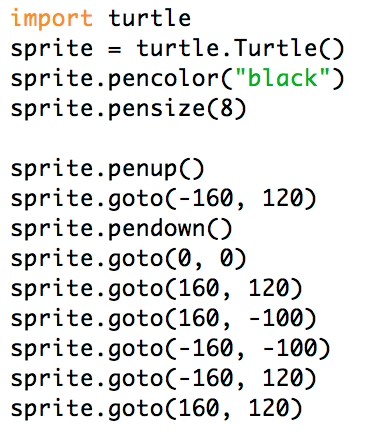
**Opgavetype #5: Omsæt koden**

Elev-handout

Programmet nedenfor er lavet i IDLE, som er Python’s software applikation, hvor du kan lave små programmer, bl.a. i Turtle Graphic, som du kan bruge til at få en skildpadde til at tegne geometriske former eller andet. Alle tallene i parentes fra linje 6-13 er X og Y koordinater. Kig på koden. Kan du regne ud, hvad den gør? Lav evt. selv et koordinatsystem på et stykke millimeterpapir først, og indtegn kommandoerne.



Når du tror, du har svaret, skal du åbne Scratch. Prøv nu at lave det samme program her.

Tegner din sprite den figur, som du forestillede dig?

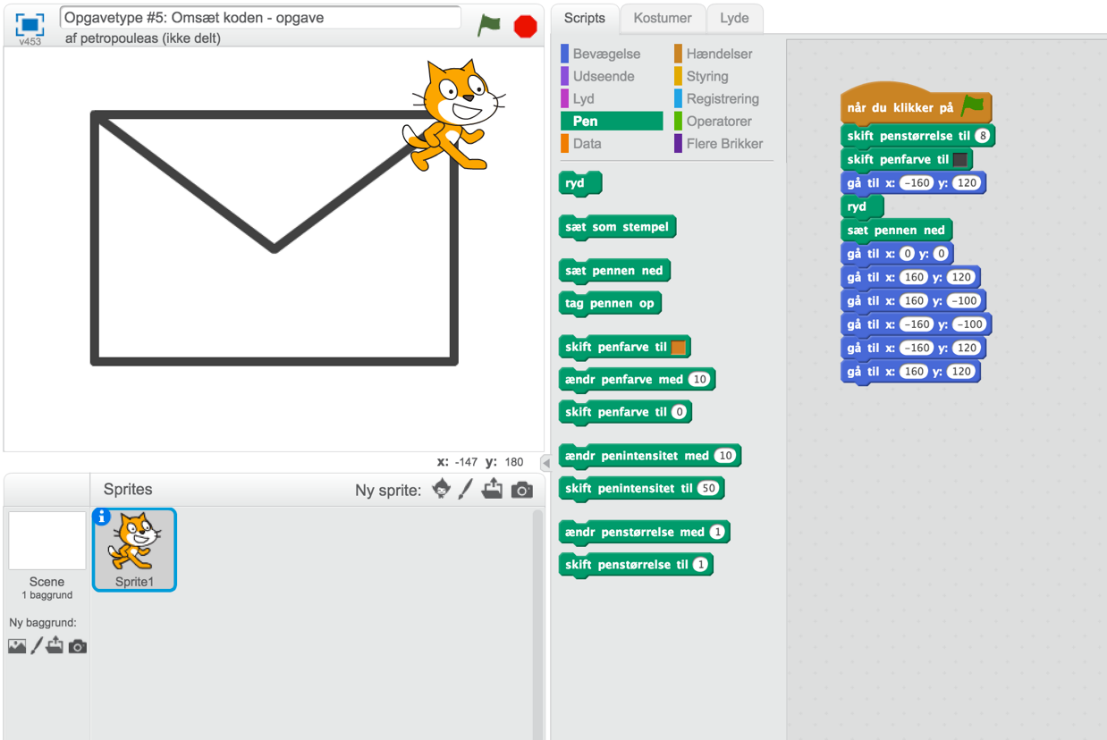
Husk, at i Scratch ryddes din scene ikke automatisk, når du genstarter programmet, så du er også nødt til at tilføje kommandoen ”ryd”.

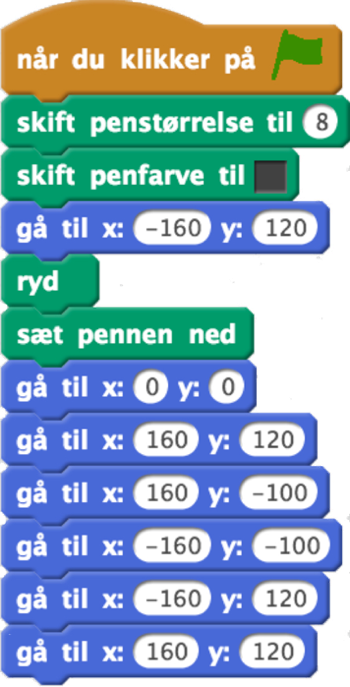
**Til underviseren**

Opgavetype #5: Omsæt koden

**Løsning**

Programmet tegner en konvolut. Vær dog opmærksom på, at i Scratch skal eleverne huske at tilføje kommandoen ryd. Til gengæld behøver de så ikke kommandoen tag pennen op, da ryd visker alle streger ud (også den, der tegnes, når spriten skal placere sig på x -160, y 120 fra startpositionen x 0, y 0), inden selve konvolutten tegnes. Programmet afvikles øjeblikkeligt. Det kan være en god idé at indsætte vent-kommandoer imellem de øvrige, så eleverne bedre kan se, hvad hver kommando gør.





**Hvad eleven bør vide på forhånd**Eleven skal kunne åbne Scratch og udskifte sprites, samt vide, hvordan de forskellige kodeblokke er organiseret, og hvordan de sættes sammen og fjernes igen. De skal også vide, at de selv kan ændre på tallene i de hvide cirkler, som indgår i nogle af kodeblokkene, og hvor det grønne flag, som aktiverer programmet, er. Eleverne bør også vide, hvordan Scratch anvender et koordinatsystem til at placere sprites’ene, så de forstår de mange X- og Y-angivelser, og de skal kende penne-værktøjet på forhånd.

**Beskrivelse af opgavetype #5: Omsæt koden**

Her skal eleverne omsætte det samme program fra et programmeringssprog til et andet, eksempelvis:

* fra tekstsprog til visuelt sprog
* fra visuelt sprog til tekstsprog
* fra et tekstsprog til et andet tekstsprog.

Det første er ofte det nemmeste for eleverne, da de tit kender det visuelle sprog, de skal omsætte til, på forhånd. Da tekstsproget, som de skal omsætte fra, ofte er nyt for eleverne, skal de til gengæld hjælpes med at forstå, hvad programmet egentlig gør, før de kan omsætte det til et visuelt sprog. Der er altså fokus på læsning og forståelse af et endnu ukendt programmeringssprog.

I bevægelsen fra visuelt sprog til tekstsprog afhænger sværhedsgraden selvfølgelig af, hvor komplicerede programmer, der skal omsættes. Bevægelsen fra visuelt sprog til tekstsprog giver mulighed for at introducere semantiske og syntaktiske regler i tekstsproget i små bidder.

Det tredje forslag kan virke meget udfordrende og kan først gennemføres, når eleverne er fortrolige med ét tekstprogrammeringssprog. Da programmeringssprogene ligner hinanden, er opgaven dog ikke helt så kompliceret, som den kunne lyde.

Det er i øvrigt muligt at bruge opgavetypen til nybegyndere, hvis der i klassen er arbejdet med symbolsk repræsentation, eksempelvis pile eller pseudokoder forinden. Her vil opgaven så handle om at omsætte den symbolske repræsentation eller pseudokoden til et visuelt programmeringssprog.

***Didaktiske principper:*** *#2: Der tages udgangspunkt i et program, der virker. #5: I omsætningen konkretiseres en abstrakt idé.*

***Computational Thinking: logisk ræsonnement, mønstre og generaliseringer, evaluering.***

***SOLO-taksonomi:*** *relationelt niveau. Eleven kan se, hvordan det samme program kan udformes i forskellige programmeringssprog.*