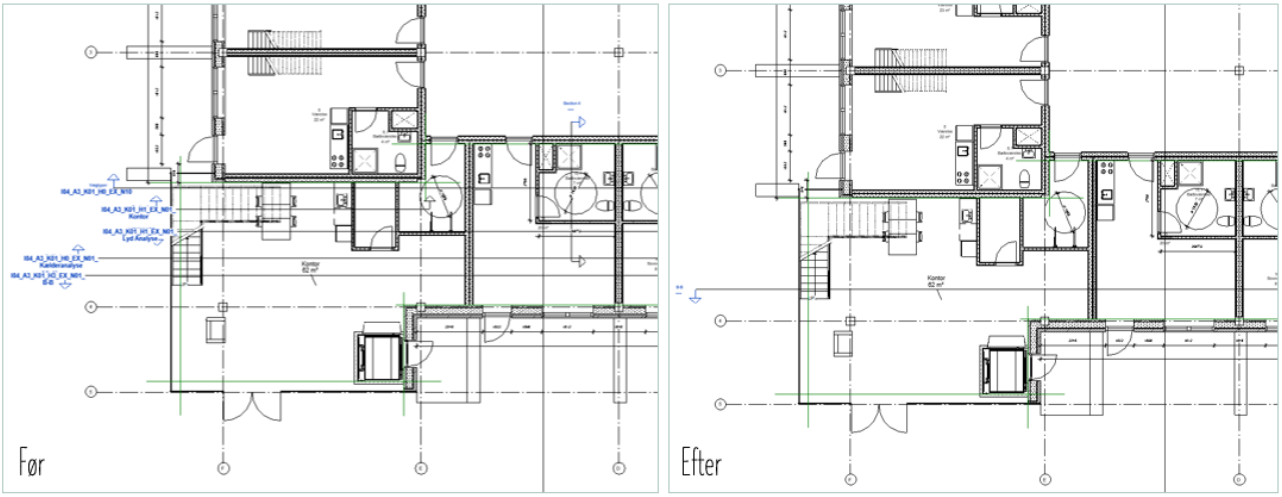
```
skjul_ids = List[ElementId]()
skjulte_navne = []

for el in collector:
  if el.Category and el.Category.Name == "Views":
    if el.Name.lower() not in [n.lower() for n in tilladte]:
      skjul_ids.Add(el.Id)
      skjulte_navne.append(el.Name)
```



Revit kræver en transaktion for at kunne ændre modellen

HideElements() skjuler snit i den aktive visning t = Transaction(doc, "Skjul snit-elementer") t.Start() active_view.HideElements(skjul_ids) t.Commit()
Og til sidst retuneres en liste med navnene på de snit der bliver skjult, så det kan ses i Dynamo OUT = "Skjulte:", skjulte_navne



Ulemper:

1. Kører kun i aktiv visning
   Scriptet virker kun i den visning, der er aktiv, når det afvikles. Ønsker man at skjule snit i flere visninger, skal det køres igen i hver enkelt visning.
2. Ikke dynamisk opdatering
   Nye snit, der bliver oprettet efter scriptet, er kørt, vil ikke automatisk blive skjult. Man skal selv køre scriptet igen for at opretholde renset visning.
3. Afhænger af præcis navngivning
   Listen tilladte kræver, at navnet på snittene matcher helt præcist (bortset fra store/små bogstaver). Hvis der opstår variationer i navne, bliver de forkerte snit vist eller skjult.
4. Teknisk opsætning kræver forståelse
   Brugen af Python og Revit API kræver en vis teknisk forståelse. Det kan være en barriere, hvis scriptet skal deles med kollegaer uden erfaring med Dynamo eller programmering.

Konklusion:

Dette script er nyttigt for at rydde op i sine tegninger, så du hurtigt og nemt kan skjule unødvendige snit uden at fjerne dem alle manuelt. Det er hurtigt, fleksibelt og nemt at tilpasse dem ved blot at ændre i tilladte-listen. Scriptet er især nyttigt i samarbejdsprojekter, hvor mange brugere arbejder i samme model og opretter forskellige snit og hvor det hurtigt kan blive uoverskueligt at finde rundt i dem alle.