# Dette bilag er en del af inspirationsmaterialet til faget [Erhvervsinformatik](https://emu.dk/eud/erhvervsinformatik/digital-myndiggorelse) på emu.dk. Programmering indgår i kompetenceområdet ”Teknologisk handleevne og computational tankegang”. Nedenfor ser du en kort beskrivelse af forskellige sproglige udtryk for programmeringselementer, som vil være relevant at inddrage i undervisningen.

Om programmering

### Algoritmer, sekvenser og sætninger

En algoritme er det, programmet gør trin for trin, altså en slags opskrift for at programmet kan løse en given opgave. En algoritme består som regel af forskellige sekvenser, som er en serie af instruktioner eller instruktionsblokke, som skal udføres efter hinanden.

### Kontrolstrukturer

De enkelte instruktioner i en algoritme kaldes sætninger eller kommandoer. Algoritmer benytter sig af forskellige kontrolstrukturer. De vigtigste er:

|  |  |  |  |
| --- | --- | --- | --- |
| SEKVENTIEL KONTROL | SAMMENSÆTNING |    UDVÆLGELSE |  GENTAGELSE |
| https://lh5.googleusercontent.com/vK61lgLfvj1y_1njbZfX66dens5qZ3hEvlPLE2Lp0-aBZEf1oXkV79pPyzCKhxW82W_i_4s258aU2KgexXc_4hlO9VacBGztDqNSKXlkNXCaiDiOBvCWHGPybS6z5DBv7TJFxSzI | https://lh3.googleusercontent.com/msl6K9k7ffVaRtDbILuKBxFh-Tzg0HKmO-JlSsY10EEr1OJhBcApUrczCzQTEdbKd9Fmz0LNkRSNGgWv3vxIjVh7UYRds-UuoY7TWwAq0KrzaXWCKA1n6JQxpxSL49_XfRBJn_UX | https://lh6.googleusercontent.com/BYqCaRWp_J0givatFmHHE-M9GFs4t3faSecqvvWCI9rC-9uO1HN9P1SLq59ztKP1lyjLZOHWxxuLkEvayYrqfwtP8tyJzAgDYwzmc03cwkxXpiIj_3U_8jFK6bu02hxCE2RXSsJm | https://lh4.googleusercontent.com/b4qVOG9l9MK_CVLeuM52XxY0ly2LiYg7JjJCfu4Sa9ugvvP05r2p-9Tn58LiI05w8l18TnFx-Nh7a6zmYUzLPSLqtlZsor7pqOxUTnJWEYUm1DsDSo28JyTL22QtZRjyzC9VphTI |
| Kommandoer følges i den angivne rækkeføl-ge. Første kommando skal udføres, før programmet går videre til den næste. | Kommandoer sammensættes til en enkelt blok, fx når **variable** defineres ved start af programmet (sæt point til, sæt liv til, osv. eller sæt liv til x, hvis y rører x, sæt x = x-1 | En kommando udvælges blandt flere muligheder: **if** (logisk/ boolsk udtryk); noget er enten **sandt** eller **falskt)...else.** | Fx **løkker.** En kommando gentages et antal gange. Antallet af gentagelser kan styres af et angivet fast tal, men styres også ofte af et logisk udtryk. |

### Mere om variable

**Variabelerklæringer**, hvor man opretter en variabel og angiver dens *datatype*. En variabel kan ændre sig og kaldes også nogle gange for et parameter. En datatype kan fx være Numbers - eks. heltal (integers), decimaltal (floats) - eller tekst (string).

**Tildelinger,** (eng.assignment), hvor man giver en variabel en bestemt *dataværdi*. Når en variabel får tildelt en bestemt værdi, kaldes den enten et “aktuelt parameter” eller et ”argument”.

I forbindelse med data taler man også om lister og arrays. Begge dele er måder at organisere flere dataforekomster på. Læs mere herom på undersiden om data og datatyper.

### Mere om udvælgelse

**Udvælgelse** kaldes også **forgreninger.** Her afgør et sandhedsudtryk, om programmet skal udføre den ene eller anden gruppe sætninger. Forgreninger indledes med en betingelse (if/if-else-statements) og fortsætter ofte med then/else-statements. I et sandhedsudtryk er der kun 2 værdier (sandt/falsk), fx ”if touches colour red → turn right, else→ turn left”. Her røres der enten vil farven rød eller ikke. Den type udtryk kalder man også for boolske udtryk (eng. Booleans), og de regnes også for en datatype.

### Mere om gentagelse

**Løkker:** Hvis en gruppe sætninger skal udføres flere gange, indfører man en løkke i programmet.(eng. Loops). En løkke kan angives med forskellige udsagn, fx:

* while x do x (gentager sålænge noget andet sker)
* repeat x gange (gentager det antal gange, der er angivet)
* repeat forever (gentager for evigt)
* repeat until x (gentager indstil noget andet sker)

**Funktioner-/procedurer og funktions-/procedurekald.** Her er en gruppe sammenhørende sætninger/en sekvens er blevet samlet i en selvstændig programdel, som så *kaldes,* dvs. aktiveres inde i programmet. Det bruges fx, når man har brug for at udføre den samme sekvens flere steder i et program eller når brugeren beder om det.

Forskellen på funktioner og procedurer er, at en funktion returnerer et tal, når den modtager et input, mens en procedure returnerer en handling. I begynderprogrammering er forskellen ikke så vigtig - især ikke, fordi nogle programmeringssprog bruger begrebet "funktioner" om det hele. Der er også sprog som bruger helt andre ord, fx (sub)rutiner eller metoder.

### Operatorer

Programmeringssprog benytter en række operatorer for at kunne udtrykke logik og relationer mellem de enkelte elementer. Disse kan opdeles i matematiske og logiske operatorer.

Matematiske operatorer er fx lig med (=), ikke lig med (!=), større end (>), mindre end (<), lige med eller større end (=>), lig med eller mindre end (=<). Disse bruges fx i et spil, hvor næste level skal starte, når point = 100 eller slutte, når liv = 0.

Logiske operatorer kan sammenkoble flere betingelser, fx bruges "AND" (&&), hvis to betingelser skal være til stede, før noget sker, eller "OR" (II), hvis blot en af betingelserne skal opfyldes, inden noget sker. Logiske operatorer kan også bestemme, at noget sker, hvis en betingelse ikke er opfyldt. Det gøres med "NOT" (!)

### Syntaks

Tekstprogrammeringssprog har oftest en fast definition for, hvad der er muligt at udtrykke i sproget, og hvordan det gøres. Syntaks er de regler for form/grammatik, som sproget har, fx en parantetisk BEGIN…END struktur, brug af citationstegn, kolon, semikolon, store/små bogstaver, osv.

### Fejlsøgning (aflusning/debugging):

Proces, hvor man søger efter og retter evt. fejl

###