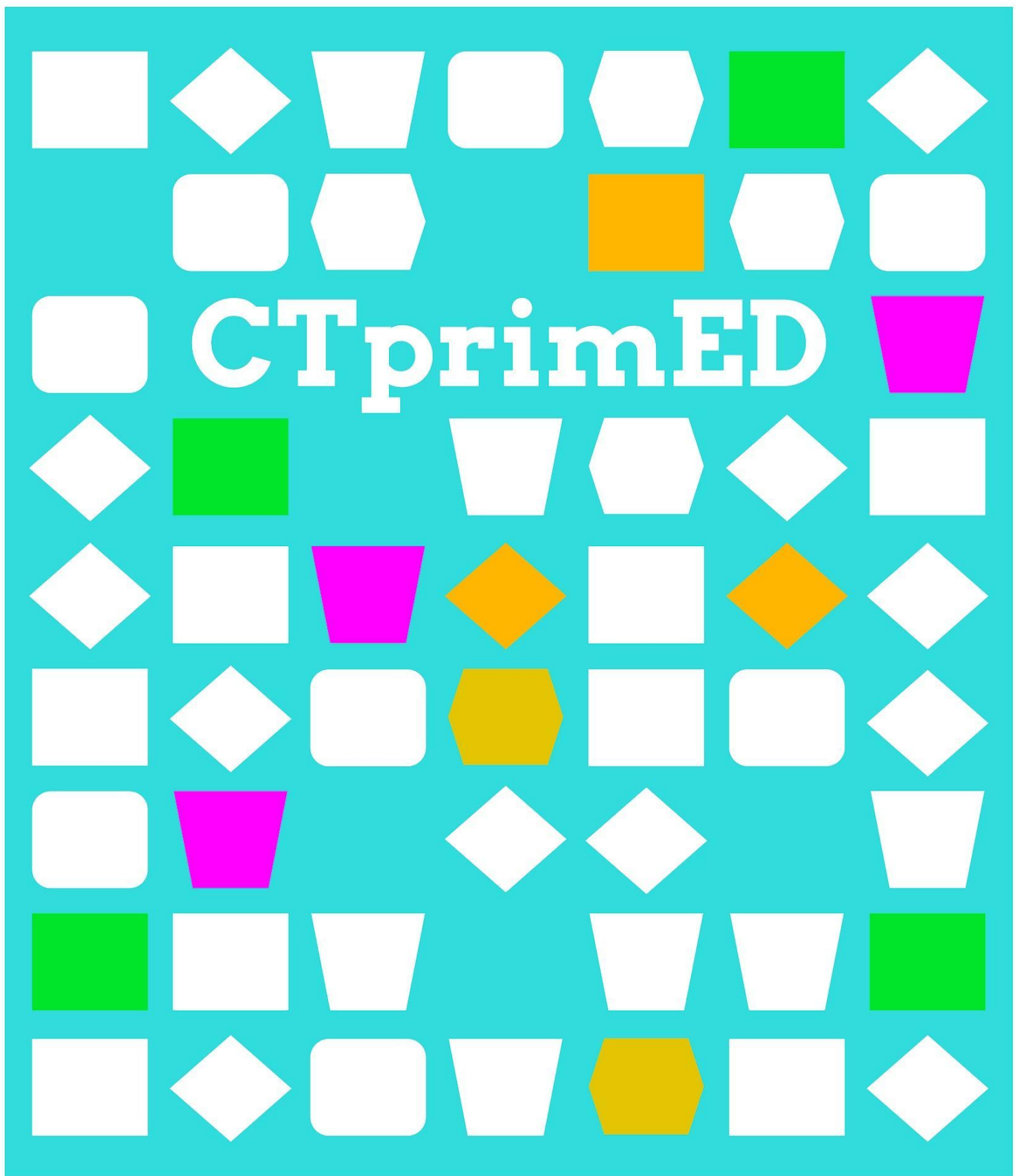
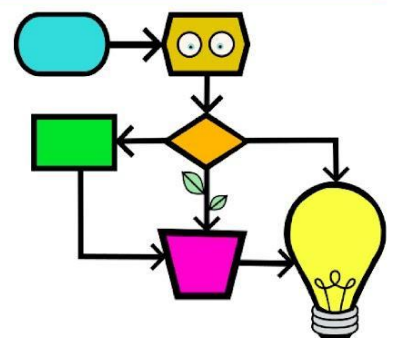
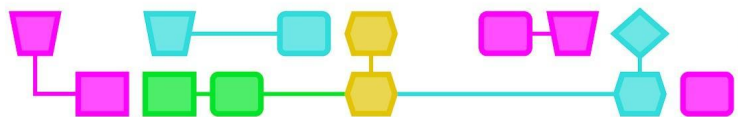




CT grundlag – Introduktion til mønster genkendelse





Sammendrag

Pattern recognition helps you to find similarities between problems and systems and apply previous solutions to solve new problems. In this lesson, students will use pattern recognition to divide cards into different categories. As an additional task, there is the possibility of writing an algorithm so a computer can also arrive at the same categories.

Målgruppe: 6-12 år

Varighed: 25 minutter – supplerende opgave - 25 min

Læringsmål: Efter undervisningen er eleverne i stand til at:

- Identificere mønstre i et sæt (eller datasæt)
- Forstå, hvordan mønstre er synlige i dagligdagen
- Bruge tidligere løsninger til at løse et nyt problem

Online/offline: offline.

Computational Thinking:

- Kreativitet, samarbejde og beslutningstagning
- Grundlæggende CT (Computational Thinking): mønstergenkendelse, abstraktion, algoritmer
- CT-begreber: sortering, klassificering

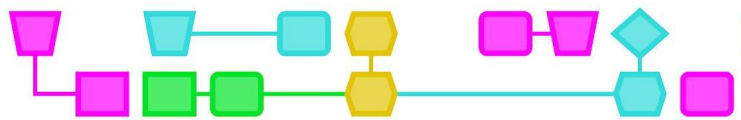
Materialer:

- Fotokortsæt (Bilag 2)
- Timer på et smartboard eller en telefon
- Små kasser – mindst fire til hver gruppe på 3-4 elever
- Arbejdsark til ekstraopgave

Tip: Laminer kortene så de er mere holdbare.

Forberedelse:

Kortene (Bilag 1) skal printes og klippes ud på forhånd. Del klassen op i grupper af 3-4 elever, og sørg for, at hver gruppe har sin egen arbejdsflade (bord eller lignende), hvor de kan lægge kortene. Ved lektionens start gives et sæt kort til hver gruppe



Generel introduktion til de fire lektioner om grundlaget for Computational Thinking

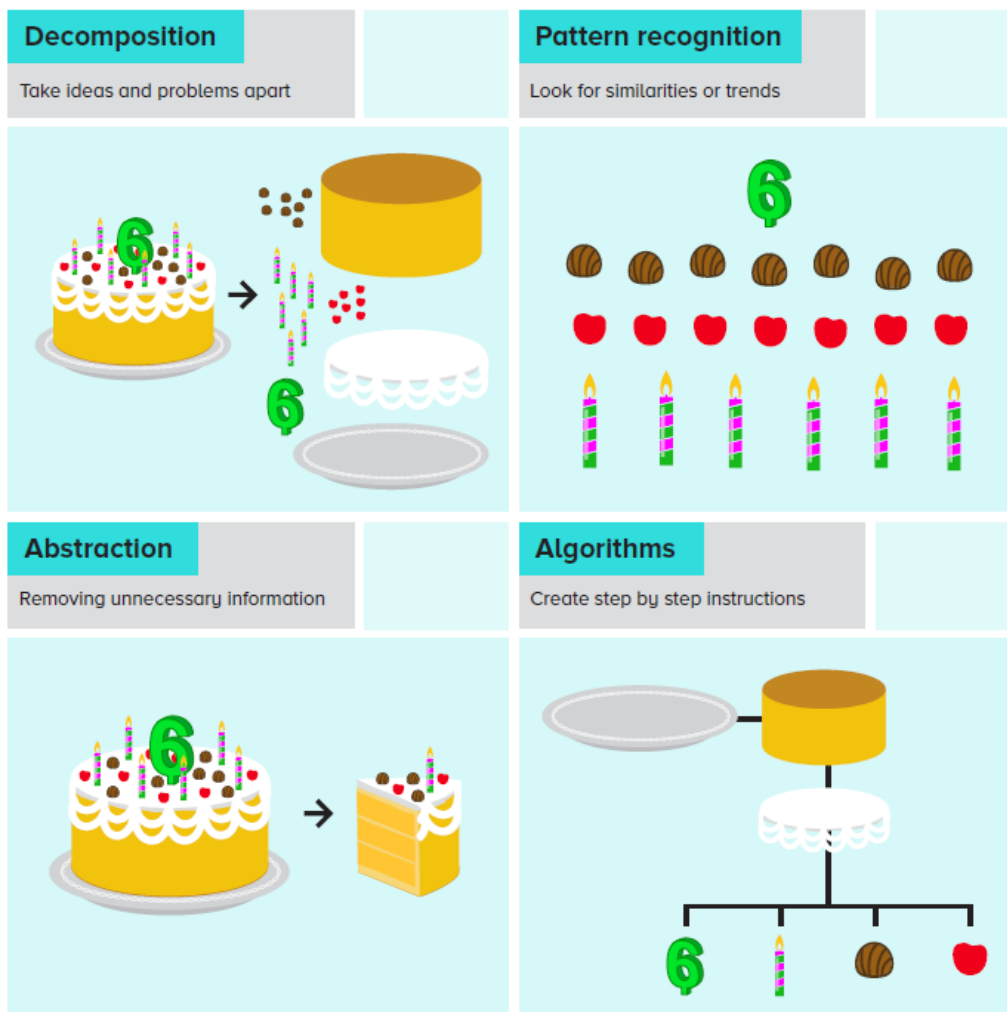
Spørg eleverne:

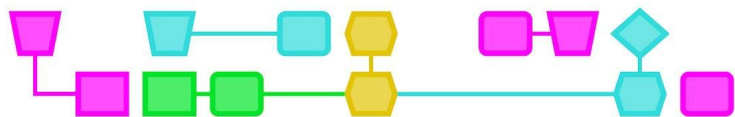
- Hvad ved I om, hvordan mobiltelefoner og computere virker?
- Kan de tænke selv? (Hvorfor, hvorfor ikke?)
- Hvem kontrollerer, hvad en computer gør?

Forklar for eleverne, at de skal arbejde med Computational Thinking. Simpelt sagt, drejer det sig om at lære, hvordan man får en computer til at løse et problem for dig. Det er ikke blot programmering, men også at lære, hvordan man bryder et problem ned i mindre dele eller genkender mønstre, så man bedre kan løse et problem. Der er fire hovedgrundlag for Computational Thinking.:

- Dekomposition → opdeling af et problem i mindre dele.
- Mønstergenkendelse → at lede efter ligheder eller mønstre i disse mindre dele, som kan hjælpe med at løse problemet.
- Abstraktion → at skelne mellem hoved- og sekundære aspekter. Hvad er virkelig vigtigt for at løse problemet?
- Algoritmer → at udarbejde trin-for-trin-instruktioner til at løse problemet.

I denne lektion bliver vi introduceret til dekomposition.





Lektionsbeskrivelse – Mønstergenkendelse

Opgave 1 – Sortering (10 min):

Del kortene ud, og instruer eleverne i at sortere kortene på tre minutter. Forklar om nødvendigt, hvad ordet 'sortere' betyder. Giv ingen yderligere forklaring, men lad eleverne selv bestemme, hvilke kategorier de vil sortere kortene i.

Sæt timeren til tre minutter. Når de tre minutter er gået, diskuteres følgende med eleverne:

- Hvilke kategorier valgte de? (Grupperne har sandsynligvis lavet forskellige kategorier.)
- Hvordan besluttede de sig for kategorierne?
- Hvad oplevede de som svært?

Merriam-Webster: sortering -
at placere noget på et bestemt
sted eller i en rangorden efter
slags, klasse eller natur.
Sig til, hvis du vil have den
omskrevet til mere enkelt sprog
til elever!

Forsæt med opgave 2.

Opgave 2 – Lav et antal kategorier (10 min):

Til denne opgave skal eleverne dele det samme sæt kort op i fire kategorier. Det er op til eleverne selv at vælge, hvilke fire kategorier de vil sortere kortene i. Normalt vil eleverne have svært ved at sortere kortene i fire kategorier, fordi de skal samle de kategorier, de brugte i den tidligere opgave. For eksempel, hvis de i første omgang havde kategorierne 'ting, dyr, mennesker, mad, former og farver,' skal de nu finde ud af, hvordan de kan slå disse kategorier sammen til fire.

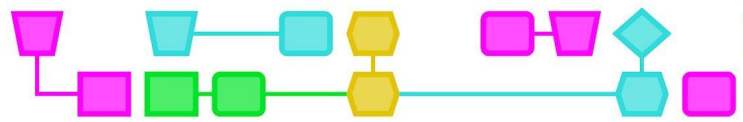
Sæt timeren til tre minutter. Når de tre minutter er gået, diskuteres følgende:

- Hvilke kategorier valgte de?
- Alle vil have forskellige kategorier, så der er ikke noget rigtigt eller forkert
- Hvordan besluttede de sig for kategorierne?
- Hvordan var denne aktivitet i forhold til den forrige?

Forklar, at mens de sorterede, ledte de efter mønstre: de så på, hvilke ting der var nogenlunde ens, og hvilke der ikke var. De skannede kortene med øjnene og opdagede for eksempel, at der var mad og dyr. Når de lavede kategorierne, forsøgte de altså at finde mønstre: hvilke ligheder og forskelle var der? Og ud fra det besluttede de hurtigt, hvad der skulle høre til i hvilken kategori.

Tip: Mange spil bruger mønstergenkendelse. Tænk på Minecraft, hvor du skal bruge bestemte "opskrifter" (mønstre) for at bygge værktøjer, eller i Roblox, hvor du lærer, hvordan forskellige spil fungerer, ved at genkende mønstrene.

Konklusion (5 min):

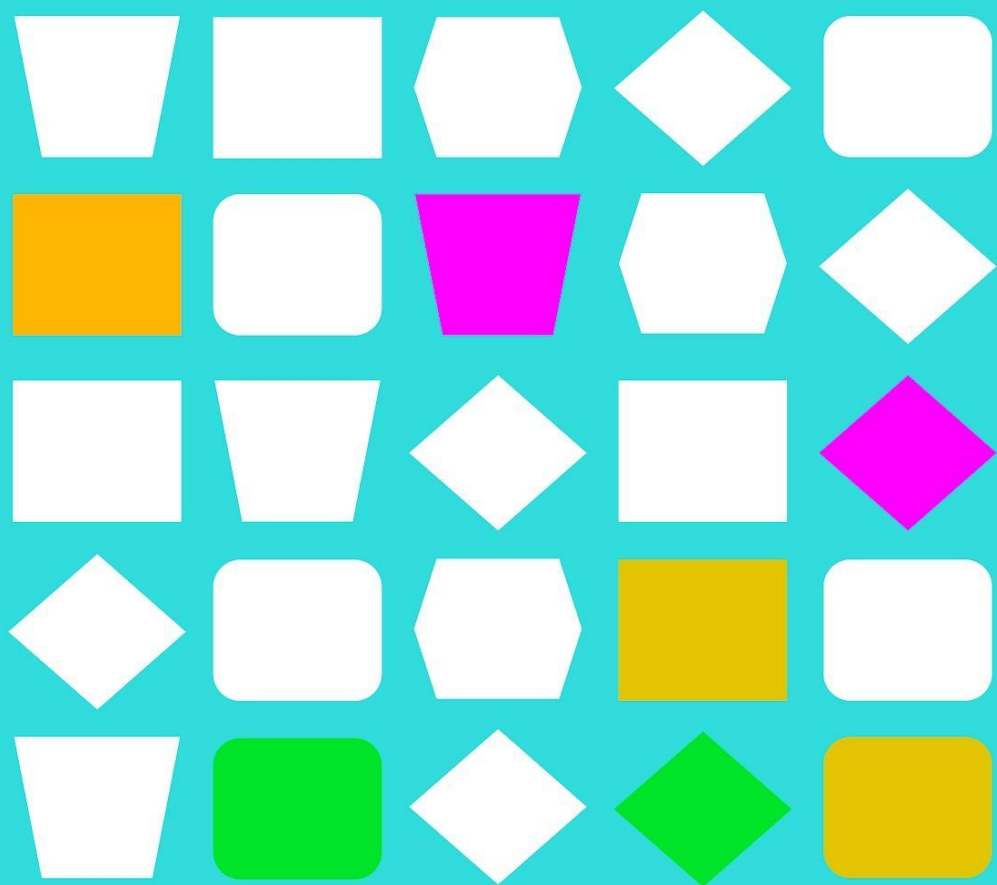


Alle disse kort samlet kaldes data. Computere arbejder ofte med store mængder data. For at fortælle computeren, hvordan den skal håndtere alle disse data, er det afgørende at kunne genkende mønstrene i dataene. Når man for eksempel designer en app om mad, behøver computeren ikke tænke på de enkelte ting, men derimod på de karakteristika, som alle tingene i kategorien har (f.eks. type, forarbejdning, vægt, farve, emballage og mere). På den måde modelleres datastrukturen, som derefter kan fyldes med konkrete data. Datastrukturmodellen er meget specifik, for når man fx designer en app til brug i et supermarked, er ovenstående punkter ikke interessante. Her er man mere interesseret i pris, rabatter, kvalitet osv. Mønstre er derfor meget specifikke i deres anvendelse, hvilket kan gøre det svært at afgøre, hvilket præcist mønster man leder efter i dataene.

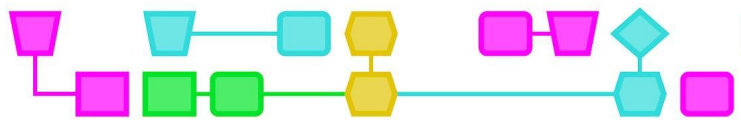
Mønstergenkendelse er også meget vigtigt i dagligdagen. For eksempel kan elever løse regnestykker hurtigere, hvis de kender mønstrene. Tænk på at løse 6 gange 5, hvis man allerede kender 5 gange 5. De kan også lettere genkende dyr, fordi de ved, hvilket mønster dyret hører til (har det ben, pels, knurhår osv.?). Mønstre gør altså livet en lille smule nemmere.

Tip: Du kan bruge elevernes svar til at forklare begrebet **abstraktion**.

Abstraktion betyder, at man kan se bort fra de små detaljer og i stedet få øje på det vigtige overblik. Det er netop det, man gør, når man "skanner" kortene og bestemmer kategorierne. Man kigger ikke på alle detaljer ved hver ting, men på hvad der nogenlunde passer sammen. Når man skal løse et problem, kan man også vælge at se på det store billede i stedet for på hver eneste detalje. Man vil ofte opdage, at man hurtigere kan finde en løsning på den måde.



Appendiks



Appendiks 1: Ekstra aktivitet – Sorteringsalgoritme for dyrekort (25 min)

Betingelser og regler (5 min):

Brug et sæt dyrekort og forklar eleverne, at de også kan bruge en computer til at sortere kortene (dataene) for dem. I denne opgave skal de lave en algoritme til en computer, som sorterer kortene. For at gøre instruktionerne så klare som muligt for computeren, kan man bruge betingelser. Det er en hvis-så-regel, der siger, at noget skal ske, før noget andet sker. Et eksempel på en betingelse til sortering kunne være:

- "Hvis den har ben, er det et dyr, og det skal i den første kasse"
- "Hvis den har fjer, skal den i den anden kasse."

Algoritmer (25 min):

Eleverne bruger arbejdsarket, sorteringskortene (begge i appendiks) og fire kasser til at skrive en algoritme, som en computer kan bruge til at sortere kortene i kategorier. De laver fire (eller flere) kategorier og prøver at finde betingelser for at skrive et program, der sorterer kortene i disse kategorier.

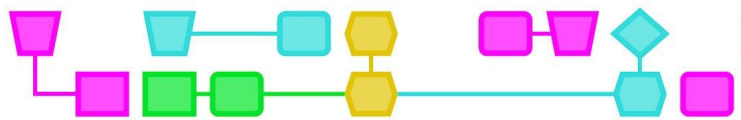
Når eleverne har skrevet sorteringsprogrammet, skal de bytte programmer og prøve at afprøve en anden gruppes algoritme. Diskuter, hvordan algoritmen fungerede, og om instruktionerne var klare nok.

Eleverne skriver sandsynligvis ikke en perfekt sorteringsalgoritme, da det er meget svært at forklare betingelserne helt præcist. For eksempel kan eleverne have betingelsen: "Hvis det har fjer, er det en fugl, og det skal i den første kasse mærket 'fugle'." Men hvis de også har en kasse mærket 'husdyr', bliver det uklart, hvor en høne egentlig skal placeres. En computer kan slet ikke tænke selv, og derfor er det meget vanskeligt at skrive sorteringsalgoritmer, selv når man kun sorterer dyrekort.

Afslutning (5 min):

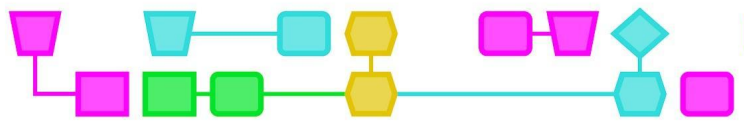
Forklar eleverne, at de har lavet hvis-så-regler (= betingelser) for at skrive en algoritme, som en computer kan bruge til at sortere dyrekortene. Ved at lave betingelser kan computeren afgøre, hvilket element der hører til i hvilken kasse. Da en computer slet ikke kan tænke selv, kræver sorteringsalgoritmer præcision og meget forklaring, hvilket er grunden til, at de som regel er svære at skrive.

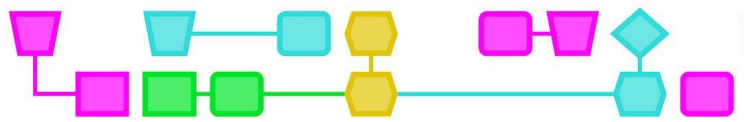
Forklar, at de selv også udfører algoritmer i deres hverdag! Tænk på at børste tænder eller lave en dans. Det er alle trin-for-trin-instruktioner, som de følger for at gøre en bestemt opgave nemmere eller mere glidende.

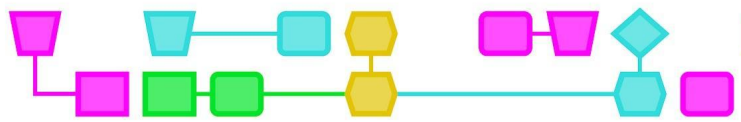


Appendiks 2: Kort til udprintning









Kolofon

© CTPrimED

Denne publikation er et produkt af CTPrimED (2021-1-NL01-KA210-SCH-000031319), finansieret med støtte fra Erasmus+-programmet under Den Europæiske Union.

Denne publikation afspejler forfatternes synspunkter, og Kommissionen kan ikke holdes ansvarlig for nogen brug, der måtte blive gjort af de indeholdte oplysninger.

Project Koordinator

NEMO Science Museum, Holland

Partnere

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



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



Universidad de Deusto
University of Deusto

Deusto

