

Indhold

Indledning	2
CFU og kodning i undervisningen	2
Læringsmål –og programmeringsmål	2
Tegn på læring	3
Programmet Scratch	4
Hjemmesiden	4
PC eller iPad?	5
Inden I går i gang	5
Lektioner	6
Scratch	6
Funktioner	7
Opgave 1. Funktionsmaskinen og variable.....	7
Opgave 2. Funktionsmaskinen og flere variable	8
Opgave 3. Sammenhænge og mange Sprites.....	8
Opgave 4. Sammenhænge med flere Sprites	8
Opgave 5. Sammenhænge og priser	9
Opgave 6. Sammenhænge og proportionalitet.....	9
Opgave 7. Koordinatsystemet og funktioner	9
Opgave 8. Koordinatsystemet og funktioner – figurer og mønstre	10
Geometri	10
Opgave 1. Geometri og trekanter	10
Opgave 2. Parallelforskydning	11
Opgave 3. Mange kanter	11
Opgave 4. Polygoner.....	11
Opgave 5. Omkreds og areal	12
Opgave 6. Tegn en fodboldbane	12
Opgave 7. Trekantens areal.....	12
Analog programmering	13
Opskrifter.....	13
Robotten på lageret.....	13
Evaluerings	13

Indledning

Præstelundskolen i Ikast-Brande kommune har ansøgt om midler til projektet gennem Undervisningsministeriets udlodningsmidler. Tanken har været, at den logik, struktur og metodik der findes inden for programmering må korrespondere godt med mange matematiske problemløsningsstrategier.

Projektet er et forsøg på at sætte programmering i fokus, i forbindelse med den generelle undervisning i matematik faget. Igennem lektionerne vil eleverne arbejde både færdigheds –og problemorienteret, samt lære at bruge et programmeringsværktøj, fx Scratch. Der er desuden mulighed for arbejde med analog programmering igennem nogle specifikke opgaver og øvelser.

CFU og kodning i undervisningen

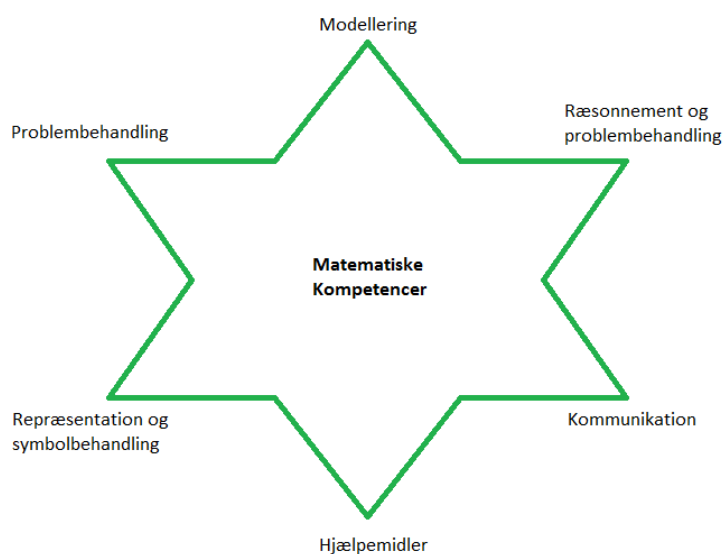
I VIA CFU arbejdes på at implementere kodning i undervisningen. Begrebet kodning skal forstås som en måde at arbejde med computer forståelse. Det handler altså ikke om, at eleverne alene skal være gode til at skrive kode, men i højere grad at få begreb om computere og hvordan de fungerer i en verden påvirket af computere i stigende grad, samt at skabe interesse for de mange muligheder som forståelse for computersprog følger med sig. På [VIA CFUs hjemmeside](#) kan man følge med i aktuelle projekter og bl.a. se bud på mål for kodning i undervisningen.

På siden findes også en oversigt over programmer og apps som kan bruges i forbindelse med kodning. Her ses det, at fx Scratch er vurderet som velegnet til brug med 3.-9.klassetrin.

Læringsmål –og programmeringsmål

Til hver lektion er der opstillet lærings –og programmeringsmål. Intentionen er, at eleverne skal kunne forstå målene og bruge dem som rettesnor for arbejdet med indholdet. Derfor er målene formuleret i forhold til det faglige niveau i 6.klasse. Læringsmålene er formuleret i forhold til de nye forenklede fælles mål for faget matematik.

Lærervejledning



Kilde: [Undervisningsministeriet](#)

På figuren ses de seks færdigheds –og vidensområder som læringsmålene er centreret om. Alle områder er tænkt ind i forløbet, men særlig vægt har områderne "Problembehandling" og "Ræsonnement og problembehandling" fået.

Under "Forslag til færdigheds -og vidensmål" kan man se en oversigt over trinmål der har været udgangspunkt for lektionerne, markeret med rødt.

Det er oplagt at arbejde med at opstille læringsmål og derefter vælge af aktiviteterne, så aktiviteterne understøtter målene.

Tegn på læring

Læreren kan vurdere elevernes programmer formativt og summativt og dermed evaluere læringen. Dette gøres ved at vurdere det matematiske indhold i de elevproducerede programmer i forhold til de opstillede læringsmål. For at opnå et tilfredsstillende billede af elevernes læring, skal læreren afsætte tid til kommunikation med eleverne og stille spørgsmål i forhold til elevernes forståelse. Hvis valget af opgaver stemmer overens med elevernes faglige niveau i matematik, så vil det sandsynligvis frigive tid til læreren, som fx kan anvendes til den løbende evaluering.

Eksempler på tegn:

- Eleven anvender de korrekte programmerings "klodser".
- Eleven viser forståelse for betydningen af at ændre værdier i "klodserne".

Lærervejledning

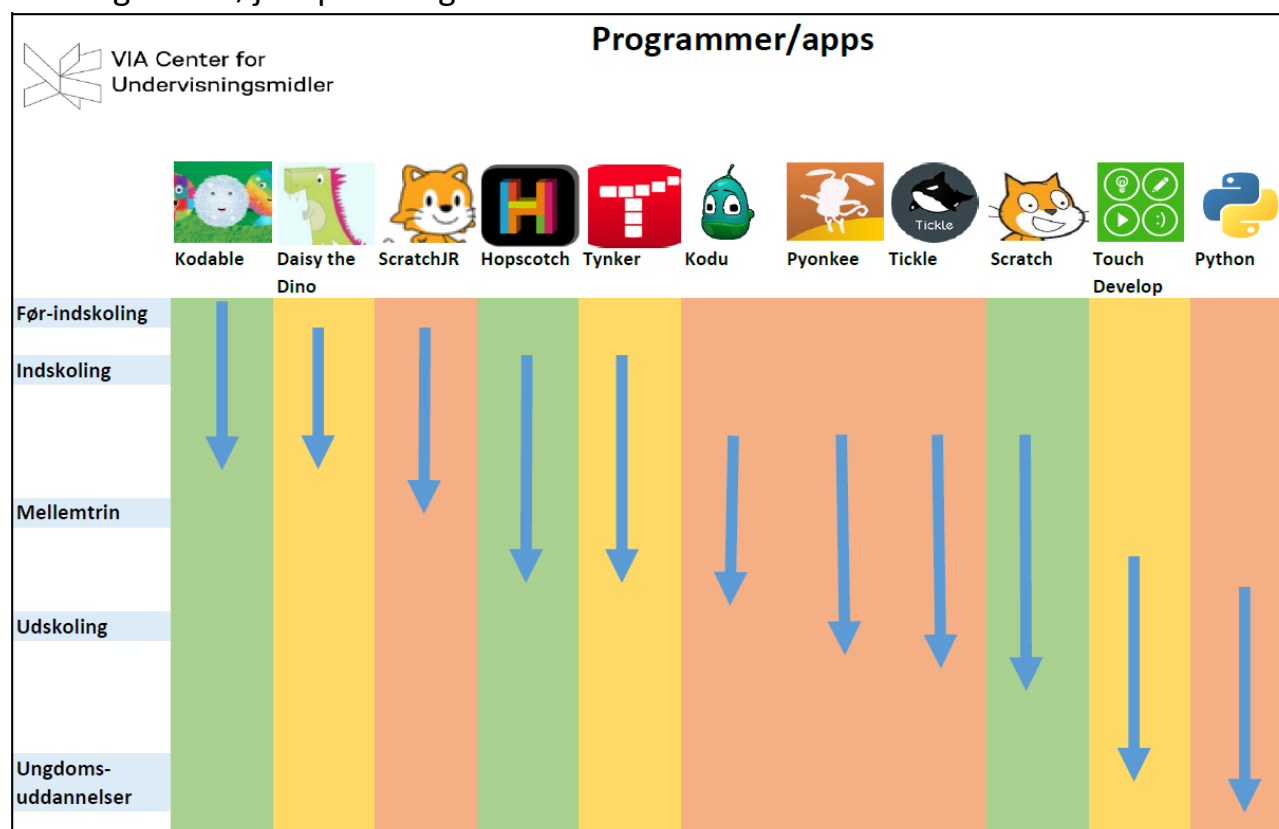
- Eleven løser opgaverne og videreudvikler programmerne, altså laver dem mere avancerede.
- Eleven kan simplificere programmer, altså bruge færre klodser til den samme handling.

Programmet Scratch

"Scratch er et programmeringssprog og et online fællesskab hvor du kan lave dine egne interaktive historier, spil og animationer – og dele dine ting med andre i hele verden. Under processen med at designe og programmere, lærer børn at tænke systematisk og at arbejde sammen med andre".

Kilde: Scratch.mit.edu

En af de særlige fordele med Scratch, i forhold til andre gratis programmeringsværktøjer er, at programmet og sitet er oversat til dansk. Dette åbner for brug i indskolingen. VIA CFU har lavet en niveaudelt oversigt af forskellige kodningsværktøjers placering i forhold til hinanden.



Hjemmesiden

Til projektet er der blevet oprettet en hjemmeside,

<http://programmeringimatematik.dk/>, som indeholder de nødvendige informationer

Lærervejledning

i forhold til projektet. Her er der fx adgang til lektioner, inspirationsvideoer og en blog hvor repræsentative elevproduktioner deles.

På hjemmesiden under inspiration findes [instruktionsvideoer](#) til farvekoderne i Scratch, som både lærere og elever kan benytte, hvis de har behov for hjælp. Videoerne er korte og forklarer en lille del af anvendelsesmulighederne af de forskellige farver koder i Scratch.

Blå	Bevægelse(gå, dreje)
Brun	Startknapper(fx start af program med specifik tast)
Grøn	Pen(starte og stoppe tegnefunktion)
Lilla	Figur udseende(størrelse, tale)
Lyseblå	Registrering(når en figur rører en anden så....)
Lysegrøn	Operationer(fx definere et værdiområde for bevægelse)
Magenta	Lyd
Orange	Styring(Gentage bevægelser, stoppe sekvenser/bevægelser)

PC eller iPad?

Der kan benyttes både PC og ipad, til at løse opgaverne. Vælger man at bruge iPad er det fx gratisappen "Pyonkee", der skal benyttes. Appens interface er næsten identisk med det der bruges i Scratch. En sammenligning af de to platformes mulighed kan være gavnligt , som her er stillet op i en tabel:

	Scratch/PC	Pyonkee/iPad
Interface	Muligt at variere størrelse af fx arbejdsfelt.	Interface er låst i størrelsen og arbejdsfeltet er fx lille.
Gem	Gemmes automatisk i skyen. God oversigt.	Kan gemmes på iPad.
Styring	Især med mus er styringen hurtig og responsiv.	Dobbelt tap for at vælge ny kommando, ellers ok.
Deling	Kan deles let på diverse sociale medier.	Umiddelbart få muligheder for deling.

Inden I går i gang

Inden arbejdet med lektionerne påbegyndes, er der nogle foranstaltninger det anbefales, at læreren er opmærksom på.

Lærervejledning

- Hver elev skal have en computer når man starter med lektioner. Dvs at eventuelle problemer med anskaffe en computer, skal være løst umiddelbart inden.
- Lektionernes trin skal følges. Lader man eleverne springe i trinnene tidligt i forløbet, vil det sandsynligvis forøge frekvensen af spørgsmål til læreren.
- En introduktion af kodning/programmering, så eleverne får en idé om hvorfor det er vigtigt. Computerspil og film er oplagte som eksempler.

Lektioner

I begge forløb, funktioner og geometri, er der tænkt progression i forhold til sværhedsgrad. Det fungerer sådan, at den første opgave er lettere end den næste osv. Emnet funktioner er desuden generelt lettere end emnet geometri. Ud over inspirationsvideoerne til farvekoderne i Scratch, er der også inspirationsvideoer i forhold til løsningen af udvalgte opgaver. Videoerne har tilsammen det formål at lette arbejdsbyrden for læreren i opstarten af forløbet.

Nederst på hver opgave er der udfordringer. Udfordringerne er tænkt som muligheder for at fremme dialogen mellem eleverne og differentiere undervisningen. Farven lilla symboliserer de letteste opgaver og rød de sværeste, og begge er fortrinsvis matematisk fagligt funderede. Den grønne farve repræsenterer innovative udfordringer, som handler om at skulle skabe noget. Det skabte kan være en idé eller et særligt produkt. De lyseblå udfordringer er produktorienterede, og har til formål at gøre eleven skabende. Skabelsen af multimodale produkter, kan gøre eleverne engagerede og motiverede, bl.a. fordi det giver ejerskab at producere, samt at eleverne møder forskellige modaliteter i deres hverdag. Hvis eleverne fx skal kunne screencaste, kan det anbefales, at de har stiftet bekendtskab med den måde at arbejde på, inden forløbet.

Lette
Svære
Produktorienterede
Innovative

Scratch

Har man som lærer ikke anvendt Scratch før, kan man begynde med disse lektioner og dermed få oprettet sine elever. Det er meningen, at lærer og elever sammen lærer at bruge Scratch, og det derfor ikke er nødvendigt at lærerne gennemgår alle

disse lektioner på klassen, i mindste detalje. Det kan selvfølgelig være en god idé at starte timen med sammen at læse de første linjer af den næste opgave. Desuden kan eleverne få en simpel indføring i Scratch, hvis de bruger ”trin-for-trin introen” under hjælp. På denne måde bliver læreren mindre afhængig af, at skulle have sat sig detaljeret ind i brugen af programmet på forhånd. I denne kategori findes der også hjælp til oprettelse af det første projekt, og en vigtig funktion som at nulstille scenen i programmet.

Funktioner

Lektioner til emnet er skrevet ud fra fagmålene som er blevet omsat til konkrete læringsmål. Meningen med dette er, at eleverne skal forstå målene og kunne bruge dem i forhold til målrettet arbejde med lektionerne. Følgende er et eksempel på hvordan man kan lave/opstille læringsmål:

Mål, voksensprog	Mål, elevsprog
Du skal lære at forstå og anvende variable	Du skal lære, at tal har en fast værdi
Du skal forstå sammenhænge mellem tal	Du skal lære, at bogstaver kan have forskellige værdier
	Du skal lære, at tal sammen med et andet tal kan ændre værdi

Det anbefales, at eleverne har repeteret brugen af koordinater i koordinatsystemet og en grundlæggende forståelse af funktioner, inden forløbet startes.

Til hver opgave er der et løsningsforslag til den første del.

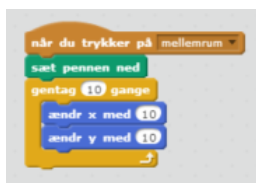
Opgave 1. Funktionsmaskinen og variable

Niveauet af opgaven, både matematisk og programmeringsmæssigt, er lavet så eleverne kan lave opgaven direkte efter at være blevet registreret og præsenteret for Scratch. De bliver ført igennem brug af forskellige basale kommandoer i programmet, og skal derfor ikke bruge særlige kræfter på at sætte sig ind i det. Det er vigtigt at instruktionerne følges nøje, så eleverne får brugt de nævnte kommandoer, da de danner grundlag for udfordringerne, samt de næste opgaver. Har eleverne spørgsmål til kommandoer eller løsning, kan læreren med fordel henvise til inspirationsvideoerne. Billedet af programmet ved punkt 8 i opgaven, kan bruges som rettesnor for både lærere og elever, så en god progression i arbejdet opnås.

Lærervejledning

Opgave 2. Funktionsmaskinen og flere variable

Der er ligheder mellem opgave 1 og 2, men opgave 2 adskiller sig ved at have tolkningsspørgsmål og en produktorienteret opgave. Det forventes, at eleverne anvender mindst 2 modaliteter, når de laver et produkt.



Opgavens løsning kan se sådan ud:

Det kan være en god idé, at eleverne laver et program der flytter Sprite tilbage til



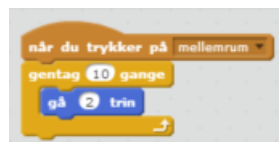
(0,0) og scenen samtidig ryddes.

Opgave 3. Sammenhænge og mange Sprites

Opgaven har en kortere indførselsopgave end 1 og 2, for at skabe progression i forhold til niveauet. Spørgsmålet der stilles inden udfordringerne, kan evt. diskuteres på klassen.



Program til Sprite1 kan se sådan ud:



Program til Sprite2 kan se sådan ud:

Opgave 4. Sammenhænge med flere Sprites

I denne opgave bruges begreberne input og output som direkte kan sammenlignes med x og y i sildebens modellen. Begreberne kan bruges som udgangspunkt for en klassesamtale om det bagvedliggende koordinatsystem i Scratch.



Programmet kan se sådan ud:

Lærervejledning

Opgave 5. Sammenhænge og priser

Der arbejdes som i opgave 4 med input og output. Udfordringen i opgave 5 skal ses som en slags afrunding af opgaverne 4 og 5, hvor sammenhænge og begreberne x og y har været centrale.



Programmet kan se sådan ud:

Opgave 6. Sammenhænge og proportionalitet

Fokus for opgaven er begrebet proportionalitet. Udfordringerne er tænkt som differentieringsmuligheder, i forhold til et normalvis besværligt begreb.



Programmet kan se sådan ud:



Scenen ser i så fald ud som følger:

Opgave 7. Koordinatsystemet og funktioner

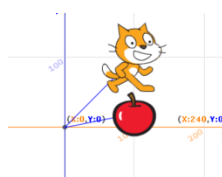
Opgaven har fokus på synligheden af det bagvedliggende koordinatsystem og kvadranter i Scratch, med støtte af den indsatte baggrund. Sværhedsgraden af opgaven er i den sammenhæng sænket i forhold til opgave 6.



og



Programmet kan se sådan ud:



Scenen ser i så fald ud som følger:

Lærervejledning

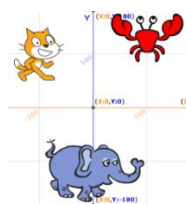
Opgave 8. Koordinatsystemet og funktioner – figurer og mønstre

Denne opgave hænger sammen med opgave 7 i den forstand, at de samme begreber og kommandoer benyttes, bortset fra en. Det kan være en god idé at lave den sidste udfordring, som en slags opsamling i forhold til elevernes forståelse af Scratch og koordinatsystemet.

Programmet kan se sådan ud:



og



Scenen ser i så fald ud som følger:

Geometri

I forhold til niveauet af programmering i opgaverne, kan det være nødvendigt at lave de første opgaver inden for emnet "Funktioner", da der ellers kan være mange begreber at håndtere. Generelt er sværhedsgraden i "Geometri" lidt højere end i emnet "Funktioner". I flere af opgaverne er der særligt fokus på, at eleverne skal kunne forklare deres fremgangsmåde og de begreber der benyttes.

Opgave 1. Geometri og trekanten

Første del af opgaven er introducerende i forhold til centraler kommandoer, når man skal tegne figurer i Scratch. Udfordringens formål er primært at træne disse centrale kommandoer.



Programmet kan se sådan ud:



Scenen ser i så fald ud som følger:

Lærervejledning

Opgave 2. Parallelforskydning

Opgaven har fokus på parallelforskydning som disciplin. I den første del af opgaven stifter eleverne bekendtskab med y som en variabel, i forhold til placering af figurer i et koordinatsystem.

Programmet kan se sådan ud:



og

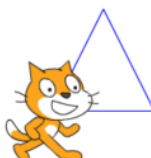


Scenen ser i så fald ud som følger:

Opgave 3. Mange kanter

Denne opgave skal ses som en forlængelse af opgave 1, med fokus på kommandoer til at skabe linjestykker og kanter i en figur.

Programmet kan se sådan ud:



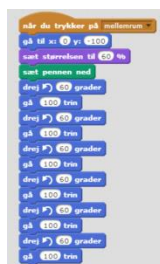
Scenen ser i så fald ud som følger:

Opgave 4. Polygoner

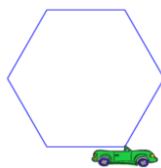
Opgaven har fokus på begreberne irregulære og regulære polygoner.

Udfordringerne har til formål at give eleverne synlige billeder af forskellene på de to slags figurer.

Programmet kan se sådan ud:



Lærervejledning



Scenen ser i så fald ud som følger:

Opgave 5. Omkreds og areal

Eleverne skal i denne opgave, bruge formlerne for firkantens omkreds og areal. Der er særligt fokus på elevernes til at vise hvad de laver og hvordan de gør.



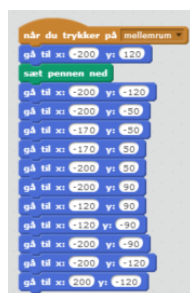
Programmet kan se sådan ud:



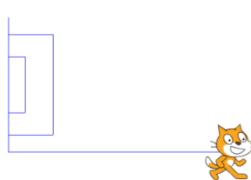
Scenen ser i så fald ud som følger:

Opgave 6. Tegn en fodboldbane

I opgaven skal eleverne bruge færdigheder fra de forrige opgaver. For de fleste elever vil det være en tidskonsumerende opgave.



Første del af programmet kan se sådan ud:



Scenen ser i så fald ud som følger:

Opgave 7. Trekantens areal

Igen er den mundtlige dimension i fokus, hvor eleverne skal forklare hvordan man finder arealet af en trekant, v.h.a. en formel.

Lærervejledning



Programmet kan se sådan ud:



Scenen ser i så fald sådan ud:

Analog programmering

Eleverne arbejder i disse opgaver med at opstille algoritmer. Målet med opgaverne er at give elever en forståelse for nødvendigheden af nøjagtighed i programmering og en simpel forståelse af begrebet debugging (fejlrretning). Desuden vil bevægelsesopgaverne være et afbræk i forhold til det stillesiddende arbejde med Scratch og give en hands-on oplevelse af arbejdet med algoritmer og programmering.

Opskrifter

Almindelige madopskrifter er et godt udgangspunkt til forståelse af algoritmer. Konkrete og specifikke ingredienser skal behandles og anvendes i en bestemt rækkefølge for at opnå et ønsket resultat.

Robotten på lageret

Bevægelse er i centrum for disse opgaver, hvor eleverne får visualiseret hvordan konkrete kommandoer kan udmønte sig i bevægelser. Ved at sætte system i bevægelser af arme og ben, bliver algoritmerne synlige og fejlrretning lettere. Få konkrete materialer er desuden nødvendige for et godt resultat.

Evaluering

Evaluering bør være en løbende proces, hvor læreren observerer og stiller relevante individuelle spørgsmål, i forhold til både de matematiske læringsmål og læringsmålene til programmering. Ud over den løbende dialog kan læreren fx lave en quiz med spørgsmål som er relevante i forhold til indholdet. Et bud på gratis værktøj til produktion af quiz, kunne være testmoz.com. En kort vejledning til anvendelse af værktøjet, findes [her](#).