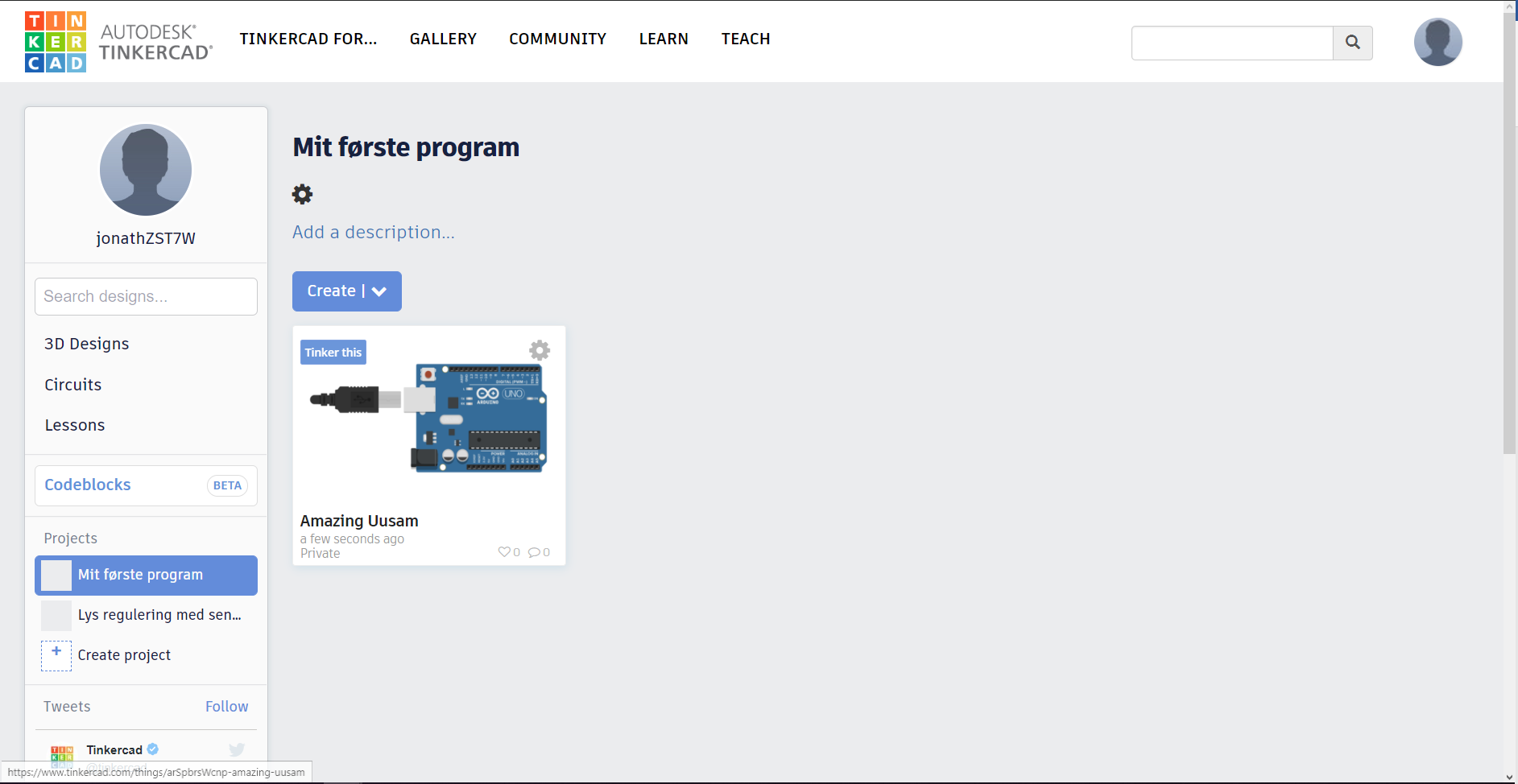
## TinkerCad EP 2 Hvordan Bygge krets i nettleser + gjennomgang av Blinky og seriel

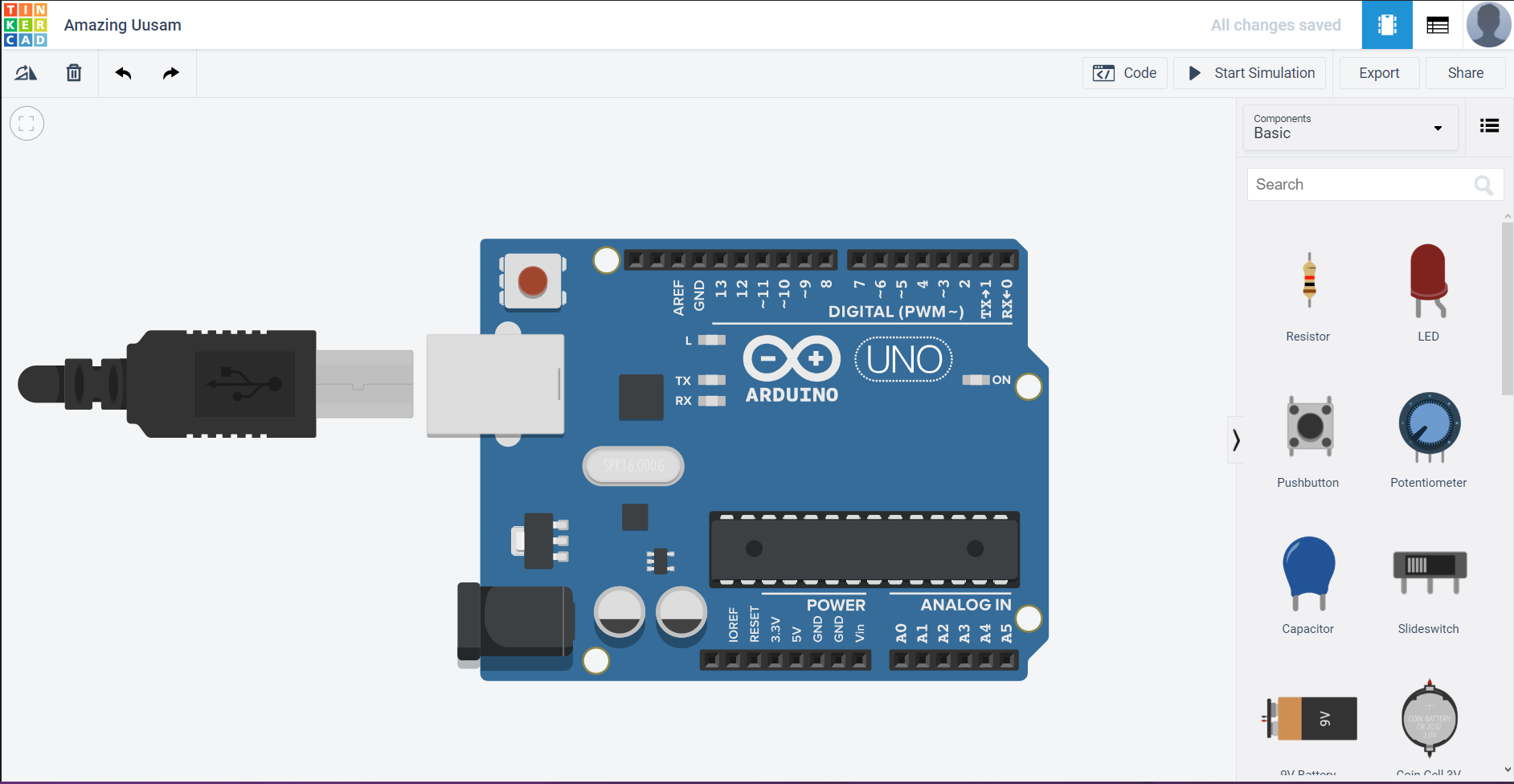
Hvordan bruke TinkerCad til å bygge kretser og vi kommer til og gå lit dypere igjennom Blinky og Hello World som vi benyttet os av forrige modul(TinkerCad intro ep 1).

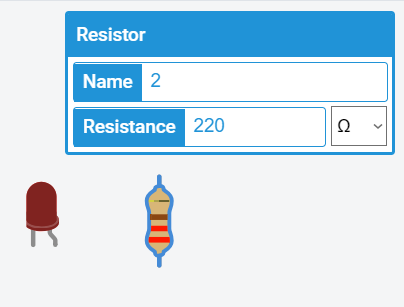
## Steg 1 - bygge blinky krets

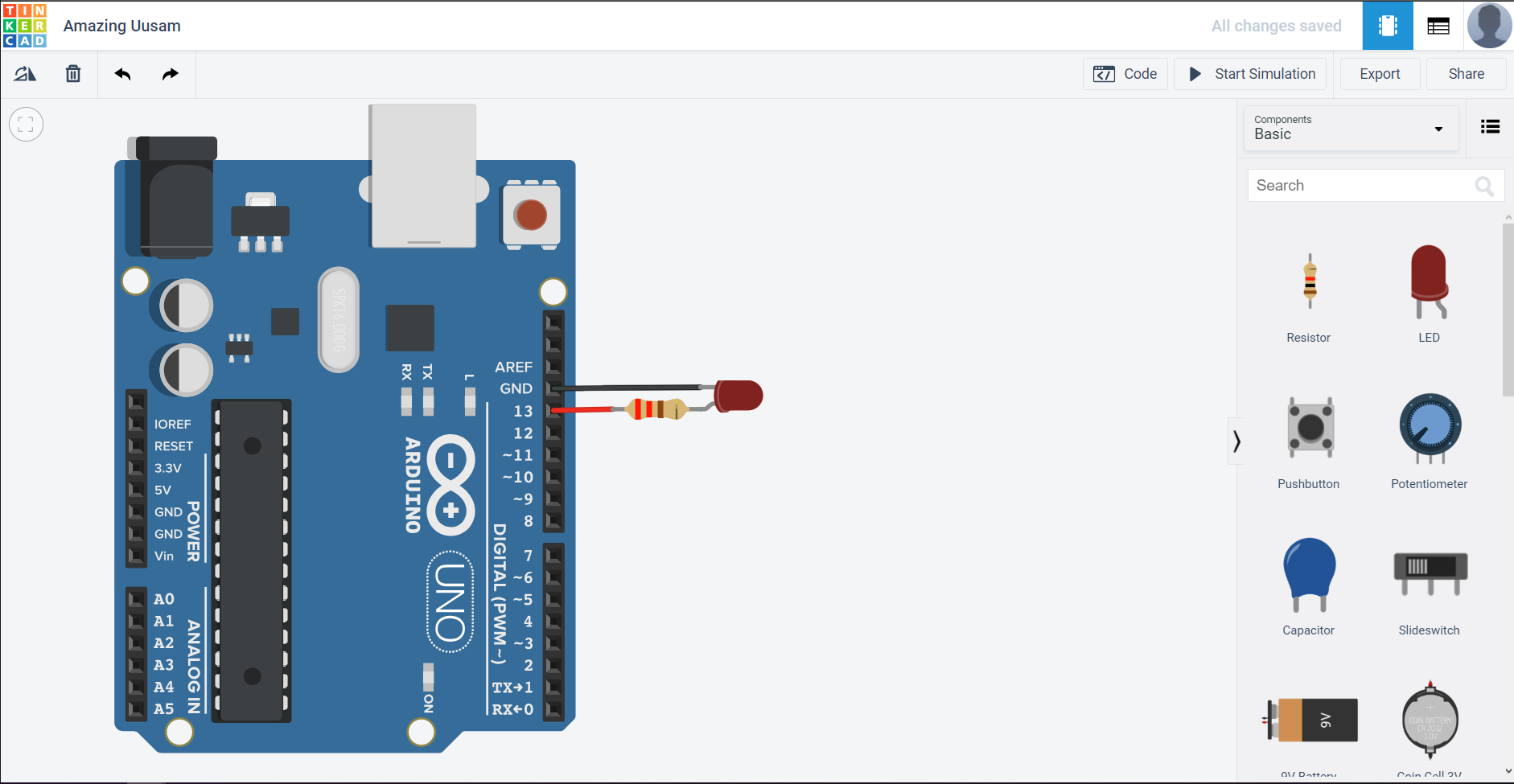
La så gå til tinkercad igjen og logge inn, velger prosjektet vi startet med sist gang (Mitt første program) og jobber videre fra den. Går også fint an å lage ny cuircuit om man vil det.



Da starter vi med bare Arduino’en som den var sist men nå må vi legge til flere komponenter fordi vi ønsker å bygge en enkel krets. Vi må ha en motstand og en LED og det finner vi i komponent biblioteket på høye side (markert i blått).



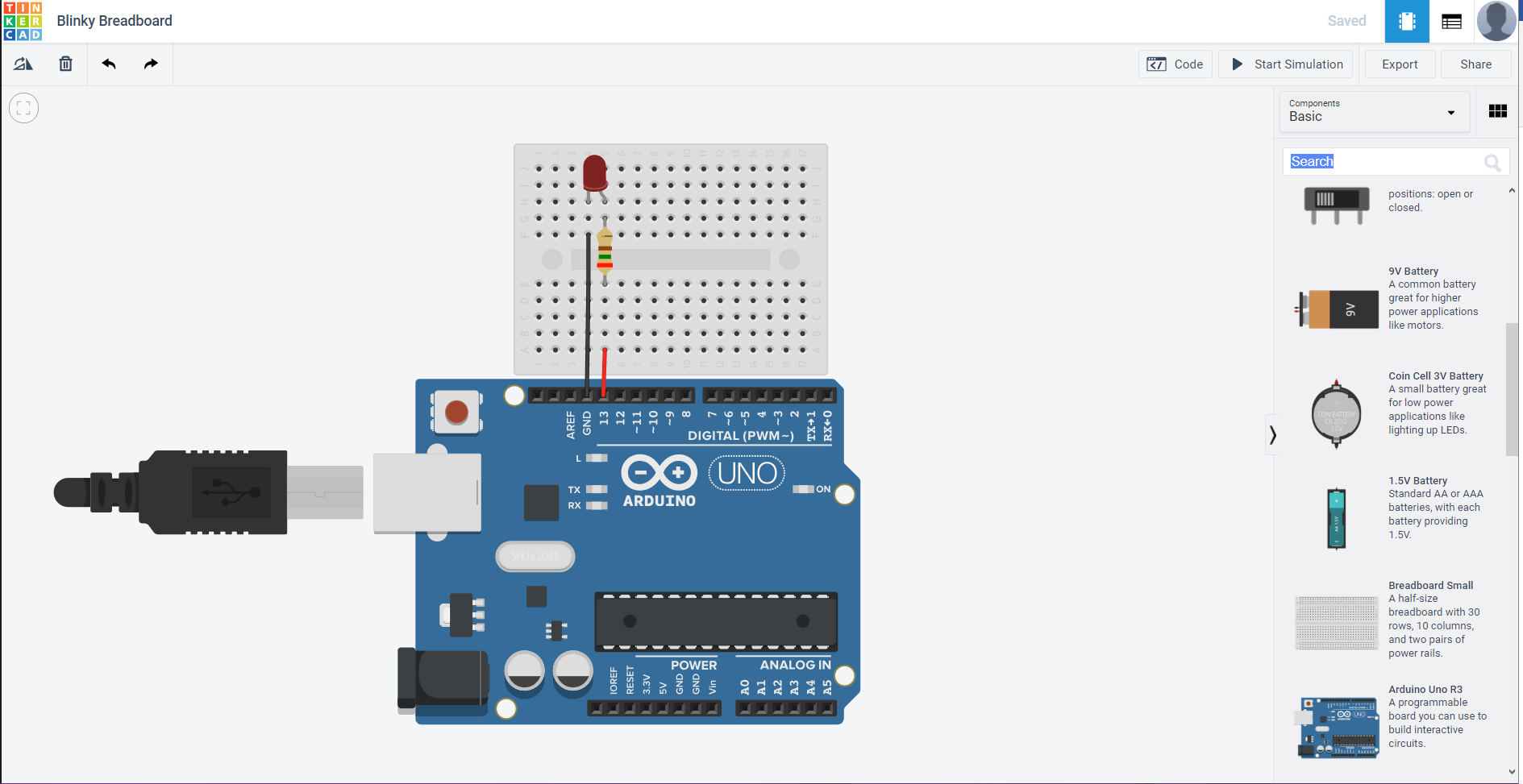
Etter vi har dratt en motstand og en LED fra komponent biblioteket og over til arbeidsområdet (markert i rødt). Etter du har dratt komponenten ut så kan man klikke på den og få opp flere alternativer, for resistoren kan vi velge hvilken verdi den skal ha og vi vil gjerne endre den til 220 Ohm, da dens funksjon er og avlaste LED’en, med en for høy motstand vil vi nesten ikke se at det lyser og en for lav motstand vil det fremdeles være en risiko for å skade LED’en. Alternativene for LED’en er farge så her kan man velge hva man vil om han har noen favoritt farger men vi gjør det enkelt og bruker rød her.

Oppkobling av kretsen vil se ut som på bildet under, og her ser vi at det er en motstand i serie med en LED’en og at den går fra GND (som er jord eller minus) og pinne13 (som er utgang /+).  
  
LED’en har to bein og hvert bein har sin funksjon, så her gjelder det og koble riktig. Et bein er katoden(cathode) og den skal til minus (jord). Og da må det andre beinet altså anoden til pluss.

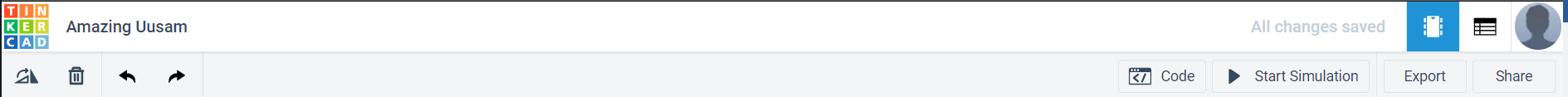
Om det er noe vi skall lage kretsen i virkeligheten så kan det være greit å ta med et breadboard (protobrett) da dette vil gi et innblikk i hvordan man setter opp ting i virkeligheten. Så da trenger vi følgende:

* Breadboard(protobrett)
* LED
* Resistor (motstand)
* Ledninger

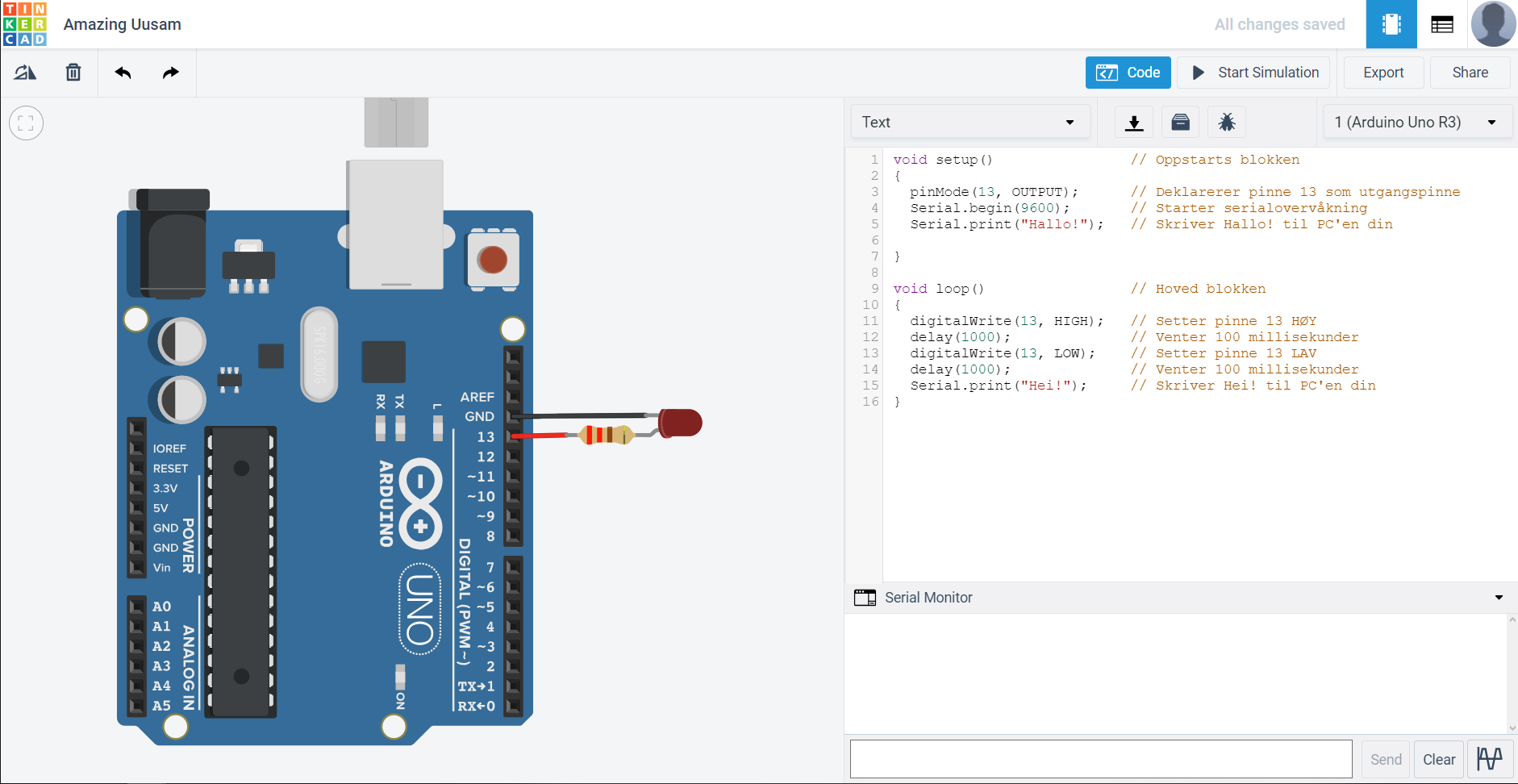
Alle komponenten finner vi i komponentlisa på høyre side. (det er mulig og endre visningen av komponentlisa ved å klikke på ene boksen over søkefeltet, mer informasjon om hver komponent vises). Under ser vi både med og uten protobrett. Leddingen kommer da man klikker på et punkt og drar musa over til der man vil feste den. Samme oppsett som i stad bare nå med protobrett.



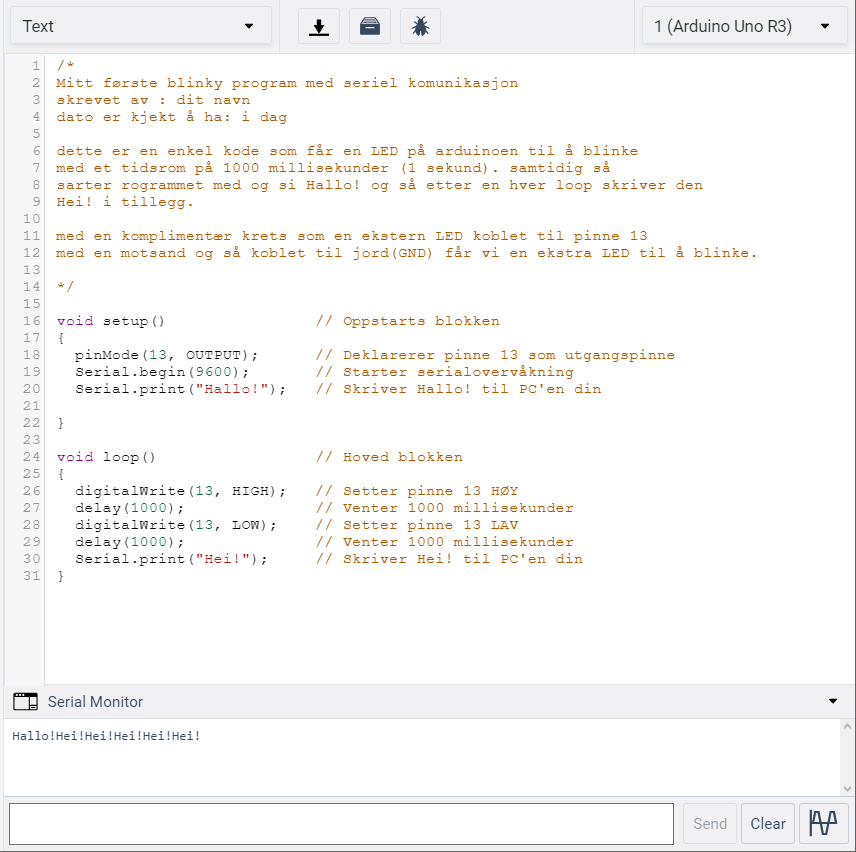
## Steg 2 - Tilbake til koden

La oss gå lit dypere inn i koden og se på hvordan den fungerer. La os starte med Arduino Sketchen som er koden som vi skriver. Klikk på Code knapen for å se koden din igjen.  
  


La os starte med å sjekke at alt fremdeles fungerer så vi kan klikker Serial Monitor barn så den er oppe og vi kan se om ting blir skrevet og da er det bare å trykke på Start Simulation å se at Sketch’en vår funker.



La os starte med og legge til en beskrivende tittel til sketch’en vår, dette kan vi gjøre ved å kommentere slik vi alle rede har gjort med to skråstreker før hver linje eller vi kan skrive /\* og så avslutte med \*/ alt i mellom vil bli kommentert ut og det kan strekke seg over flere linjer.

Her er det god skikk å ha med navn dato og lit om hvordan den funker og hva hensikten med koden er.

Nå har vi en god mal på hvordan en sketch skall se ut (se bilde til høyre) og lit om hvordan lage den men selv om alt ser fint ut nå så er det kanskje ikke alt som er like selvforklarende så la oss gå lit mer i dybden.

Vi kan starte med en funksjon, hva er det og hvordan funker det?   
En funksjon er min rutine eller en planlagt gjøremåte som er definert.   
  
En funksjon kan se ut som dette:  
voide setup() {} her er navnet ‘’setup’’ og informasjonen før definerer retur typen(data) og etter er parametere som retur typen bruker. Det skrives i parenteser. Og til slutt i krøllparentesene er kroppen til funksjonen og det er det er der det funksjonen gjør er. Vi vil se lit mer på funksjoner ved en senere anledning men det viktigste er at man må ha med setup og loop funksjonene i en arduino sketch. Og en funksjon kan ha flere funksjoner inni seg funksjonseption

La oss gå dypere inn i pinMode, digitalWite, Serialbegin og Serial.print. Som er selve maten i koden vår.

pinMode:   
Syntax = pinMode(pin, mode)

* Er en Digital I/O (Input/Output) funksjon som spesifiserer (bestemmer) oppførslene til en digital pinne.
* Har to parametere, hvilken pinne og hvilken modus. det som står inni parentesene under syntax
  + pin : nummeret på pinnen du vil bruke
  + mode: INPUT,OUTPUT, INPUT\_PULLUP som er valgene til hvordan pinnen skall oppføre seg.
* Returnerer ingen verdier.

digitalWrite:  
Syntax = digitalWrite(pin, value)

* Er en Digital I/O (Input/Output) funksjon som spesifiserer (bestemmer) oppførslene til en digital pinne.
  + Man kan velge mellom å sette verdien på pinnen høy(HIGE) eller lav(LOW).
* Har to parametere, hvilken pinne og verdien den skal ha (høy/lav, på/av).
  + pin: nummeret på pinnen du vil bruke
  + value: HIGE, LOW
* Returnerer ingen verdi.

Serial.begin:  
Syntax = Serial.begin(speed)

* Er en sub-funksjon som ligger under serial funksjonen, den er en del av communication funksjonene som er kommunikasjon funksjonene for ardunio. begin funksjonen brukes for å starte kommunikasjon og definer hvilken hastighet det skal kommunisere på.
  + den vanligste kommunikasjons hastigheten når man kommuniserer med en datamaskin er 9600, men det finnes mange andre hastigheter også, 2400,4800,14400 for å nevne noen.
* Har en parameter, baud-rate(kommunikasjonhastighet)
  + speed: antall bits per sekund (baudrate)
* Returnerer ingen verdi.

Serial.println:  
Syntax = Serial.println(val,format) eller  Serial.println(val)

* Er en sub-funksjon som ligger under serial funksjonen, den er en del av communication funksjonene som er kommunikasjon funksjonene for ardunio.pintln funksjonen brukes for å printe verdier serielt. den fungerer på samme måte som .print funksjon men den legger til et linjeskift (new line)  på slutten.
* Har to parametere, val (hva som skal printes) og formatet.
  + val: er verdien som skal printes det kan være tall eller tekst eller en variabel.
  + format: DEC, HEX, OCT, BIN er alternativne, om man ikke spesifiserer så er det decimal som er standar.
* Returnerer antall bits som er skrevet. size\_t:println()

Mer informasjon om disse + andre funksjoner finner du her: <https://www.arduino.cc/reference/en/>