

# Opplegg 17 - Uniformt sannsyn

## Uniformt sannsyn

Alle utvegar har like stort sannsyn for å hende.

Døme på uniformt sannsyn kan vere å trekke eit kort frå ein kortstokk, eller å kaste mynt eller krone. Det er ingen grunn til at det skal vere meir sannsynleg å trekke akkurat hjerter tre enn kløver fem. Det er heller ingen grunn til at det skulle vere meir sannsynleg at ein mynt landar med krona opp, enn myntsida opp (den med hovudet). Vi ser bort fra det utruleg vesle sannsynet for at mynten landar på kanten.



## Diskuter

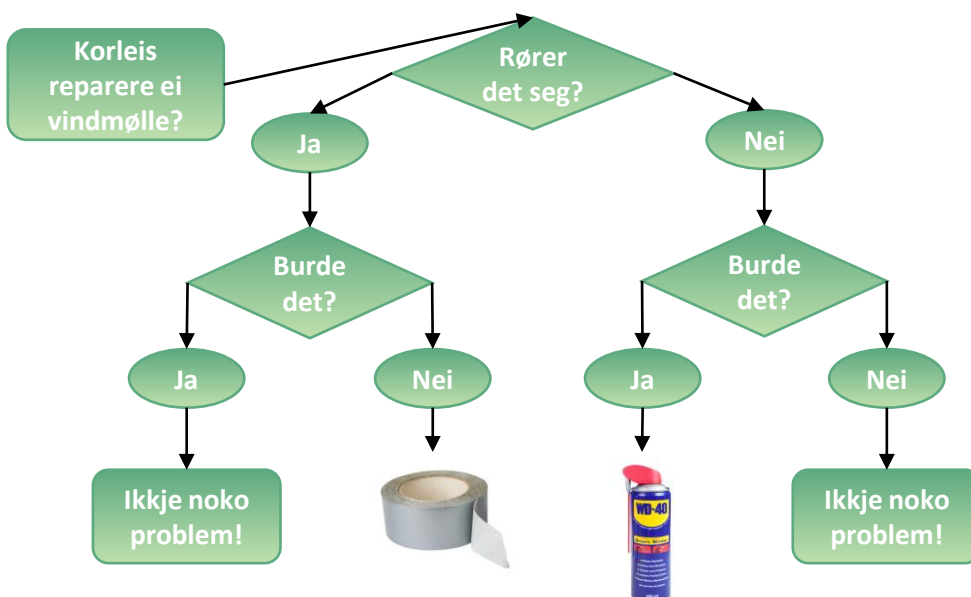
1. Kan du finne nokon andre eksempel på uniformt sannsyn?
2. Kan du finne nokre døme som ikkje er uniforme?
3. Korleis kan vi sjekke om døma våre er uniforme eller ikkje?

## Pseudokode og flytskjema

I byrjinga av boka er det to oppgåver der de skal programmere kvarandre, utan å bruke eit programmeringsspråk. Da endte de opp med å skrive ei oppskrift, ein algoritme, med vanlege ord. Dette kan kallast pseudokode. Altså det å skrive eit program med det vanlege språket vårt, gjerne med stikkord, men ikkje med eit programmeringsspråk.

Når ein skal lage eit litt større program, kan det vere lurt å bruke pseudokode først. Da skriv du med eigne ord kva programmet skal gjere, og etterpå set du dette om til enten Scratch- eller Pythonkode.

Eit alternativ er å lage eit flytskjema først. Et flytskjema er eit slags oversyn over programmet. Ein nyttar ulike små figurar som tyder ulike ting. Ein form for spørsmål, ein form for svar, og ein form for byrjing og slutt. Du kan bruke både flytskjema og pseudokode når du skal lage eit program, men som regel er det greit å velje det som passar best for deg.



Byrjing eller slutt

Svar

Spørsmål

## Oppgaver

1. Kva viser dette flytskjemaet?
2. Lag eit flytskjema for korleis rydde rommet ditt (søk på «funny flowcharts» for inspirasjon).
3. Lag ein pseudokode for å rydde rommet ditt.



# Lag ein terning eller eit lukkehjul

- og eit program som simulerer sannsynet



© Inter IKEA Systems B.V. 2020

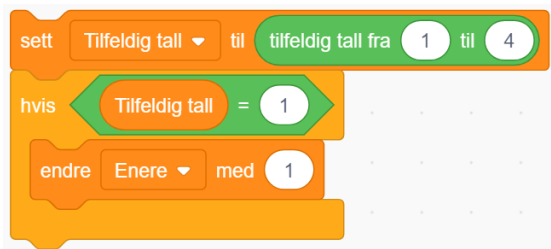
## Oppgåve

Lag ein eigendesigna terning som ikkje har seks sider, eller eit lukkehjul som seinare skal vere ein del av eit spel. Du skal lage eit program som du kan bruke til å simulere terningkast eller lukkehjul-snurr og finne sannsynet for dei ulike utfalla.

**Fase 1:** Finn informasjon og inspirasjon til ein terning eller et lukkehjul. Kva for ei form kan ein terning ha? Kva for nokre delar er eit lukkehjul samansett av? Korleis ser desse delane ut?

**Fase 2:** Ha ei idèmyldring for deg sjølv. Teikne gjerne ei skisse før du diskuterer med dei andre. Deretter må gruppa avgjere korleis terningen/lukkehjulet skal sjå ut.

**Fase 3:** Tid for å lage simuleringsprogrammet, og å lage terningen eller lukkehjulet. Sjå på programmeringsdøma for eit firedelt lukkehjul under:



Kva trur du desse programma gjer?  
Korleis kan du bruke det i ditt simuleringsprogram?

Lag eit simuleringsprogram – det finnest ark med HINT (spør lærar)

Del oppgåva opp i fleire små delar

- Veit du korleis du må programmere for å få eit tilfeldig terningkast eller lukkehjul-spinn?
- Veit du korleis du må programmere for å telje opp kor mange det blir av kvart utfall?
- Dersom du ikkje veit, kan du finne det ut?
- Kan du bruke noko frå programmeringsdøma over?
- Korleis skal du kombinere dei ulike delane?



**Fase 4:** Test terningen eller lukkehjulet og programmet ditt.

**Fase 5:** Verkar alt slik det skal?

**Fase 6:** Hopp gjerne attende til tidlegare punkt og gjer endringar for å få terningen eller lukkehjulet best mogleg. Gjer gjerne endringar i modelleringsprogrammet ditt. Kan det gjerast meir effektivt eller ryddigare?

**Fase 7:** Lag eit ark med bilete frå heile prosessen saman med resultatet for sannsynet til dei ulike utfalla.