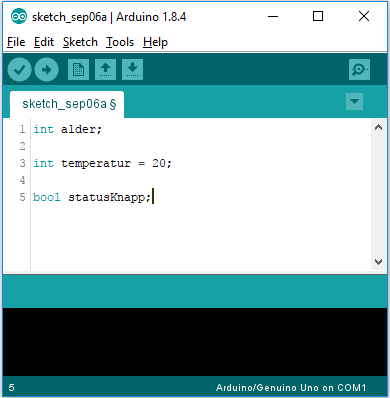
## Variabler

Variabler er plassholdere i datamaskinen for informasjon vi ønsker å ta vare på og endre i løpet av programkjøringen. I vinduet til høyre ser du eksempel på tre variabler. Der opprettes (deklareres) de ved at vi både gir dem navn og knytter til dem en datatype.

Først har vi variabelen **alder**. Den har ingen verdi ennå, men vi forteller mikrokontrolleren at det må sette av plass til den. Den skal være av datatypen Int, men mer om akkurat det i neste kapittel.

**temperatur** gir vi verdien 20. Denne verdien er nå lagret og vi kan hente den ut ved å referere til variabelnavnet temperatur. Datatypen er Int.

**statusKnapp** er også en variabel uten verdi, men av typen bool i stedet for int. Vi merker oss bare det nå, og får forklaringen lenger ned i dette dokumentet.

**Plassholdere**: Nedenfor ser vi illustrert de tre deklarerte variablene med sine plassholdere i datamaskinen. Du kan også tenke på plassholdere som bokser eller skuffer som datamaskinen lagrer informasjon i. Merk at det var bare *temperatur* vi ga en verdi fra starten av, slik at verdien for *alder* og *statusKnapp* inntil videre er ukjent. De vil få verdier senere i programkjøringen:

int alder;

int temperatur = 20;

bool statusknapp;

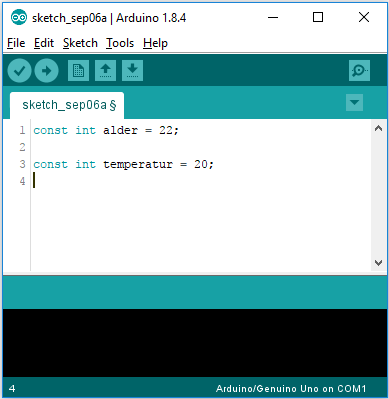
?

?

20

Hva benyttes så variabler til? Jo, egentlig bare to ting: vi gir den en verdi (kalles tilordning), eller vi leser verdien. Når vi sier at vi gir den en verdi, så vil det gjerne skje flere ganger – altså den kan få verdien endret flere ganger i løpet av programkjøringen. Etter hvert som du får prøvd å programmere vil bruken av variabler bli åpenbar.

## Konstanter

Konstanter er nært beslektet med variabler, så ofte snakker vi om «konstanter og variabler» i samme åndedrag. De hører sammen.

Vi velger å bruke en ***variabel*** dersom vi ønsker en plassholder for verdier som skal kunne variere (endres) i løpet av program-kjøringen.

Dersom vi ønsker å holde en verdi konstant (uendret) gjennom hele programkjøringen, velger vi logisk nok å bruke en ***konstant*** i stedet.

Variabler og konstanter deklareres (opprettes) nesten helt likt – forskjellen er at vi setter **const** foran konstantene. Og så må konstanter alltid gis en verdi når vi deklarerer dem.

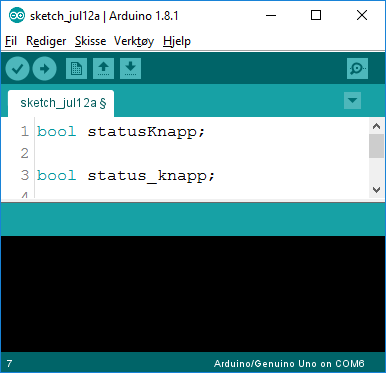
I vinduet over ser du at **alder** og **temperatur** har blitt deklarert som konstanter, og gitt hver sin verdi som kommer til å ligge fast igjennom hele programkjøringen. Faktisk vil du ikke få til å endre deres verdier om du prøver. Datamaskinen låser verdien til konstanter.

**Nytten av konstanter**: De lar oss gi konkrete verdier et symbolsk navn (som **alder** i stedet for 22 i eksempelet over), og det øker lesbarheten av programmet din. Altså hvor lett det er å forstå hva koden gjør når du leser programmet.

I tillegg vil de forenkle vedlikeholdet av programmet. Dersom du eksempelvis har brukt konstanten **alder** i stedet for 22 mange steder i programmet ditt, og så senere bestemmer deg for at verdien på **alder** heller skal være 25, da trenger du bare å endre startverdien i deklareringen av konstanten. Og med det har 22 enkelt blitt erstattet av 25 alle steder i programmet ditt hvor konstanten **alder** benyttes.

**Andre ting å huske med variabler og konstanter** er beskrivende navn og case sensitivitet:

Beskrivende navn! Det er både lurt, og det vi kaller god programmeringsskikk, å gi variabler beskrivende navn, slik at de som leser programmet enkelt kan forstå hvilken informasjon de representerer. Det øker lesbarheten.

I vinduet til høyre ser du to forskjellige måter å skrive variabler på. Den første metoden bruker det vi kaller camelback (Camel Case). Camelbackmetoden går ut på at det første ordet har liten bokstav, påfølgende ord har stor forbokstav, f.eks. statusKnappAlarm.

Du kan også bruke \_ (underline) til å dele opp ord hvis du vil, men det er uansett viktig å være konsekvent og holde seg til en måte å skrive på.

Vi liker camelback-metoden best, men du bestemmer selv.

Case sensistivitet: Det betyr at datamaskinen skiller strengt mellom små og store bokstaver.

A er ikke det samme som a. Og variabelen statusKnapp er ikke den samme som Statusknapp eller statusknapp, slik at du gjennom hele programmet ditt må være konsekvent på bruk av store og små bokstaver i et variabelnavn. Det samme gjelder naturlig nok også konstanter.

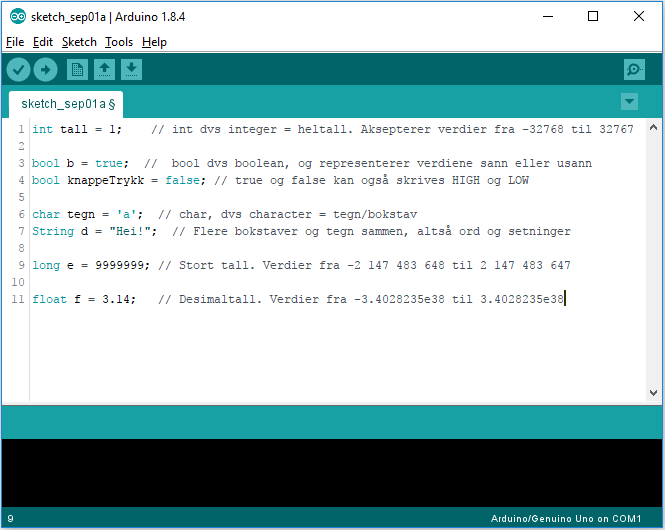
## Datatyper - Forklaring

Data (informasjon) generelt kommer i flere former: tekst, heltall, desimaltall, etc.

I programmeringsverden klassifiserer vi data som skal holdes i variabler og konstanter i det vi kaller datatyper.

Årsaken til det er at en mikrokontroller ikke har all verdens av plass til lagring av data, så for at den skal vite hvor mye plass den må sette av for en variabel/konstant, forteller vi den også hvilken type data som kommer til å bli lagret i variabelen/konstanten.

Nedenfor kan du se noe grunnleggende datatyper som er viktige å vite om for gjennomføring av laboratorieøvingene og robotbilprosjektet.



Mer om variabler kan leses på: <https://www.arduino.cc/en/Reference/HomePage>

Boolean: <https://www.arduino.cc/en/Reference/BooleanVariables>

Int: <https://www.arduino.cc/en/Reference/Int>

String: <https://www.arduino.cc/en/Reference/StringObject>