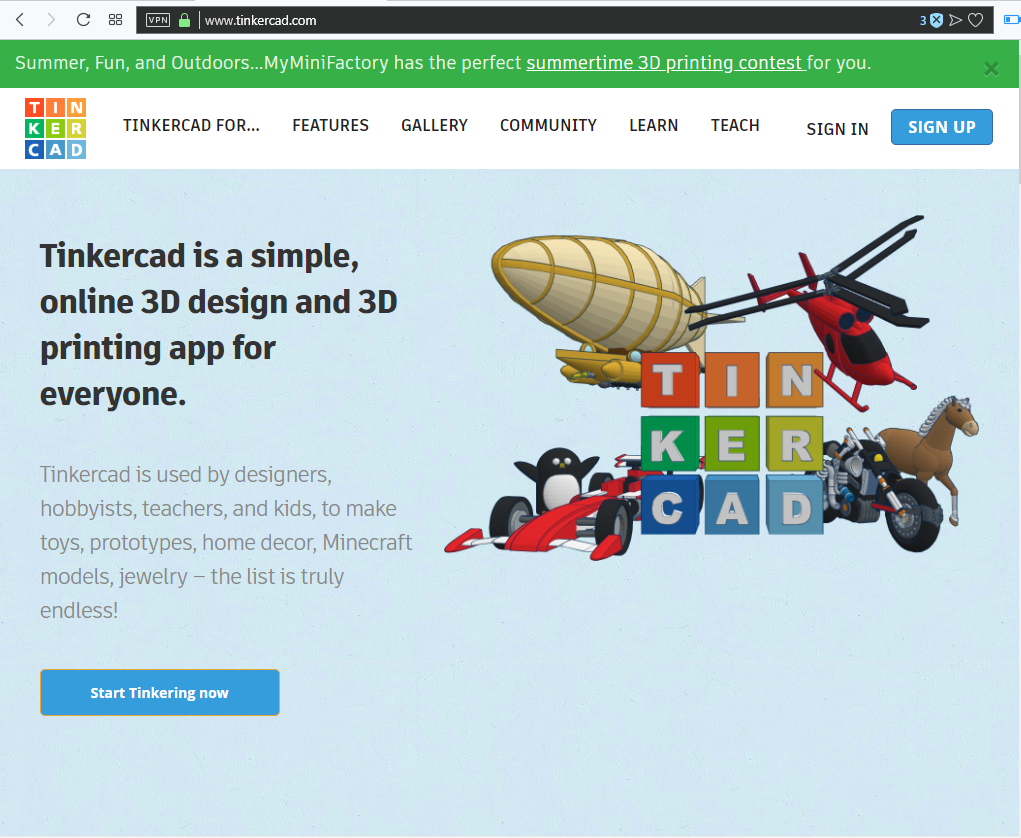
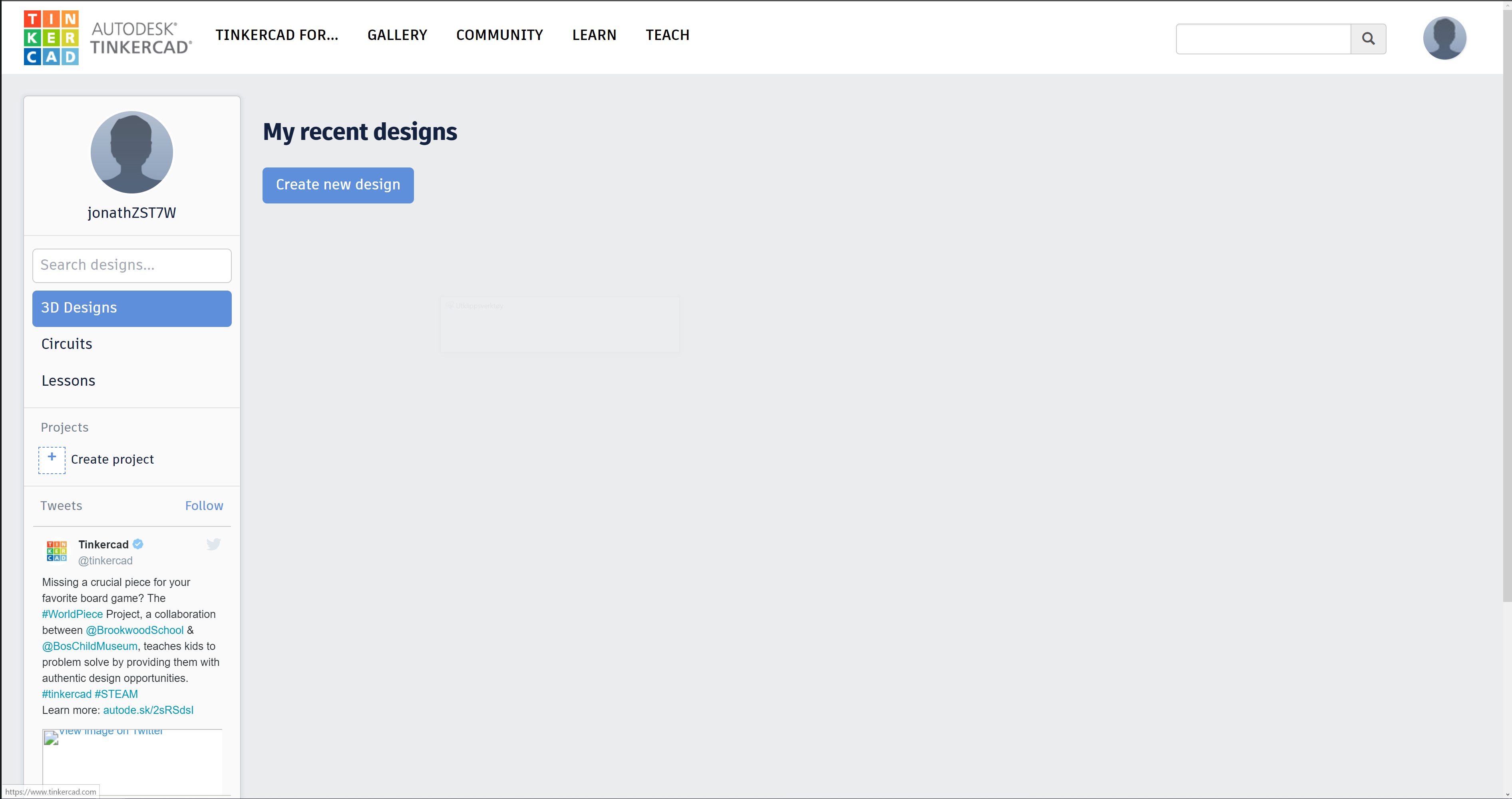
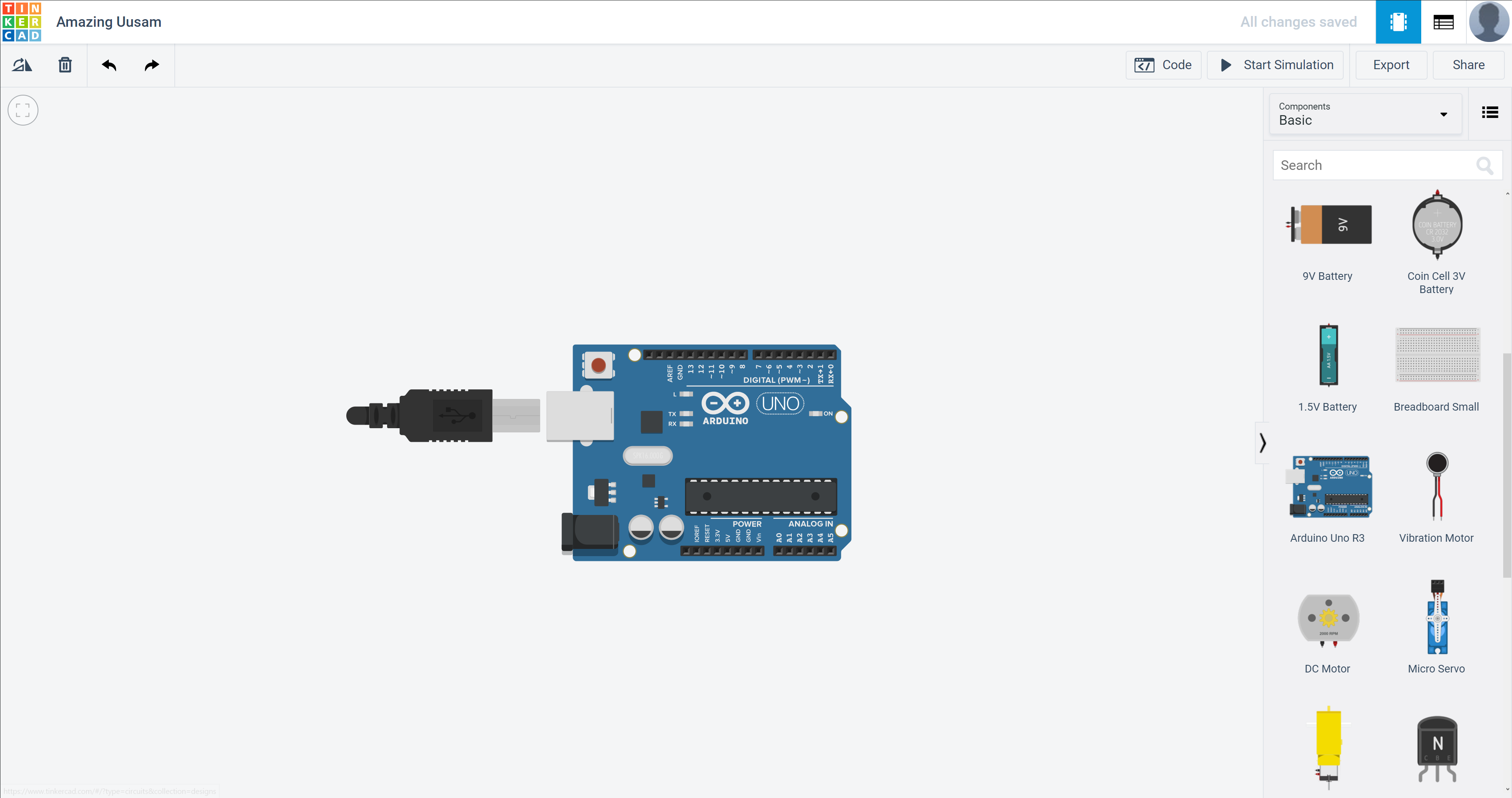
## TinkerCad EP 2

Hvordan bygge krets i nettleser + gjennomgang av Blinky og seriel funksjonen.

## Steg 1

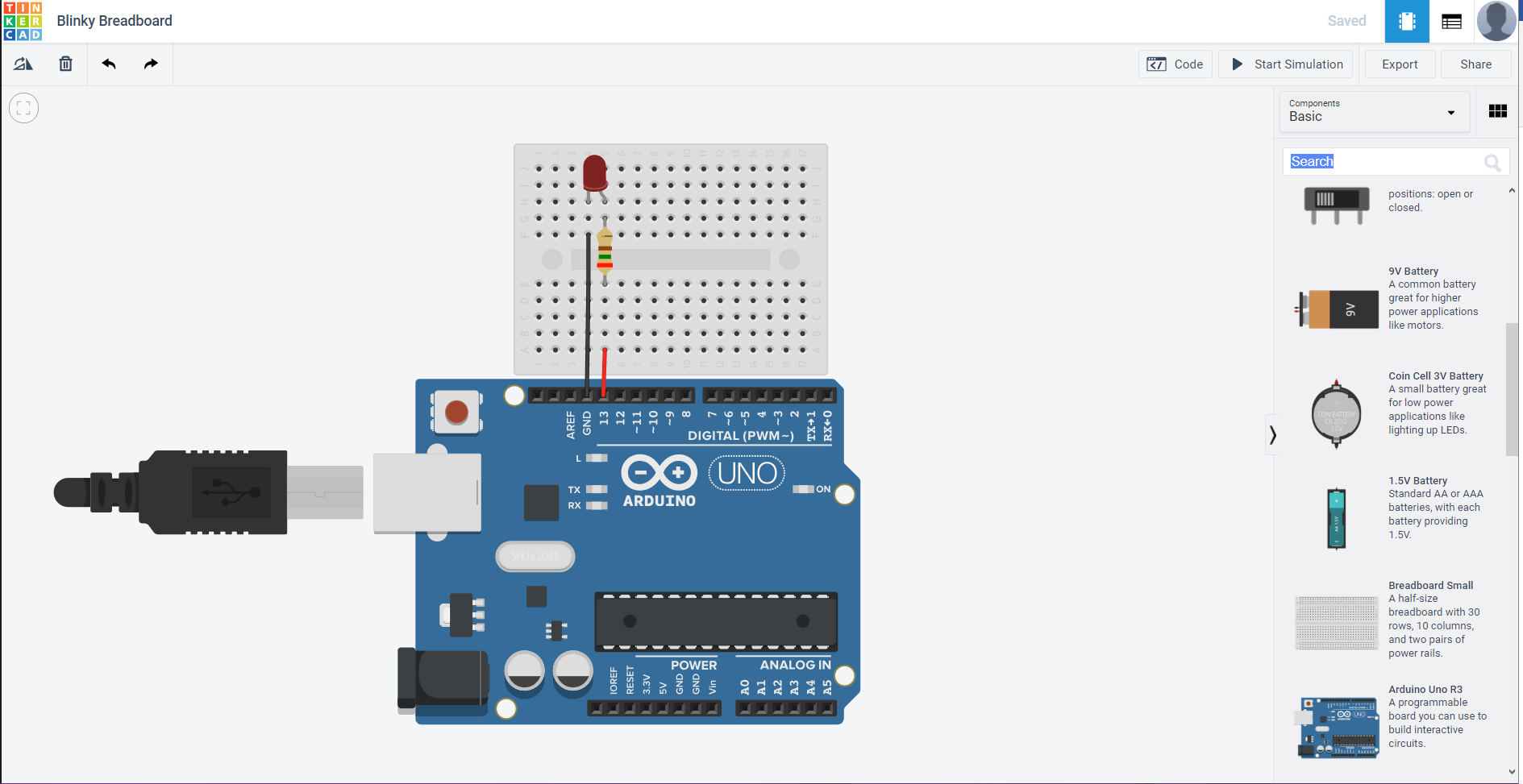
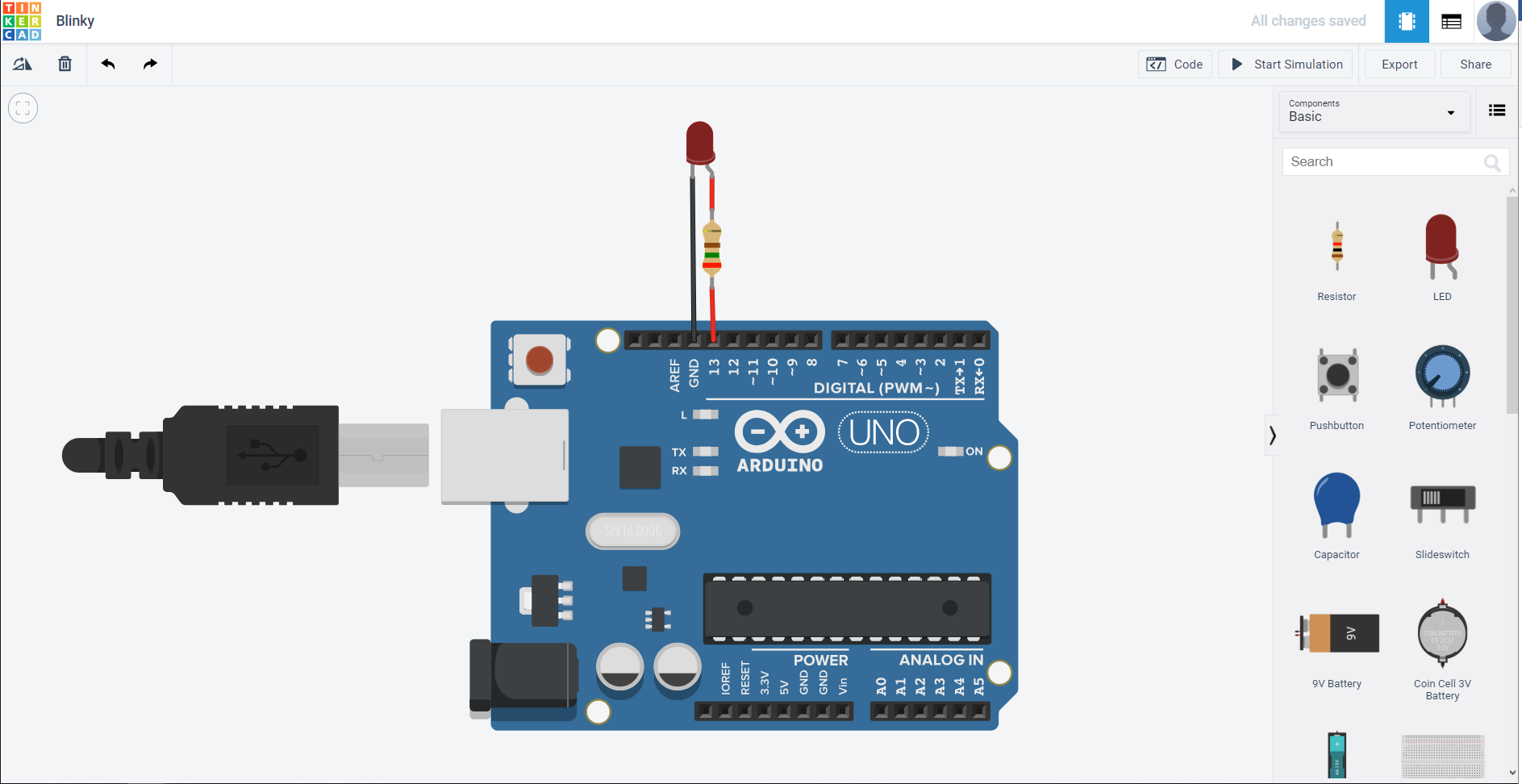
   
La oss starte med å gå til <https://www.tinkercad.com> og logge in. Opprett et nytt prosjekt eller en krets(cuircuit).   


Vi begynner med å sette ut Arduino Uno R3 på ‘’arbeidsbenken’’.  


For å bygge resten av kretsen må vi ha flere komponenter, om det er noe vi skall lage i virkeligheten også 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:

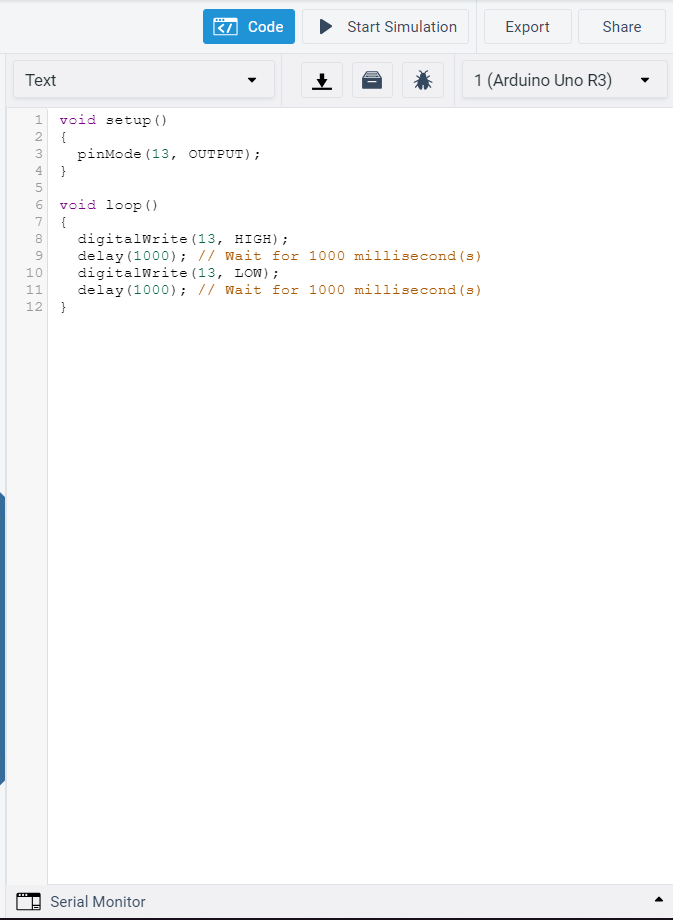
* Breadboard(protobrett)
* LED
* Resistor (motstand)
* Ledninger

Alle komponenten finner vi i komponentlisa på høyre side. (det er mulig og endre visningen av komponentlisa ved å klike på ene boksen over søkefeltet, mer informasjon om hver komponent vises). Under ser vi både med og uten protobrett, begge er riktig men protobrett

Når alt skal monteres på protobrettet er det viktig og sette LED riktig vei slik at riktig ben blir koblet til riktig port på Arduinoen. En LED har som oftest 2 bein, en anode og en katode. Det er viktig at disse blir koblet riktig eller så vill det ikke bli noe lys. Katoden går til jord, anoden til positiv spenning. Motstanden som kobles til kan ha en verdi på 250 Ohm for å beskytte LED’en.

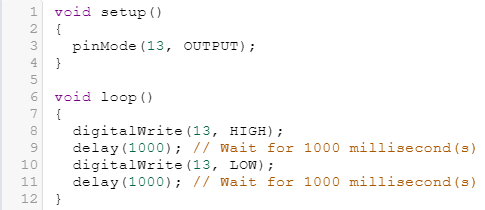
## Steg 2

  
La oss ta en titt på koden. Vi må klikke på Code knappen og velge text om det ikke alle rede er det som står som programmerings form. Det vil se noe ut som dette:  


Det vi har her er en ***Sketch*** og hva koden til Arduino blir kaldt. Så la os prate litt om hva dette egentlig er. Kode eller programmere er å fortelle en maskin hva den skall gjøre og det gjøres igjennom et programmerings-språk, til Arduino har vi Arduino kode som omtales som en ***Sketch***.

For att maskinen skall forstå hva som skal gjøres må koden vår konverteres til maskinkode, maskinkode er bare enere(1) og nuller(0) og omformingen fra programmeringsspråket som vi skjønner til maskinkode gjøres ved hjelp av en oversetter eller en kompilator.

Så hva består programmet vårt av?  
 Det er en rekke instruksjoner som kommer i formen av programlinjer, som vi ser i programmet vårt har vi mange linjer. Akkurat nå så er det 12 linjer. Programmet vårt har to deler som vi vil se lit på nå, vi starter med setup delen som er den første om blir gjort.



2

1

voide setup() er den første instruksjon og derfor også det første som leses fordi programmet leser sekvensielt, altså etter hverandre. Det er to ‘’blokker’’ og den første tilhører er ansvarlig for hva som skall gjøres klart og settes i gang så for koden kjører. Blokk nummer to er voide loop() og tar seg av hva som skall bli gjordt gjentatte ganger eller hva som skall gjøres hele tiden. Blokk 1, voide setup() inneholder nå i koden vår bare en program-linje og det er linje 3 som har pinMode (13, OUTPUT); skrevet, den er plassert imellom krøll-parentesene({ og }) som tilhører voide setup() noe som er et krav. Uten funker det ikke. Så la oss ta en titt på hva pinMode (13, OUTPUT); er: pinMode brukes for og bestemme hva som skall gjøres med en pinne. Pinnen velger vi ved å legge til et tall, vi bruker 13 i dette eksempelet, så da har vi bestemt at vi skall gjøre noe med pinne 13 så langt men hva, for og definere det må vi legge til en ting til. Nemlig INPUT eller OUTPUT. INPUT gir oss muligheten til å lese av en verdi og OUTPUT gjør det mulig å gi en verdi. Vi velger OUTPUT fordi vi skall skru et lys av eller på. Eller skru på strømmen eller av. Helt til slutt har vi en semikolon(;) som vi bruker for å si fra at programmlinjen er ferdig.

Blokk 2 voide loop() fungerer ganske likt som blokk en men med et stor unntak, den gjentar seg selv hver gang den er ferdig. Og her må også alle programlinje være innenfor klammeparenteser ({ og }).