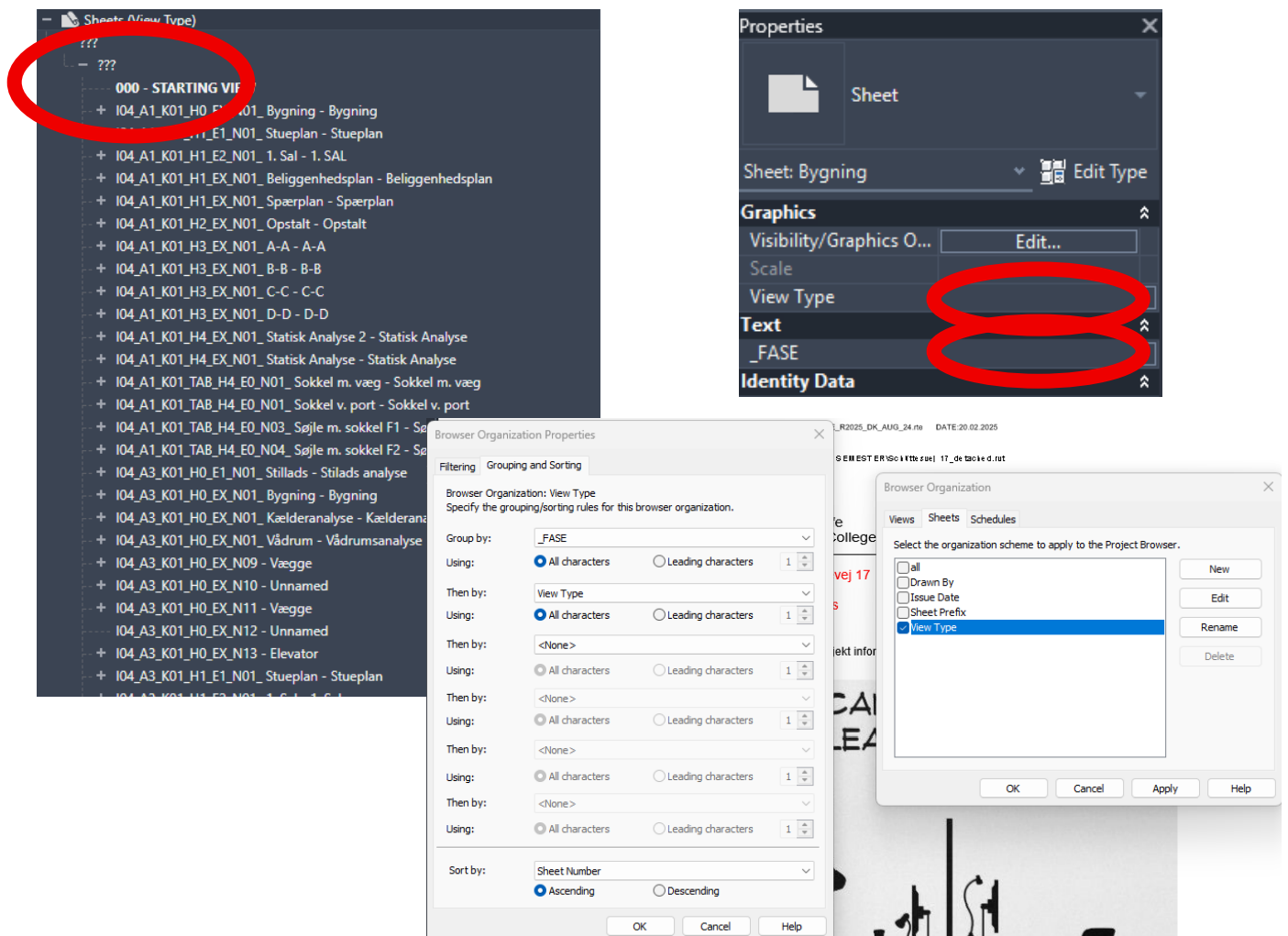


# Sortere sheets automatisk ved brug af navngivning struktur

Det er vigtigt når vi har med så mange sheets og faser at gøre, at man har det opryddeligt. I revit er det muligt at sortere dem manuelt med parameter, MEN hvis man sidder med 20 sheets i hver fase bliver det hurtigt uoverskueligt og langtrukken. Derfor undersøgte jeg om det ikke kunne være muligt at sortere sheets på en nemmere måde.

Når man laver sheets inde i revit så når man opretter en ny vil den komme under "???" fordi man ikke har valgt hvilken View Type sheetet skal være i. Manuelt ville man skulle ind og skrive i View Type om det er en Plantegning, Snit, Detaljer osv. Og så har jeg oprettet en parameter der hedder Fase. Her skal man også manuelt ind og skrive i hvilken fase sheet skal være i.



Det har jeg fundet en løsning på. Jeg har manuelt gået ind og valgt at Sheets View Type skal sortere efter Fase og derefter View Type. Og det er her Dynamo kommer ind i billedet.

Jeg har lavet et script, hvor den ud fra "Sheet Number" - Som er navngivningen på Sheet, at når den hedder tegningsnummeret hedder:

```
["A1","A2","A3","A4","A5","A6","A7"];  
["1 - Registrering","2 - Program","3 - Dispositionsforslag","4 - Projektforslag","5 - Myndighedhedsprojekt","6 - Hovedprojekt","7 - Udførelse"];
```

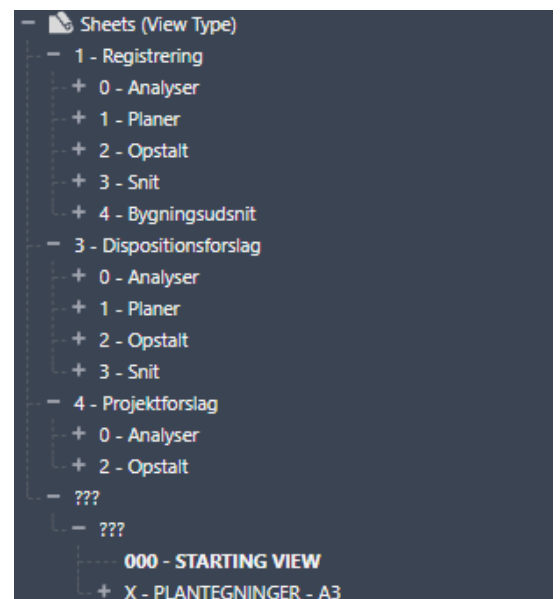
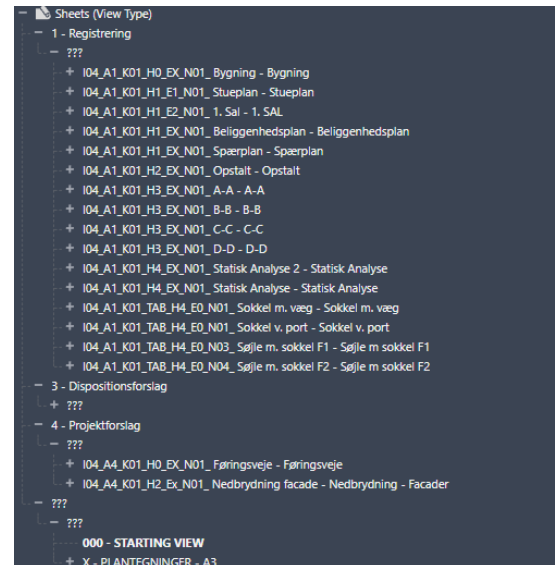
Så tilføjer den automatisk hvilken fase Tegningen hører til.

Efter man så har kørt Scriptet, har den automatisk sorteret alle Sheets til at være under de forskellige faser. Men dette kræver at man har navngivet korrekt, eller vil den stadigvæk være under "???"

Samme princip har jeg gjort i samme Script, hvor den sortere efter View Type

```
["H0","H1","H2","H3","H4","H5"];  
["0 - Analyser","1 - Planer","2 - Opstalt","3 - Snit","4 - Bygningsudsnit","5 - Detaljer"];
```

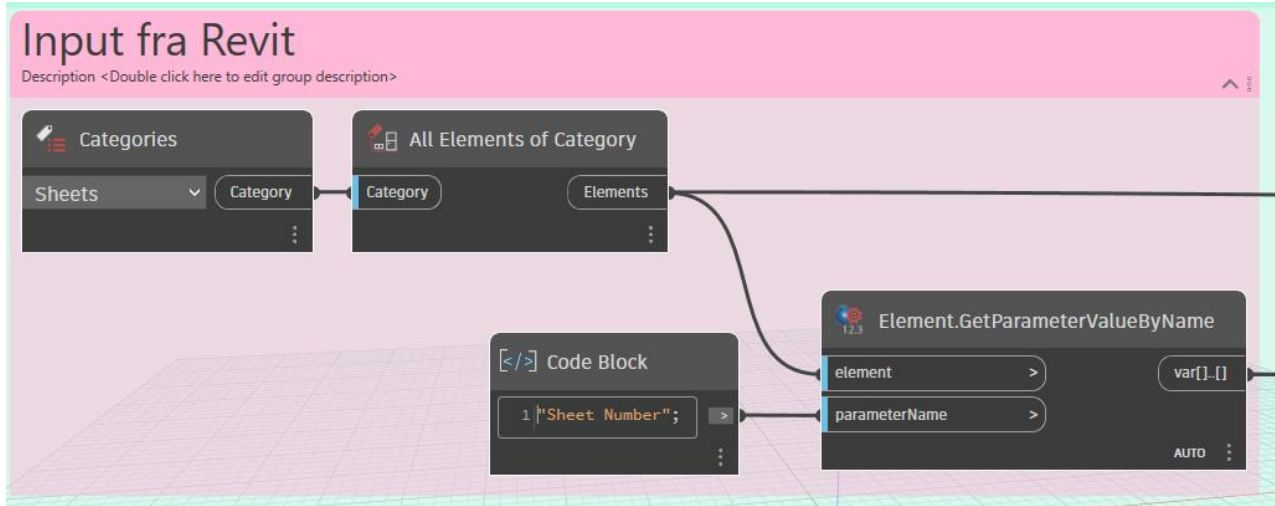
Og nu er det sorteret så det er overskueligt at navigere rundt i.



## Selve Scriptet:

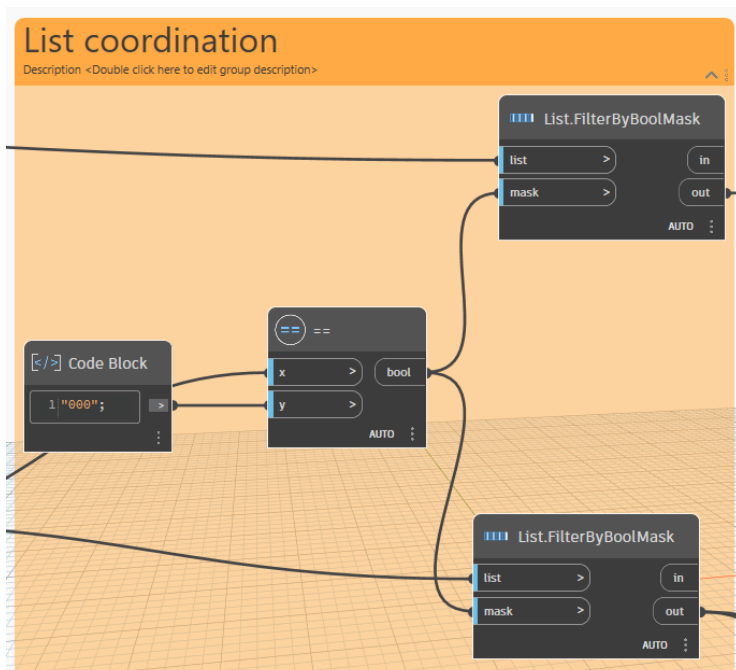
### 1. Henter alle Sheets ud fra Sheet Number

Først henter vi alle "Sheet Number" fra alle Revit-tegningerne i det åbne projekt. Det er valgt kategorien Sheets da det er det vi vil arbejde med. Hvor vi herefter har koblet den til noden "All Elements of Category" hvilket betyder vi henter ALLE elementer inden for denne kategori. For at få fat i arkets nummer bruges noden "GetParameterValueByName" hvor vi beder om parameteren "Sheet Number" fra hvert Sheet



### 2. Sorterer "000" fra

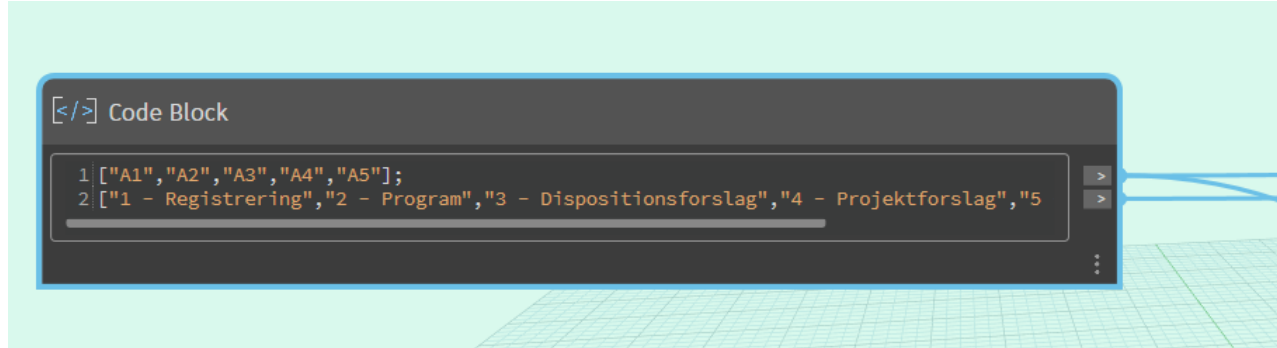
Dernæst skal vi adskille de ark som har "000" som Sheet Nummer (dette er typisk forsiden som ikke er relevant). Dette gøres ved at sammenligne hver "Sheet Number" med værdien "000". Her anvender vi en Code Block med teksten "000" og en lig-med-node (==), som sammenligner hvert nummer og returnerer en liste med true/false afhængigt om det matcher.



Ved hjælp af "List.FilterByBoolMask" sorteres Sheets i to grupper.

De Sheets hvor Sheet Number er "000"  
De andre Sheets som ikke er "000"

### 3. Generer projekteringsfase til parameteren \_FASE



Vi arbejder nu med to lister:

`["A1","A2","A3","A4","A5","A6","A7"];`

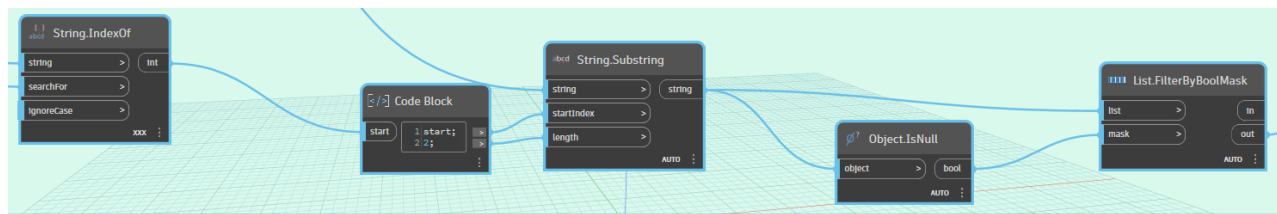
- Som er tallet udfra Sheet Number

`["1 - Registrering","2 - Program","3 - Dispositionsforslag","4 - Projektforslag","5 - Myndighedhedsprojekt","6 - Hovedprojekt","7 - Udførelse"];`

- Som er projekteringsfaserne

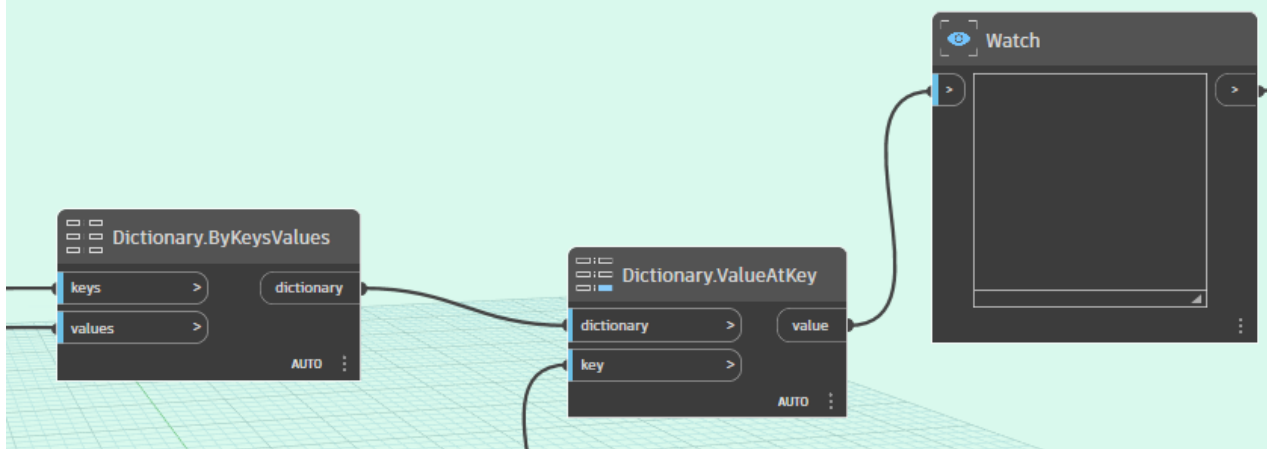
Der anvendes to metoder til at koble Sheet Number til Fasen

#### Metode 1:



String.IndexOf bruges til at finde positionen af A1 i listen ["A1","A2","A3","A4","A5","A6","A7"]; Hvilket giver index 2. Dette index bruges derefter til at hente projekteringsfaserne fra den anden liste.

#### Metode 2:



En dictionary bygges direkte med

`keys = ["A1","A2","A3","A4","A5","A6","A7"];`

og

`values = ["1 - Registrering","2 - Program","3 - Dispositionsforlag","4 - Projektforlag","5 - Myndighedhedsprojekt","6 - Hovedprojekt","7 - Udførelse"];`

Dictionary.ValueAtKey slår derefter Sheet Number op og returnerer den rigtige projekteringsfase.

Begge metoder sikrer at hver tegnings Sheet Number automatisk kobles med den korrekte projekteringsfase og skrives i parameteren `_FASE`. Dette gør det nemt at styre og opdatere faser i projekt materialet.

#### 4. Skriver til `_FASE`



Element.SetParameterByName

Denne node skriver værdi til en bestemt parameter.

Element: Alle Sheets

ParameterName = Navnet på det parameter i revit vi ændrer (\_FASE)  
Value = Den tekstværdi der skal ind i parameteren.

Det er altså det her punkt der sørger for at scriptet skriver resultatet tilbage til Revit. Uden den ville alt det vi har gjort blot være i Dynamo, uden at opdatere modellen.

## Ulemper:

### 1. Afhængighed af korrekt navngivning

Hele løsningen bygger på, at Sheet Number følger navngivningsstruktur (f.eks. "A1", "H2"). Hvis én ark får en forkert betegnelse, bliver det ikke sorteret korrekt og kan ende under "???" eller **under de forkerte faser i Revit-browseren.**

### 2. Ikke fleksibel ved ændringer i struktur

Hvis betegnelser eller tegningsnumre ændres senere i projektet, kræver det, at scriptet også opdateres manuelt. Det gør det mindre robust ved ændringer i projekthierarkiet.

### 3. Kræver Dynamo-viden for vedligeholdelse

Selvom det virker effektivt, kræver det en vis forståelse for Dynamo at oprette og tilpasse scriptet. Hvis en anden projektmedarbejder ikke har samme kendskab, kan det være svært at rette eller tilføje funktionalitet.

### 4. Fejl vises ikke tydeligt i Revit

Hvis et ark ikke matcher nogen af betingelserne (fx forkert navn), får man ikke nødvendigvis en tydelig fejlbesked. Det kan føre til skjulte fejl, der først opdages sent i projektet.

## Konklusion:

Ved hjælp af dynamo har jeg automatiseret sortering og kategorisering af Sheets ud fra navngivning. Gennem en struktur hvor tegningsnumre kobles direkte til projekteringsfase og tegningstyper, kan scriptet automatisk udfylde relevante parametre som \_FASE og View Type. Det gør at Sheets straks bliver sorteret korrekt i projektbrowseren uden behov for manuel indtastning. Det kræver dog at sheet-navngivningen er korrekt. Det gør arbejdet hurtigere, mere overskueligt og minimerer risikoen for fejl. Denne metode er værdifuld i projekter med mange sheets og faser, hvor overblik og struktur er afgørende. Dette er et fint eksempel på hvordan Dynamo kan bruges til at effektivisere opgaver i revit.