

# Kompetansemål som deler av det viktigste i faget

Nettverkssamling Oppland,  
Gjøvik, 6. januar – 2020  
Kl. - 9.30- 15.30

# Nettverkssamlinger 2019-2020

| Datoer for samlingene | Tema   |
|-----------------------|--|
| 14.10.19              | Dybdelæring, kjerneelementer og det viktigste i faget. |
| 06.01.20              | Kompetansemålene                                       |
| 23.03.20              | Matematikkfaget som del av et større hele              |

# Tidsplan

- 09.30 – 09.35 Velkommen og plan for dagen
- 09.35 – 10.15 Erfaringsdeling fra forarbeidet, I + G (10+30)
- 10.15 – 10.30 Kaffe
- 10.30 – 11.30 Erfaringsdeling fra forarbeidet, P
- 11.30 – 12.30 Lunsj
- 12.30 – 13.50 Om kompetansemålene og programmering
- 13.50 – 14.00 Kaffe
- 14.00 – 15.15 Samskaping/forberedelse til mellomarbeid
- 15.15 – 15.30 Kort evaluering

# Erfaringsdeling fra mellomarbeid

- Hva har du prøvd ut siden sist? Hva har du lykkes med?  
Hva har du endret og hvorfor? (I)
- Organisering: Rekkeframlegg med anledning til å stille oppklarende spørsmål etter hvert fremlegg. 4 min per person inkludert oppklarende spørsmål. (G)
- Velg tidtaker og referent –
  - *tidtaker neste som har bursdag*
  - *referent er den yngste på gruppa*
- Lytt aktivt under fremlegg og spørsmål
- I spørsmålsrunden etter hvert innlegg skal dere ikke diskutere, bare stille spørsmål som oppklarer: Kan du forklare hva du legger i..?  
Tolker jeg deg riktig når du sa at...?
- **Oppdrag til gruppene: Velg en praksisfortelling fra gruppa. Forbered dere til å presentere denne i plenum. Praksisfortellingen må forholde seg aktivt til valgte kjerneelementer. (P)**

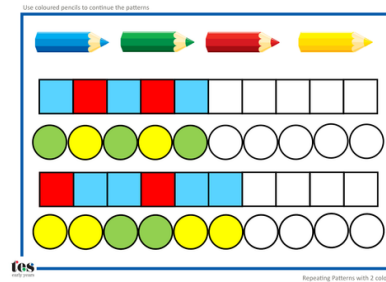
# Den digitale visningen av læreplanene

Den digitale visningen gir støtte til å se sammenhenger i og mellom fag, jobbe med progresjon og knytte opplæringen til overordnet del. Den har også forklaring av verb og lenker til ressurser. Se demo av læreplanvisningen under:

[Demo av læreplanvisning](#)

# Kompetansemålene med fokus på:

- Mønster og tallsystem
- Algoritmisk tenkning
- Programmering



# Lek, utforskning og undring i småskolen

## Underveisvurdering etter 2. trinn: (utdrag)

- Elevane viser og utviklar kompetanse i faget på 1. og 2. trinn når dei får **eksperimentere med** og **beskrive** ulike eigenskapar og strukturar i tal- og figurmønster i **utforskande leik, kunst og kvardagssituasjonar**.
- Elevane viser og utviklar òg kompetanse i matematikk når dei **undrar seg, stiller matematiske spørsmål** og **forklarer** og **argumenterer** for eigne løysingar.
- Læraren skal leggje til rette for elevmedverknad og stimulere til lærelyst ved at elevane får **utforske matematikk** gjennom å **bevege seg, leike, undre seg og bruke sansane**.

# Mønster og tallsystem i småskolen

Kompetansemål etter 2. trinn:

- kjenne att og beskrive repeterande einingar i mønster og lage egne mønster
- beskrive posisjonssystemet ved hjelp av ulike representasjonar



20-perlekjede



<https://www.mattelist.no/370>



100-perlekjede

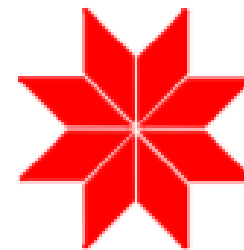
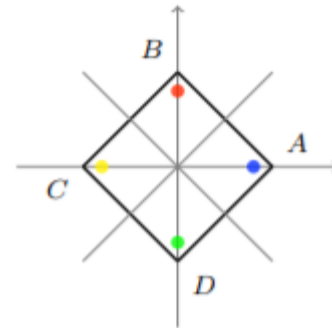
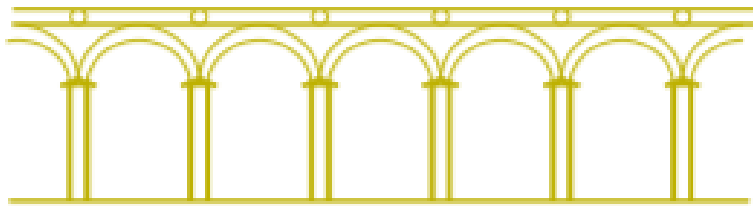




# Symmetri og mønster på mellomtrinnet

Kompetansemål etter 6. trinn:

- utforske og beskrive symmetri i mønster og utføre kongruensavbildingar med og utan koordinatsystem



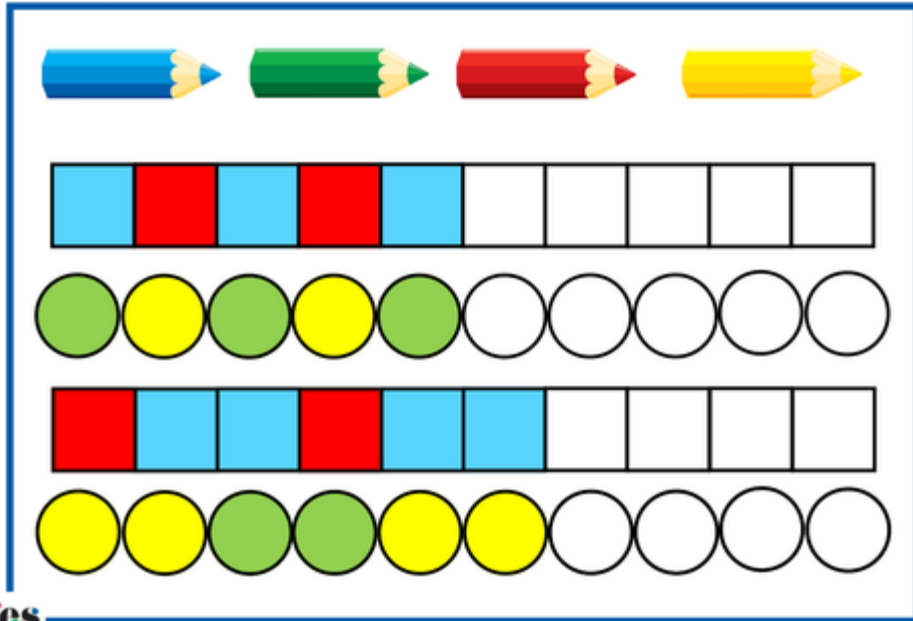
# Repeterende mønster

A: klapp  
B: knips

AAB-AAB-AAB

Kjør sekvensen!

Use coloured pencils to continue the patterns



ies  
early years

Repeating Patterns with 2 colours



Finn det minste repeterende mønster i figurene over  
 I figuren til høyre kan du velge om farge er vesentlig

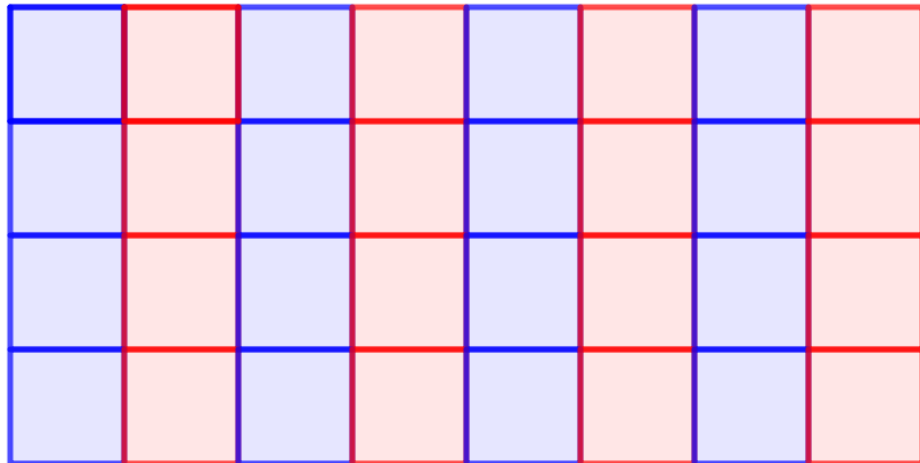
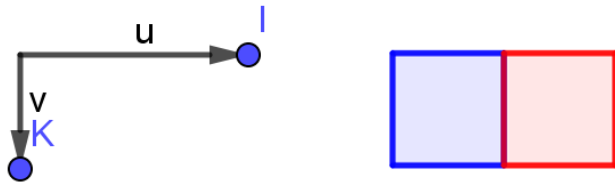
|  |  |  |
|--|--|--|
|  |  |  |
|--|--|--|



# Repeterende møster (båndmønster)



# Repeterende mønster i x- og y-retning



[Lenke til GGB-fil](#)

Klikk og dra i pilspissene og se hva som skjer



# Repetrende mønster

- Se verdien av gjenta løkker
- <https://scratch.mit.edu/projects/357169760/editor/>

The image shows a Scratch code editor with the following structure:

- When clicked:** A loop block labeled "gjenta 4 ganger" (repeat 4 times) containing:
  - Block 1: "spill lyden BB2 til den er ferdig"
  - Block 2: "spill lyden BB1 til den er ferdig"
  - Block 3: "spill lyden BB1 til den er ferdig"
  - Block 4: "spill lyden BB2 til den er ferdig"
  - Block 5: "spill lyden BB1 til den er ferdig"
  - Block 6: "spill lyden BB1 til den er ferdig"
  - Block 7: "spill lyden BB2 til den er ferdig"
  - Block 8: "spill lyden BB1 til den er ferdig"
  - Block 9: "spill lyden BB1 til den er ferdig"
  - Block 10: "spill lyden Mjau til den er ferdig"
- When clicked:** A "til lett" (until loop) block containing:
  - Block 1: "Repeterende mønster er BB2, BB1, BB1"
- When clicked:** A "Fra tungvint..." (from heavy) block.

# Programmering?

Veldig forenklet kan man si at programmering er en oppskrift du kan gi til datamaskinen, som den utfører til punkt og prikke. Datamaskinen utfører instruksene (oppgavene) i oppskriften en etter en, akkurat slik de står skrevet, i nøyaktig den rekkefølgen du har skrevet dem.



# Nøkkelelementer i programmering

- **Variabler:**  $n = 10$ ,  $k = n + 1$   
Sett  $n$  til verdien 10, sett  $k$  til verdien som er 1 høyere enn  $n$
- **Vilkår:** Hvis(test(ja,nei); gjør noe; gjør noe annet)  
Hvis(Jente; gå til høyre; gå til venstre)  
Merk: Ja, nei er ofte erstattet med 1,0 eller true, false
- **Løkker:** To typer:
  1. gjenta  $n$  ganger følgende:
  2. Inntil test er oppfylt gjør følgende:
- **Funksjoner:** Utfører en prosedyre, for eksempel finner felles nevner  
En funksjon (black box) returnerer et tall, en tekst, sann m.m.

# Introduksjon til Scratch – litt terminologi

The image shows the Scratch programming environment with a project titled "Kattens lek med musen ell...". The main workspace is labeled "scriptområdet" and contains a script for a cat character. The script starts with a "når klikkes" event, followed by "gå til x: -150 y: 0", "gjenta for alltid" (repeat forever) loop containing "pek mot musepeker", "neste drakt", "vent .1 sekunder", "gå 10 steg", "hvis berører musepeker?" (if touching mouse pointer) block with "spill lyden 'Mjau' til den er ferdig" (play sound 'Mjau' until done), "si 'Endelig!' i 2 sekunder" (say 'Endelig!' for 2 seconds), "gli 1 sekunder til x: -150 y: 0" (slide 1 second to x: -150 y: 0), "pek i retning 90", "si 'En gang til :)' i 2 sekunder" (say 'En gang til :)' for 2 seconds), "stopp alle" (stop all), and "bytt drakt til drakt1".

Annotations on the right side of the script include:

- "Her starter katta" (Here the cat starts) pointing to the "når klikkes" block.
- "skaper animasjonen" (creates the animation) pointing to the "gjenta for alltid" loop.
- "For at det ikke skal gå for fort" (so it doesn't go too fast) pointing to the "vent .1 sekunder" block.
- "Hvis berøring..." (If touching...) pointing to the "hvis berører musepeker?" block.
- "...så si mjau, si endelig og stopp" (...so say meow, say end of it and stop) pointing to the "spill lyden 'Mjau' til den er ferdig" and "si 'Endelig!' i 2 sekunder" blocks.
- "Gli tilbake til start" (Slide back to start) pointing to the "gli 1 sekunder til x: -150 y: 0" block.
- "ellers fortsett lekka" (otherwise continue the game) pointing to the "si 'En gang til :)' i 2 sekunder" block.

Other interface elements include:

- "Klossliste" (Block List) on the left, showing categories like Bevegelse, Utseende, Lyd, Hendelser, Styring, Sansing, Operatører, Variabler, and Mine klosser.
- "scenen" (Stage) on the right, showing the cat character and the text "...musens lek med katten".
- "figurliste" (Sprite List) at the bottom right, showing "Figur1" (the cat) and "Bakgrunn" (Background).
- Stage properties: Figur1, x: 35, y: 29, Storrelse: 100, Retning: -56.



# Et eksempel: kattens lek med musen eller...

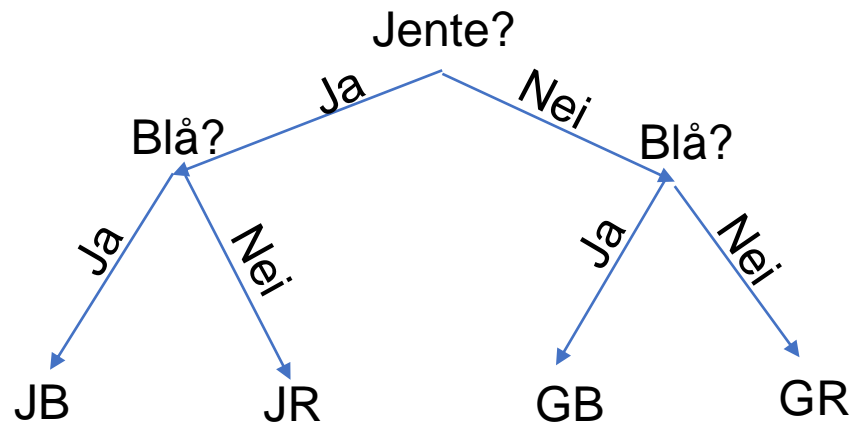
- Animasjon i fanen drakter med tegneverktøy, merk sentrum kan flyttes (rotasjon)
- Fargekoder på:

bevegelse utseende lyd  
hendelser styring sansing  
operatorer variabler mine klosser

# Programmering på småtrinnet

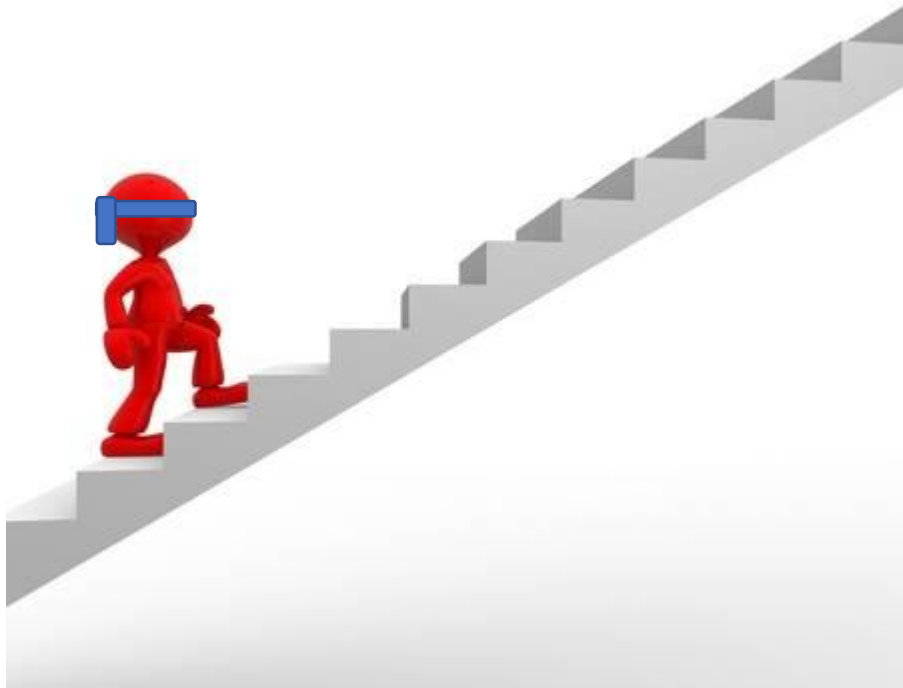
| Trinn | Kompetansemål  |
|-------|--|
| 2.    | - Lage og følge reglar og trinnvise instruksjonar i leik og spel.  |
| 3.    | - Eksperimentere med og forklare plasseringar i koordinatsystemet.   |
| 4.    | - Utforske og beskrive strukturar og mønster i leik og spel.<br>- Lage algoritmar og uttrykkje dei ved bruk av variablar, vilkår og lykkjer. |

Idéer: Doble **vilkår**: Jenter og gutter er fordelt i grupper blå og rød.



# Løkker

- **Inntil løkke (while-løkke):** Dytt foten forover, inntil du ikke møter motstand ta et trinn opp; stopp og si: «jeg har nådd toppen»



**Gjenta løkke (for-løkke):** Antall trinn kjent

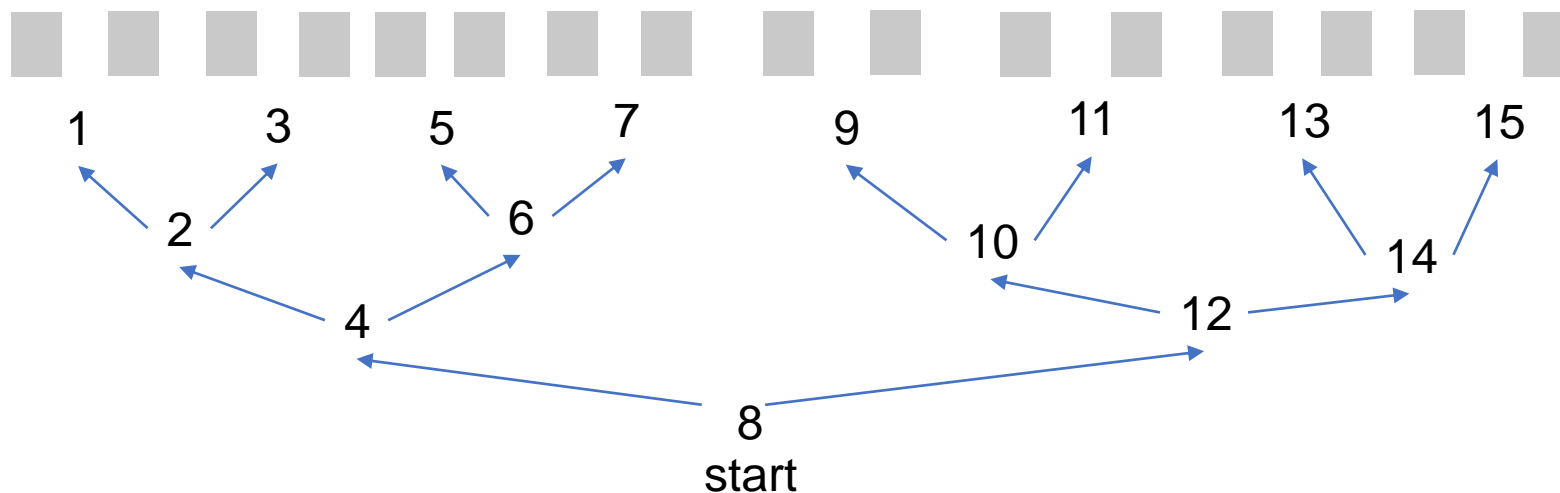
Gjenta 13 ganger: ta et trinn opp;  
stopp og si: «jeg har nådd toppen»

# Programmering på mellomtrinnet

| Trinn | Kompetansemål   |
|-------|---|
| 5.    | - Lage og programmere algoritmar med bruk av variablar, vilkår og lykkjer   |
| 6.    | - Bruke variablar, lykkjer, vilkår og funksjonar i programmering til å utforske geometriske figurar og mønster.   |
| 7.    | - Lage og vurdere budsjett og rekneskap ved å bruke rekneark med cellereferansar og formlar<br>- Bruke programmering til å utforske data i tabellar og datasett |

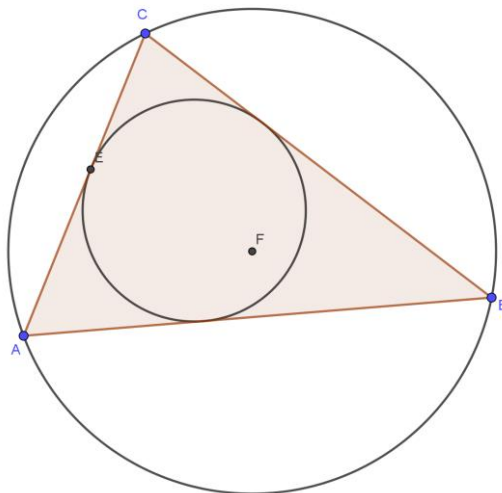
Idéer: sorteringsalgoritme.

Har du større tall enn det du er på– ta til høyre-ellers venstre



# Programmering mellomtrinnet forts.

- Programmering i Excel:  
[Simuler](#) 120 terningkast og lag en frekvenstabell.
- Programmering i GeoGebra:  
Lag et verktøy som finner omsirkelen og innsirkelen i en vilkårlig trekant
- Lag en klokke i Scratch



# Programmering på ungdomstrinnet

| Trinn | Kompetansemål  |
|-------|--|
| 8.    | <ul style="list-style-type: none"> <li>- Representere funksjonar på ulike måtar og vise samanhengar mellom representasjonane</li> <li>- Utforske korleis algoritmar kan skapast, testast og forbetrast ved hjelp av programmering</li> </ul> |
| 9.    | <ul style="list-style-type: none"> <li>- Simulere utfall i tilfeldige forsøk og berekne sannsynet for at noko skal inntreffe ved å bruke programmering</li> </ul>  |
| 10.   | <ul style="list-style-type: none"> <li>- Utforske og samanlikne eigenskapar ved ulike funksjonar ved å bruke digitale verktøy</li> <li>- Utforske matematiske eigenskapar og samanhengar ved å bruke programmering</li> </ul>                |

Idéer: Lage algoritme i Excel for å finne  $\text{sff}(a, b)$  (største felles faktor)

Programmere billøp i scratch. Trekk to biler til en tilfeldig x-pos og lag programmet slik at bilene kommer likt i mål

Ekstra utfordring, la den ene bilen gå med konstant fart, den andre starter med null fart, men er akselerert

# Algoritmisk tenkning

- Se på verbene som er brukt og sammenlikne med LK20



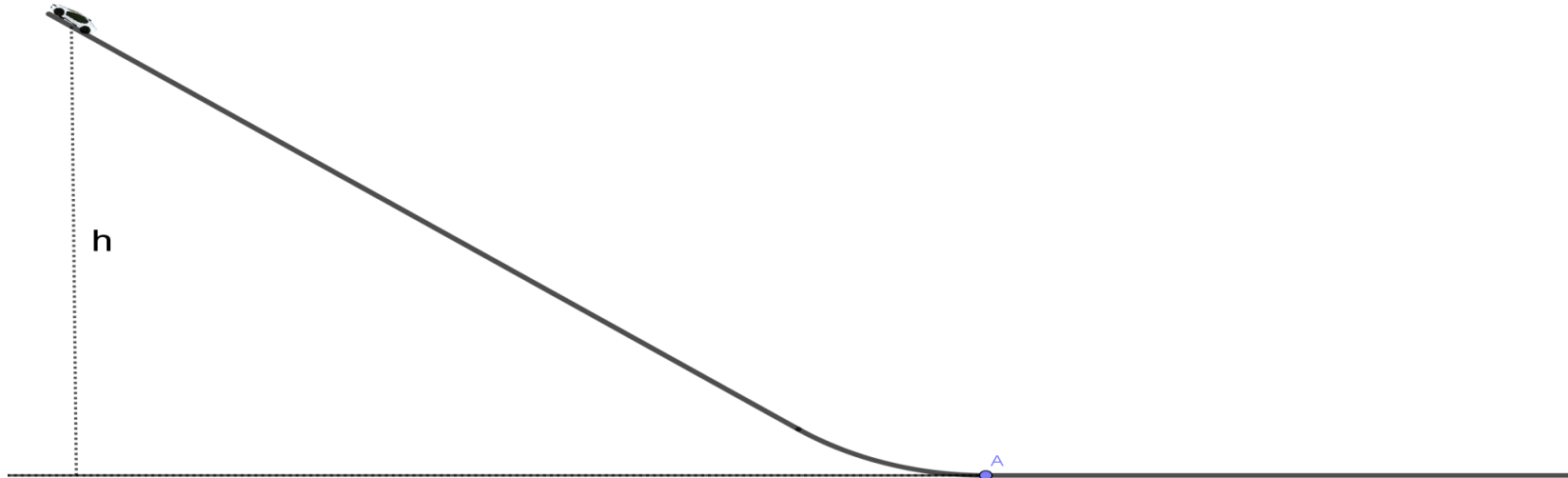
# Hva er algoritmisk tenkning?

## En problemløsningsstrategi

- Formulere og forstå problemet
- Teknologi og/eller manuelt?
- Skill ut vesentlig informasjon fra uvesentlig
- Splitt og hersk (del opp problemet i mindre problemer)
- Prøve og feile i fellesskap – mot en modell
- **Lage** (ikke følge) en oppskrift (algoritmer, programmering)



# Eksempel med bilbane



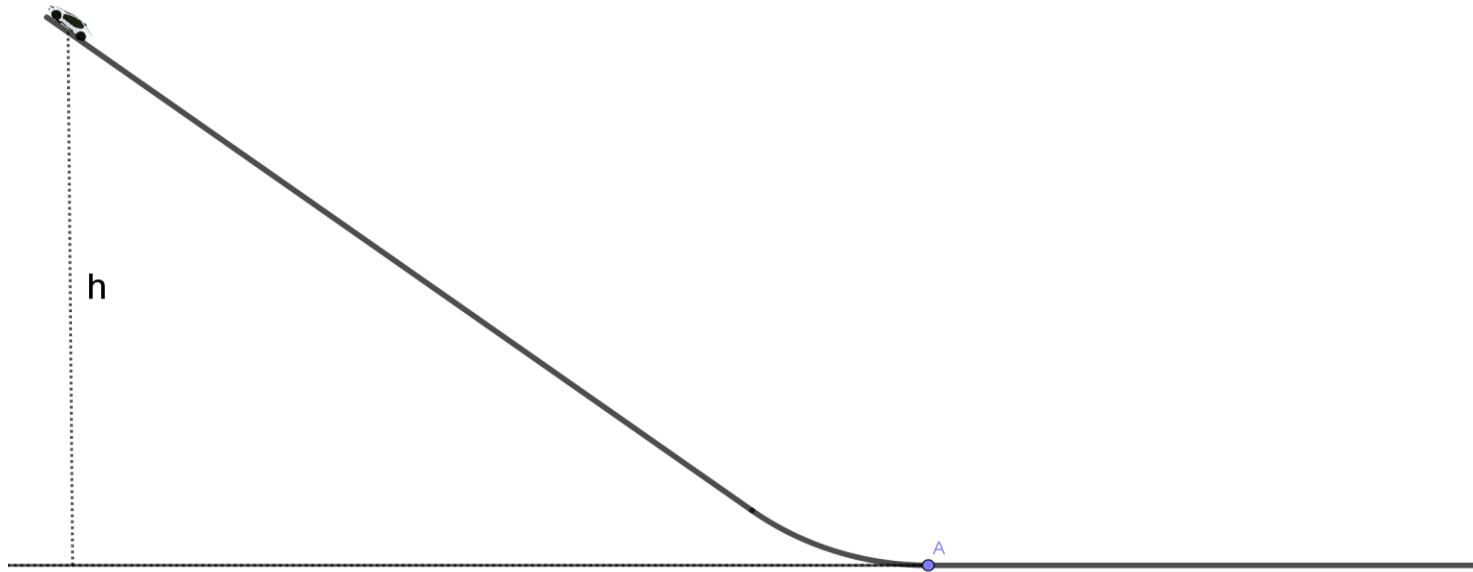
En lekebil slippes på et skråplan.

Lag en problemstilling som du mener er innenfor rekkevidde for en elev på

4. trinn

7. Trinn

10. trinn



- Formulere og forstå problemet
- Teknologi og/eller manuelt?
- Skill ut vesentlig informasjon fra uvesentlig
- Splitt og hersk (del opp problemet i mindre problemer)
- Prøve og feile i fellesskap – mot en modell
- **Lage** (ikke følge) en oppskrift (algoritmer, programmering)

# Forslag 4. trinn

- **Formulere og forstå problemet** – Hvordan går det med lengden bilen ruller når høyden øker?
- **Teknologi og/eller manuelt?** – Måling manuelt/laser, tabell, graf i excel
- **Skill ut vesentlig informasjon fra uvesentlig** (farge, høyde, tyngde, rett bane, teppe/glatt gulv)
- **Splitt og hersk** (del opp problemet i mindre problemer) Doble/ halvere
- **Prøve og feile** i fellesskap – mot en modell. Da går den kanskje midt imellom om vi slipper fra...
- **Lage** (ikke følge) **en oppskrift** (algoritmer, programmering) lengden er ca 3 ganger høyden

# Forslag 10. trinn

- **Formulere og forstå problemet** –  
Lag en funksjon som viser rullelengden som funksjon av høyden ved forskjellige underlag
- **Teknologi og/eller manuelt?** –  
Måling manuelt/laser, tabell, graf manuelt, kurvetilpassing i Excel.
- **Skill ut vesentlig informasjon fra uvesentlig**  
(farge, høyde, tyngde, rett bane, teppe/glatt gulv, banens krumning)
- **Splitt og hersk** (del opp problemet i mindre problemer)  
Holde underlaget konstant, variere høyden.
- **Prøve og feile** i fellesskap – mot en modell. –  
Det kan jo ikke stemme, for den går kortere på teppet...
- **Lage (ikke følge) en oppskrift** (algoritmer, programmering)  
 $L(h) = kh$  eller  $L(h, \mu) = ch/\mu$

# Samskaping/forberedelse til mellomarbeid 14.00 – 15.15

- Snakk sammen i gruppa om hvordan planene dine i matematikkfaget fra oktober – desember harmonerer med fagets nye sentrale verdier og kjerneelementer. Hva bør endres slik at planene harmonerer bedre med det nye?
- Arbeid med konkrete opplegg som kan prøves ut før neste samling, som involverer to eller flere kjerneelementer og som er i samsvar med faget nye sentrale verdier
- NY TEKST SKAL SKRIVES

# Mellomarbeid og forarbeid (før 23/3)

A: Enkeltlærere ser over planer og prøver ut opplegg som involverer to eller flere kompetansemål og er samsvar med fagets nye sentrale verdier.

B: Skolebasert erfaringsutveksling – Presenter fagets kompetansemål for kolleger i eget og andre fag på egen skole. Eksemplifiser med erfaringene fra utprøving.

C: Forarbeid til neste samling. Undersøk den vedtatte nye læreplanen i matematikkfaget. Begynn med det mest aktuelle klassetrinnet for deg og se etter de lange linjene før og etter dette trinnet. Hva gleder du deg til og hva er utfordrende?