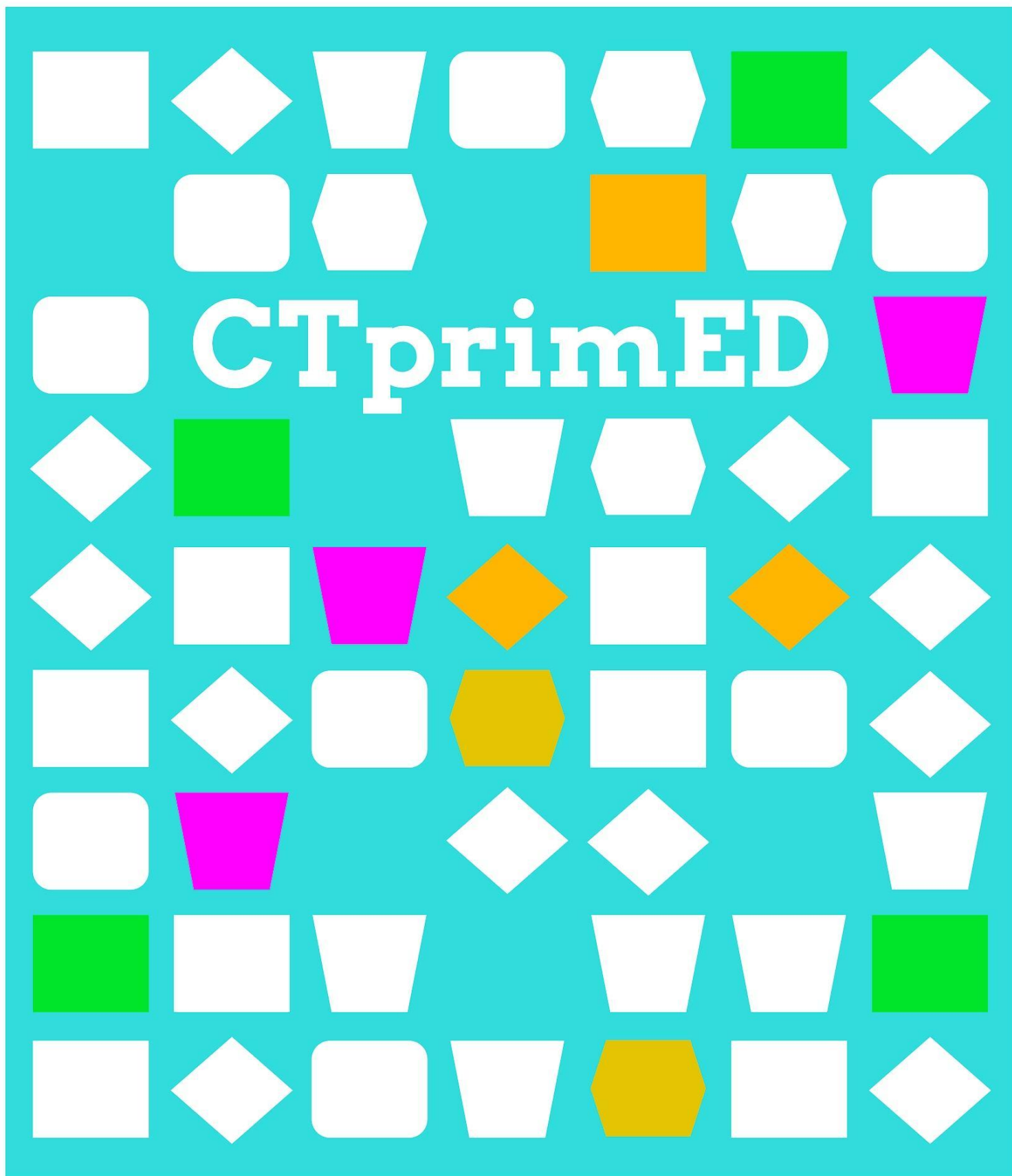
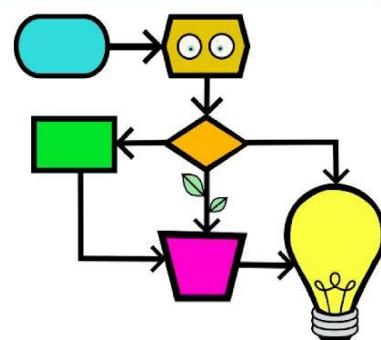
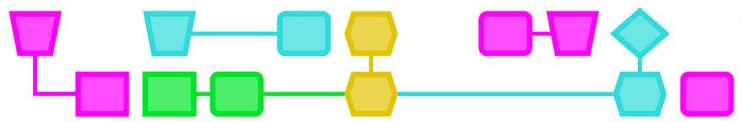




Programming –
Offline programming



Co-funded by the
Erasmus+ Programme
of the European Union

**Sammendrag:**

En algoritme kan forstås som trin-for-trin-instruktioner til, hvordan man løser et problem. Tænk på opskrifter, et manuskript til et skuespil eller et program til en computer. I denne lektion lærer eleverne, hvad en algoritme er, ved at deltage i et spil, som viser dem, hvilke instruktioner en computer kan forstå. Efterfølgende skal de programmere hinanden som dansecomputere.

Målgruppe: 6-12 år (der er tilføjet differentiering for børn, der har svært ved at læse og skrive.)

Varighed: 55 minutter

Læringsmål: Efter undervisningen vil eleverne:

- kende til, hvad en algoritme er
- have lært hvad en algoritme kan bruges til i hverdagen
- have lært, hvad en løkke og en betingelse er, og hvordan man bruger dem i et program

Online/offline: offline.

Computational Thinking:

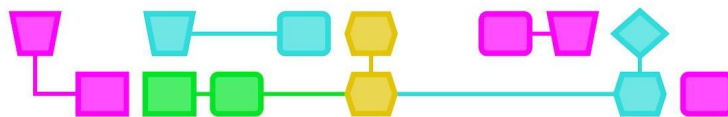
- Generelle færdigheder: kreativitet, samarbejde, logisk tænkning
- Grundlæggende computertænkning: Algoritme, mønstergenkendelse
- Computertænkningens begreber: variable, sekvenser, løkker og betingelser
-

Materialer:

- Præsentation: Dansecomputeren
- Danse piktogrammer
- Kuglepenne
- Papir

Forberedelse:

Læs lektionsplanen igennem og forbered præsentationen.



Lektionsbeskrivelse for Offline programmering

Introduktion – Komme fra A til B (10 min)

Forklar eleverne, at de skal lave deres egen algoritme. En algoritme er en proces eller et sæt regler, som man følger for at lave beregninger eller løse problemer – især for en computer.

Fortæl eleverne, at de i den første øvelse selv skal være computeren, og at de skal 'programmere' dig til at gå fra punkt A til punkt B i klasseværelset. Vælg et sted og mærk det som A, og vælg et slutpunkt, som du mærker som B. Sørg for, at der er nogle forhindringer på vejen, f.eks. et bord, som du enten skal kravle under eller klatre over.

Vælg 2-3 elever, som på skift skal prøve at styre dig fra A til B ved at give instruktioner som: *"Tag to skridt frem"* eller *"Drej til højre"*.

Det er vigtigt, at du følger deres instruktioner **helt bogstaveligt** – hvis de siger *"Gå ligeud"*, så bliver du bare ved med at gå, indtil du støder ind i noget (og fortsæt med at lade som om, du prøver at gå fremad). På den måde lærer de, at de også skal huske at sige *"Stop"*.

Når øvelsen er i gang, vil du opdage, at det ofte er svært for eleverne at forklare alt helt præcist. Fortæl dem bagefter, at det, de netop har lavet, er en algoritme: en proces eller et sæt regler, man følger trin-for-trin for at løse et problem — i dette tilfælde at komme fra punkt A til punkt B. De gav dig instruktioner trin-for-trin og forklarede hvert skridt meget præcist.

Koreografi – Algoritme (20 min)

Mål: Algoritme, tydelige instruktioner.

Forklar, at en algoritme kan bruges til mange forskellige ting: at binde snøreband, følge en opskrift eller lave en danse-koreografi.

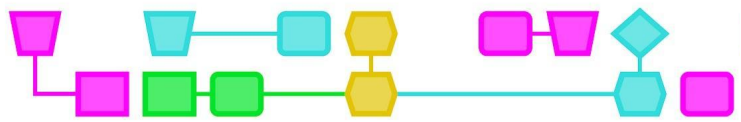
I denne opgave skal eleverne lave en algoritme for en dans, som en anden gruppe skal udføre. Del eleverne op i grupper af 3-4. Hver gruppe skal lave en dans (koreografi), der varer ca. 1-2 minutter.

Trin 1: Eleverne skal tænke over, hvilken dans eller hvilke bevægelser de vil bruge.

Det kan være en kendt dans, som fx *"Hoved, skulder, knæ og tå"*, eller en ny dans, de selv finder på.

Trin 2: Derefter skal eleverne skrive dansen meget præcist ned på papir, så en anden gruppe kan udføre deres dans ved blot at læse instruktionen.

Differentiering: Brug danse-piktogrammer til elever, der har svært ved at læse og skrive



Trin 3: Eleverne bytter deres koreografi med en anden gruppe.

Grupperne får fem minutter til at læse og øve den anden gruppes koreografi — og derefter skal de opføre dansen for hele klassen.

Vigtigt: Eleverne må kun udføre det, der står skrevet på papiret.

Trin 4: Gennemgå dansene og tal med eleverne om:

- Udførte grupperne dansen helt ens?
- Hvad fungerede godt, og hvad fungerede mindre godt?

Det er meget sandsynligt, at eleverne ikke udførte dansen helt ens.

Anerkend det — og forklar, at algoritmer er trin-for-trin-instruktioner, som man kan bruge til at løse problemer.

Man kan også bruge algoritmer til at få en computer til at udføre noget.

I denne øvelse var **algoritmen** beskrivelsen af dansen, og den gruppe, der udførte dansen, var **computeren**.

Når man beskriver trin, er det utrolig vigtigt at være præcis, fordi uklare instruktioner kan føre til fejl i programmet (= forskelle i den måde, dansen bliver udført på).

Præcise instruktioner er svære at skrive, fordi alt skal være meget tydeligt.

Forklar, at det sikkert er derfor, dansene ikke blev helt som koreograferet.

Tip: Sæt musik på, når koreografierne skal opføres.

Koreografi – Loops og Betingelser (20 min)

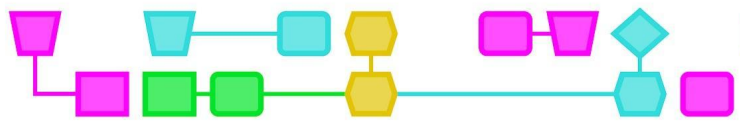
Formål: At lære, hvad loops (gentagelser) og betingelser (hvis-så regler) er.

Tal med eleverne om:

- Hvad ville I gøre anderledes nu, efter at I har set dansen blive opført?
- Hvordan kan vi sikre, at dansen bliver udført mere præcist?
 - Forenkle bevægelserne
 - Definere bevægelserne tydeligere
 - Aftale hvilke bevægelser danserne skal lave
 - Gentage bevægelserne (loops)

Start præsentationen “Dansecomputeren”:

- Vis **slide 1** og fortæl eleverne, at de skal lave opgaven igen — men denne gang ved at bruge piktogrammer.
Forklar, at hvert piktogram er en kode for en bevægelse, som computeren kan udføre.
- Vis **slide 2** og forklar piktogrammerne.



- Vis **slide 3** og lav dansen sammen med klassen.
Tal derefter om, hvordan man kunne få denne dans til at gentage sig **tre gange**.
- Vis **slide 4** — den viser, hvordan man kan gentage alle trinene, men at det hurtigt bliver uoverskueligt.
Tal med eleverne om, hvordan man kunne gøre dette kortere.
- Vis **slide 5** — dette kaldes en **loop (gentagelse)**.
Tal med eleverne om, om der kunne være andre steder, hvor man kunne bruge en loop (f.eks. ved at klappe i hænderne).
- Vis **slide 6** — dette kaldes en **betingelse (if-then-regel)**.
Forklar: Hvis noget sker, så sker noget andet.
I eksemplet: Hvis musikken spiller, gentages dansen.
Hvis musikken stopper, stopper programmet.

Fortæl eleverne:

Nu skal I arbejde i grupper og lave jeres egen koreografi ved at bruge piktogrammerne til at designe en dans.

Husk, at I skal bruge **mindst to loops (gentagelser)** og **én betingelse (if-then-regel)** i jeres dans.

Når I er færdige, taler vi sammen om, hvad I har lært i dag.

Afrunding med eleverne:

I dag har I lært at programmere en dansecomputer — med funktioner, loops og betingelser.

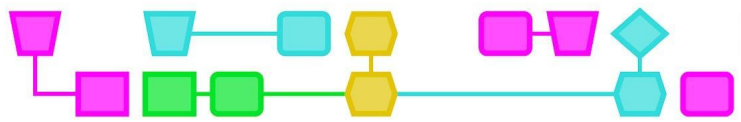
Selvom I brugte piktogrammer som “kodesprog” til at lave jeres dans, kan man på en computer bruge andre programmeringssprog, f.eks.:

- **Scratch**
- **MakeCode**
- **Kodetu**

I disse programmeringssprog kan man også arbejde med funktioner, loops og betingelser.

Når man arbejder med programmering, bruger man **algoritmer** til at designe programmer, som kan løse opgaver — præcis som I gjorde i dag med jeres dans!

Algoritmer	Et sæt logiske instruktioner til at udføre en opgave. I datalogi bruges algoritmer til at designe computerprogrammer.
Sekvens	En række instruktioner, der udføres én efter én i rækkefølge.
Funktion	Et blok af organiseret, genbrugelig kode, der udfører én enkelt, relateret handling.
Instruktioner	En specifik handling, som kan udføres af en person eller en computer

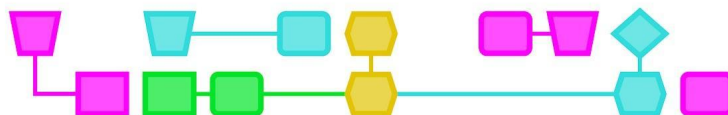


Loop	En gentagelse af én eller flere instruktioner
Betingelse (Condition)	En hvis-så-regel, der siger, at noget skal ske, før noget andet sker.

Tip: Vis hvordan en function, loop og betingelse ser ud i et programmerings program som Scratch.

Afrunding (5 min):

Fortæl eleverne, at de har lavet deres egen algoritme i denne lektion! Forklar, at der findes mange algoritmer i verden, for eksempel i trafikkontroller som trafiklys eller på sociale medier som TikTok. Faktisk bruger de også trin-for-trin instruktioner, når de børster tænder, dækker bordet eller laver matematik, som gør tingene nemmere. De ser også loops (gentagelser) og betingelser i hverdagen, for eksempel i en opskrift, hvor noget skal gentages, eller at hvis man har været på toilettet, skal man vaske hænder, eller at hvis man skal ud at rejse, skal man pakke en kuffert. Algoritmer, loops og betingelser findes overalt!



Kolonfon

© CTPrimED

Denne publikation er et produkt af CTPrimED (2021-1-NL01-KA210-SCH-000031319), finansieret med støtte fra Erasmus+-programmet i Den Europæiske Union. Denne publikation afspejler forfatterens synspunkter, og Kommissionen kan ikke holdes ansvarlig for nogen anvendelse af de oplysninger, der er indeholdt heri.

Project Coordinator

NEMO Science Museum, Holland

Partners

Universidad de la Iglesia de Deusto Entidad Religiosa, Spanien, Stichting Children's Science Museum Curacao, Curacao og Aarhus Bibliotekerne, Danmark.



Co-funded by the
Erasmus+ Programme
of the European Union



Universidad de Deusto
University of Deusto

Deusto

