

Anton S. Hermann
Frederik H. Nielsen

SCIENCE X

en gennemgang af 12 ugers spændende STEAM projekter

Science X - En gennemgang af 12 ugers spændende STEAM projekter!

©Anton Seistrup Hermann og Frederik Holm Nielsen i samarbejde med Projektleder Rasmus Grusgaard, 2022.

Grafisk tilrettelæggelse og illustrationer: Anton Seistrup Hermann og Frederik Holm Nielsen

Kopiering af denne bog er kun tilladt, hvis lovrammen for ophavsret overholdes. Undtaget her fra er citeringer og korte uddrag til opgaver og anmeldelser.

Relevante hjemmesider:

<https://www.sciencex.dk/>

<https://veluxfoundations.dk/da>

<https://www.fablabnordvest.dk/>

Arbejdet med dette projekt har taget mange timer og en stor indsats fra både (os) projektfacilitatorer - og deltagere.

Derfor vil vi gerne sige tak til de mange børn, forældre og lokale vejledere, der var med til at gøre de forskellige workshops til en realitet.

Derudover skal der også gå en stor tak til Villum Fonden for deres støtte af projekten. Uden dem ville der ikke findes noget ScienceX.

Til sidst vil vi sige tak til Rasmus Grusgaard for at være en ihærdig projektleder igennem projektet. Projekter som disse opstår ikke ud af ingenting, og dit forarbejde har nu lagt til grund for massere af læring omkring videnskab og teknologi.

- Anton & Frederik

Indhold:

- Introduktion til elektronik p.1-22
- Søbræt p.23-38
- Søbræt fortsat p.39-50
- PCB'er p.51-59
- Strømforsyning p.60-92
- Multimeter p.93-112
- Tesla Spole p.113-129
- MakeyMakey p.130-145
- MakeyMakey og Scratch Spil p.146-183
- Arduino 1 p.184-193
- Arduino 2 p.194-207
- Teachable Machine p.208-223

INTRODUKTION

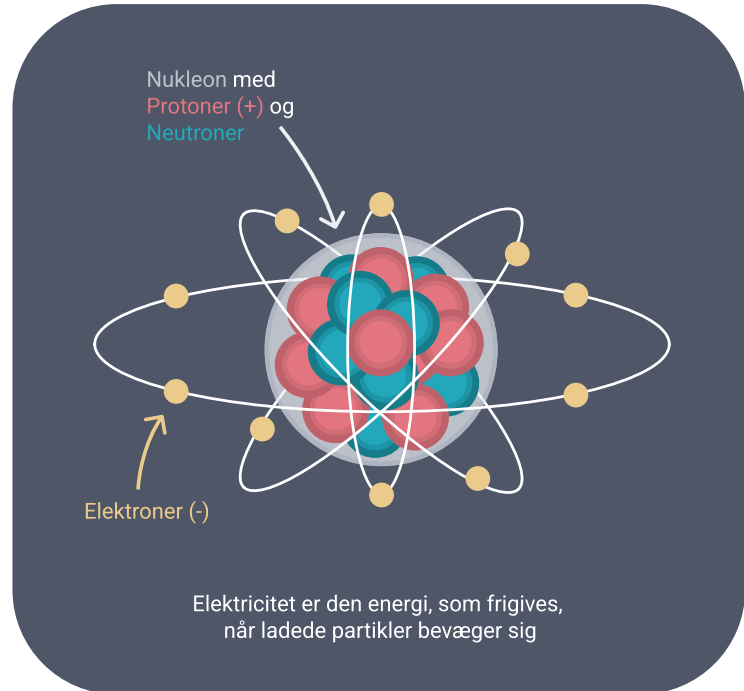
Komponenter og Teori

TEORI

Kort om elektricitet

Kernen kort og godt

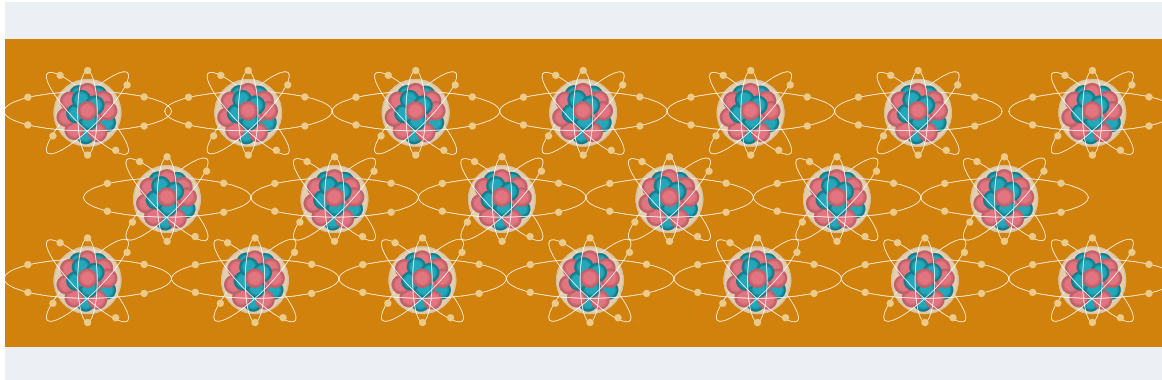
Før vi begynder at gennemgå de komponenter, man bruger i elektronik, skal vi lige forstå en lige smule om, hvad der sker på atom-niveau. Modellen viser en tilfældig atom-kerne med protoner og neutroner. Omkring kernen flyver der elektroner. Elektronerne er vigtige når vi snakker om elektricitet, da det er deres bevægelse der frigiver elektrisk energi.



Elektricitet kort og godt

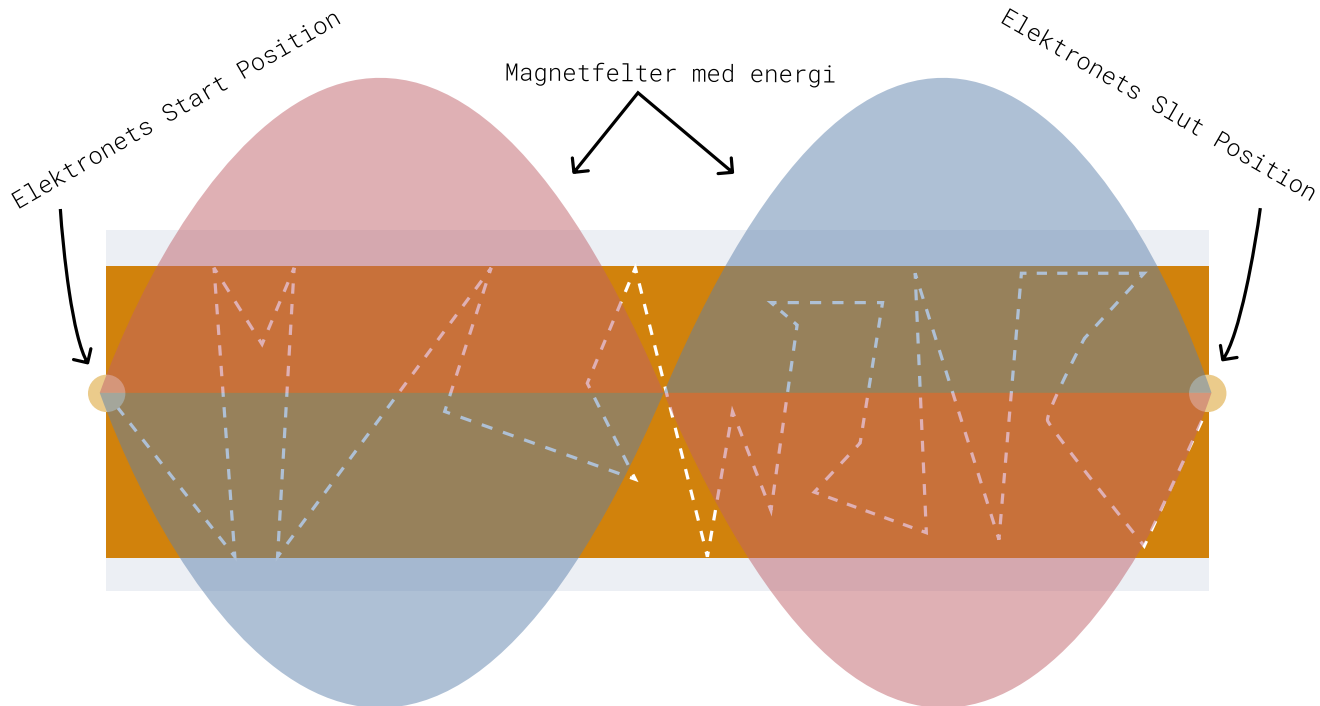
Mange tror, at elektroner bare "flyver" igennem ledninger med lysets hastighed, men i virkeligheden er deres rejse igennem en ledning ret lang. For elektroner drøner ikke bare igennem. De ændrer konstant retning!

De bevæger sig faktisk kun med få millimeter i sekundet igennem ledningen, men denne bevægelse skaber magnetfelter. Det er energien i magnetfelterne, som i virkeligheden er elektriciteten.



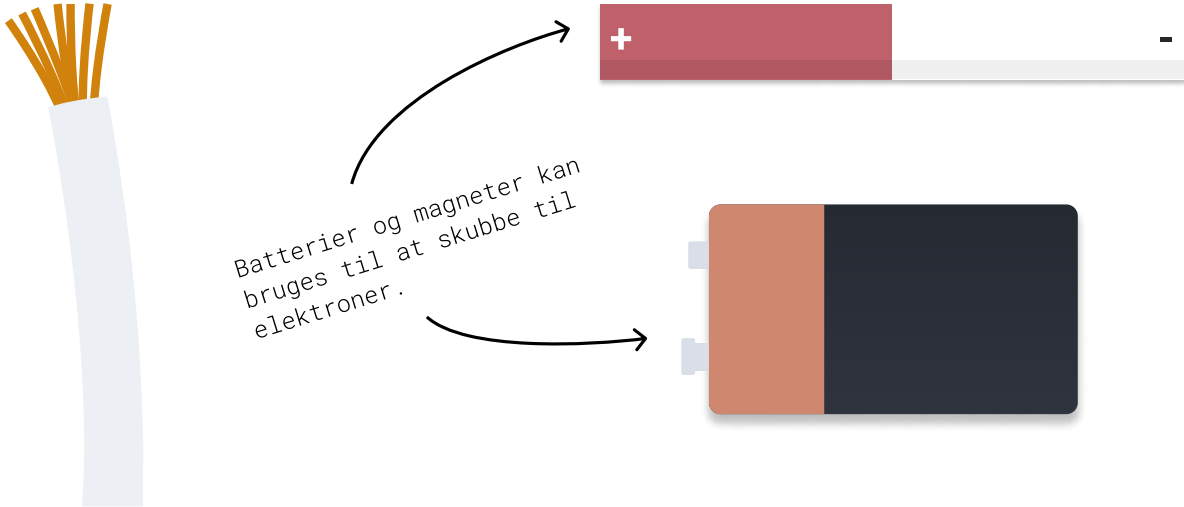
Elektricitet kort og godt

Figuren her giver en ide om hvordan elektronerne bevæger sig, og hvordan det altså får energien til at bevæge sig!



Elektricitet - Principper

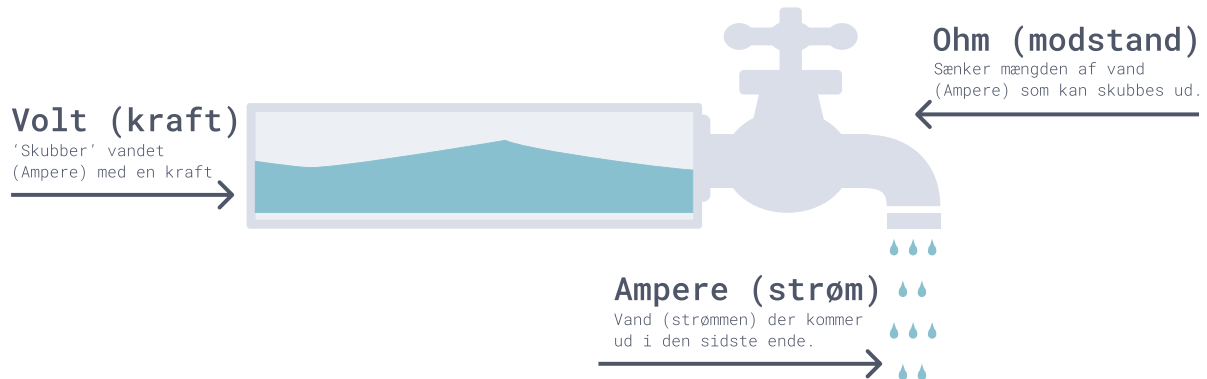
Elektroner laver dog ikke den her bevægelse igennem ledningen af tilfældige årsager. Der skal være noget, som motiverer dem til at starte den her bevægelse, og så skal det ikke være for 'svært' at bevæge sig igennem.



Elektricitet - Grundforståelsen

Vi beskriver det fænomen som 'skubber' elektroner igennem en ledning som spænding (enheden Volt) og det som holder dem tilbage som modstand (enheden Ohm).

Antallet af elektroner, som faktisk bevæger sig igennem ledningen, måles med enheden Ampere. Ampere beskriver antallet af elektroner som bevæger sig igennem en flade pr. sekund.

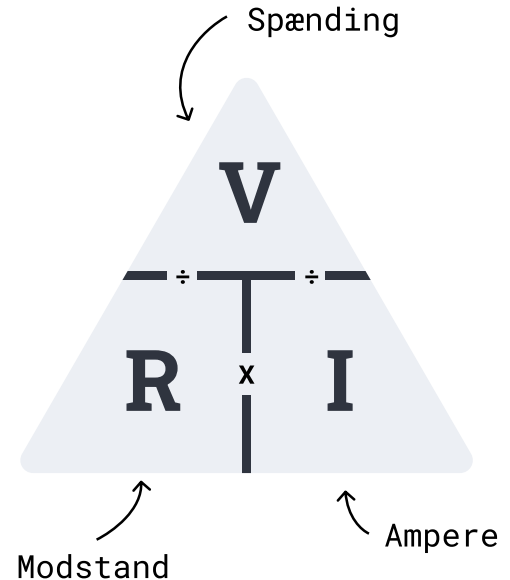


Elektricitet - Ohms Lov

Hvis man skal regne ud, hvor mange elektroner der bevæger sig igennem en ledning, kan man bruge Ohms Lov. Ohms lov beskriver sammenhængen mellem vores modstand, spænding og strøm-mængde.

Bare hold hånden over det du gerne vil udregne!

Hvis man forstår de her grundlæggende principper, er det meget nemmere at forstå nogle af de ting, vi ser i virkeligheden.

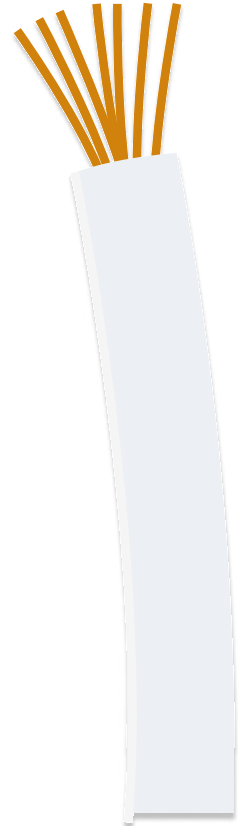


Elektricitet - Ohms Lov

Tag vores ledning for eksempel. Hvorfor er det, at strømmen løber igennem midten af ledningen men ikke igennem kanten?

Det er fordi, der ikke er så meget modstand i kobber, som er et materiale, vi (ofte) bruger til kernen i en ledning. Kobber er, det vi kalder, et ledende materiale.

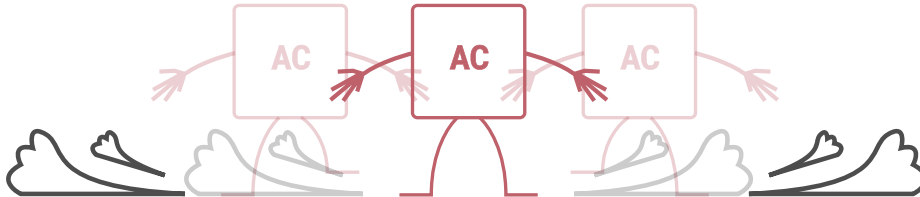
Og omvendt, så har gummi en meget høj modstand, som gør det næsten umuligt for elektronerne at bevæge sig igennem. Gummi er altså ikke et ledende materiale.



Elektricitet - To Typer

Med en grundlæggende forståelse for hvad der påvirker elektroners bevægelse igennem ledninger, så er det tid til at snakke om de to typer af strøm: jævnstrøm og væksselstrøm. Jævnstrøm er når elektronerne bevæger sig i én retning, og væksselstrøm er når de bevæger sig frem og tilbage.

Væksselstrøm eller på Engelsk 'Alternating Current' (AC)



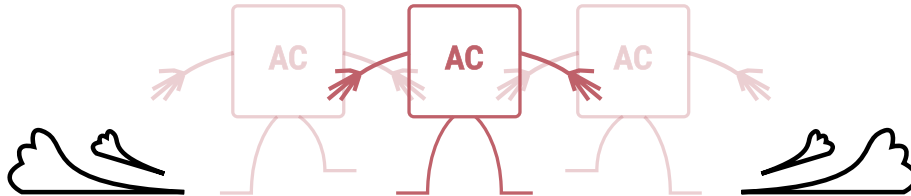
Jævnstrøm eller på Engelsk 'Direct Current' (DC)



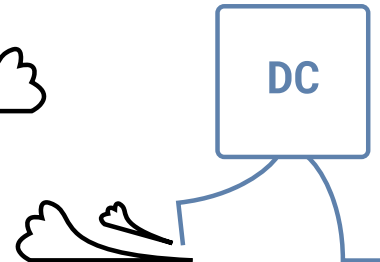
Elektricitet - To Typer

Det som for alvor er interessant, er at energien bevæger sig på samme måde uafhængig af hvilken retning elektronerne bevæger sig. Det er lidt teknisk hvorfor det lige er situationen, men tænk på det som, at bevægelse skaber elektrisk energi, ligemeget om elektronerne bevæger sig frem og tilbage - eller bare lige ud.

Vækselstrøm eller på Engelsk 'Alternating Current' (AC)

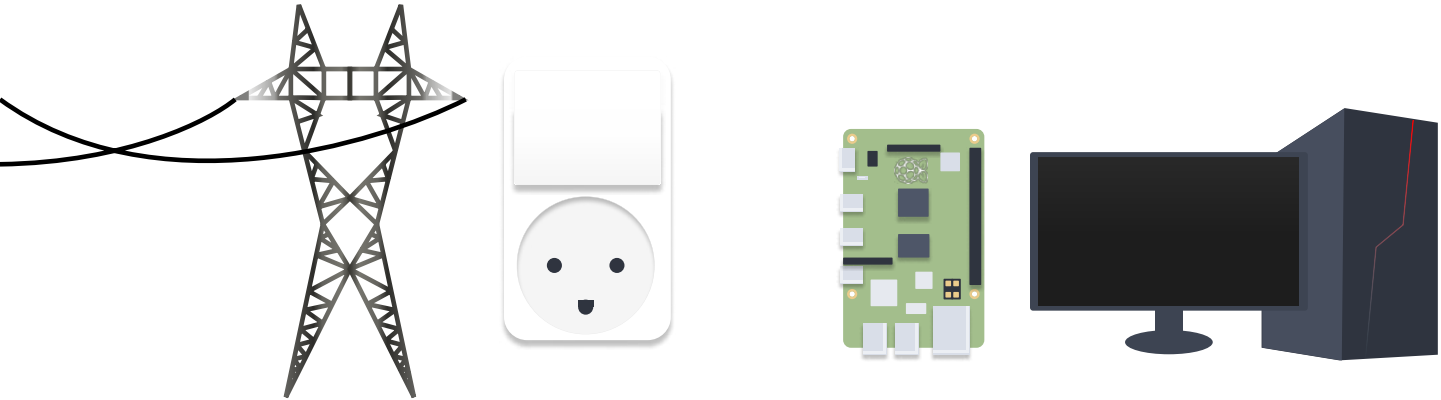


Jævnstrøm eller på Engelsk 'Direct Current' (DC)



Elektricitet - To Typer

Men, men, men... Selvom energien bevæger sig i samme retning, så er det ret vigtigt at kende forskellen på de to typer Elektricitet, og hvornår man bruger den ene og den anden. Grundlæggende så bruger man væksselstrøm når man skal bevæge strøm over lange afstande og ind i huse (stikkontakter), hvorimod jævnstrøm er mere anvendt i ting som computere, motorer, osv.



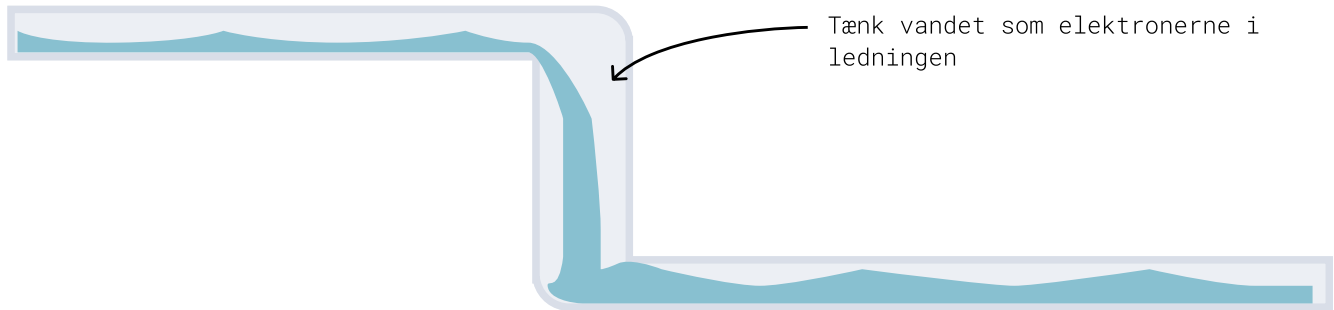
KOMPONENTER

De første par stykker

Komponenter

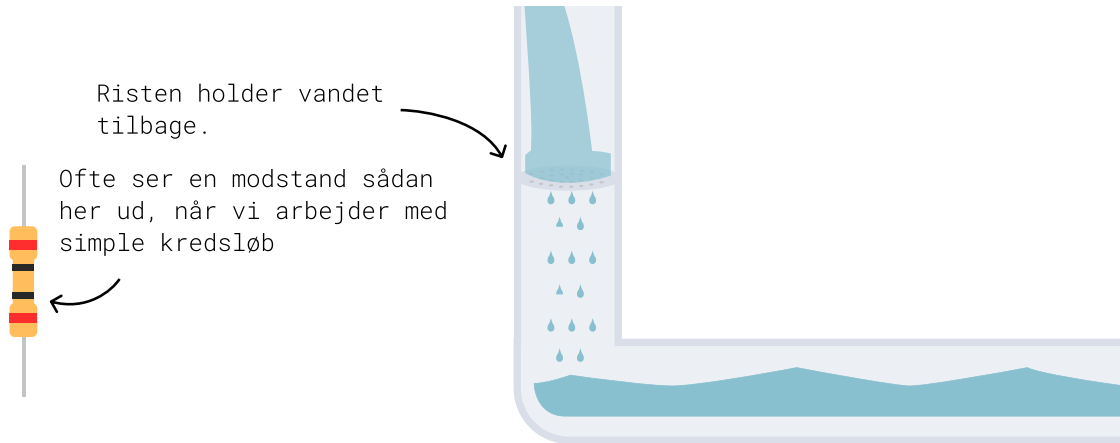
Elektriske komponenter er de byggeklodser, vi bruger, når vi bygger større elektriske kredsløb. Nogle komponenter er meget avancerede, og andre er rimelig simple.

I det her afsnit kommer vi til at kigge på nogle simple og meget anvendte dele. For at hjælpe forståelsen af, hvordan de virker, kommer vi til at bruge et system af vandrør, som et billede for et elektrisk kredsløb.



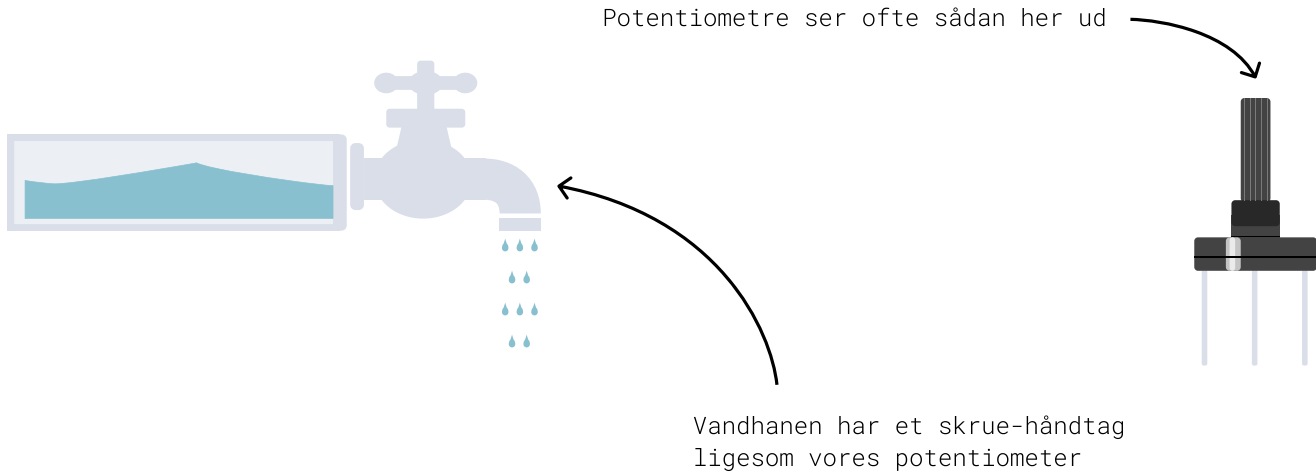
Modstande

Som navnet lidt indikerer bruger man modstande til at tilføje mere modstand. I et system med vand kan man forestille sig en rist. Risten hæmmer vandets evne til at flyde igennem røret. Man kunne også forestille sig at putte et mindre rør på, som på samme måde ville hæmme vandet.



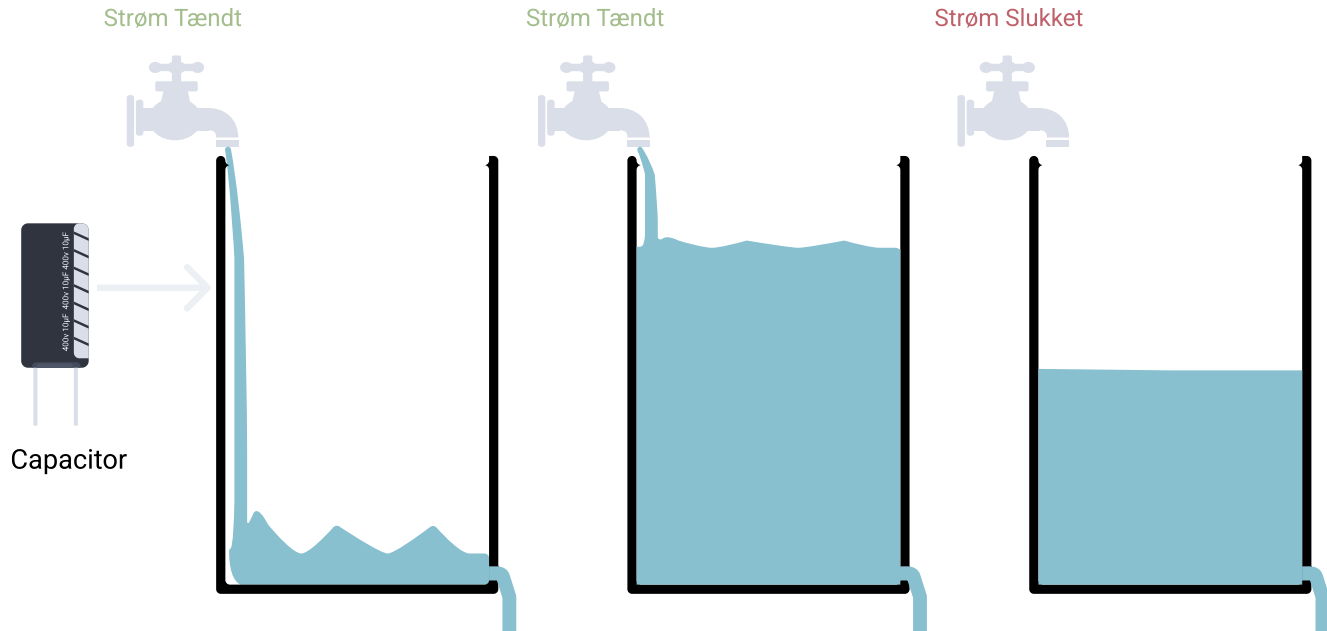
Modstande 2

På nogle modstande kan man ændre hvor meget modstand, der skal være i kredsløbet. De er kendt som potentiometre og fungerer ligesom en vandhane.



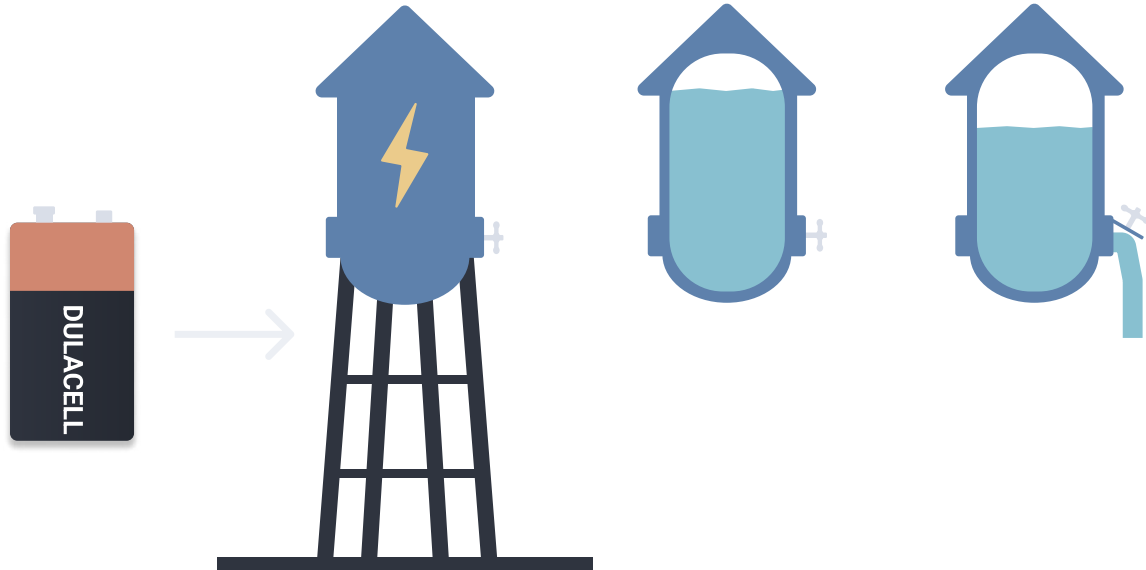
Kondensator

Kondensatorer kan gemme på strøm i et kort stykke tid. De er lidt ligesom batterier, men mængden af energi de holder på er meget mindre end batterierne. Forestil jer en tank med vand, som fungerer på følgende måde.



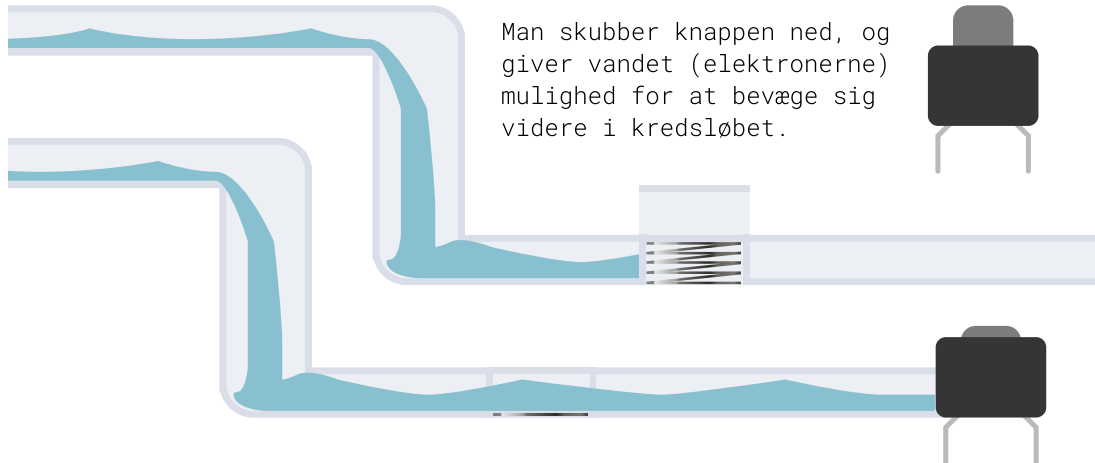
Batterier

Og lad os da bare tage batterierne. Batterierne har en stærk elektrisk energi, som vi kan bruge til at holde gang i vores kredsløb. Tænk på det som et vandtårn i relation til vores kondensator, som er en lille beholder.



Knapper

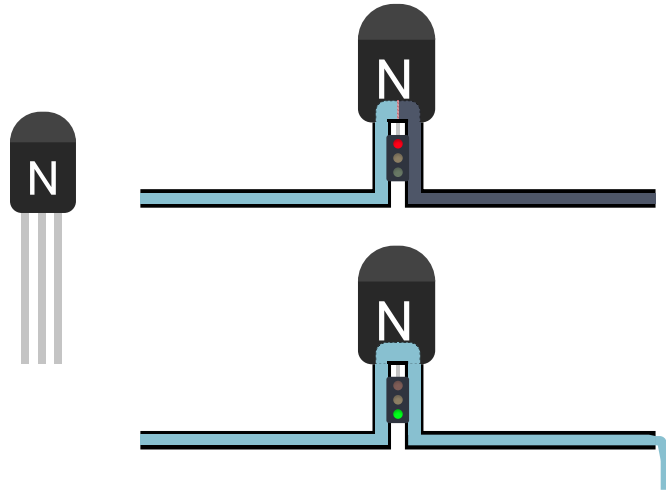
Knapper kommer i mange former og slags. Den mest almindelige er en tryk-knap. Den virker ved, at man trykker den ned og derved fuldender kredsløbet.



Transistorer

Transistorer virker lidt ligesom knapperne, men istedet for at man trykker noget ned, som fuldender kredsløbet, bruger man strøm til at aktivere den. Tænk på den som et lyskryds for vores strøm af vand! Transistorer kommer i mange former og slags, og er faktisk ret komplicerede. Men det betyder ikke, at du ikke kan bruge dem som nybegynder!

Ofte, er der pinde til vores elektroner, og så en pind som vi kan bruge til at styre om de må komme igennem eller ej

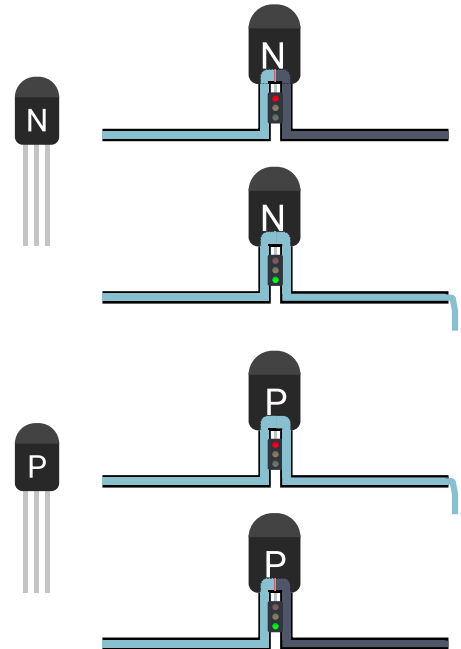


Transistorer fortsat

De to mest grundlæggende typer af transistorer er NPN transistorer og PNP transistorer

NPN transistorer holder normalt strømmen (vandet) tilbage, når den ikke får signal til midterbenet. Når midterbenet er aktiveret flyder strømmen igennem.

PNP transistorer virker modsat af NPN transistoren. Der løber normalt strøm igennem transistoren, men når midterbenet får signal, bliver der blokeret for strømmen.



Dioder

Dioder ensretter strøm. Det betyder, at strømmen kun kan bevæge sig i en retning igennem dem. Nogle af dem lyser, når strøm løber igennem dem. Dem kalder vi lysemitterende dioder (LEDer).

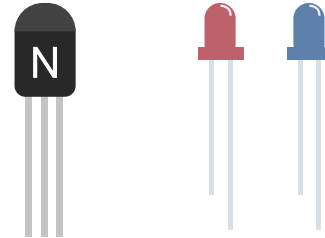
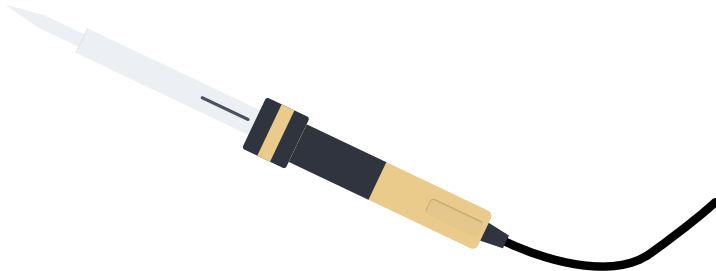


SØMBRÆT

Lodning for Begynderere

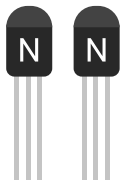
Hvorfor Søbrættet?

Søbrættet giver kendskab til, hvordan man lodder og skaber simple elektriske kredsløb. Med søbrættet som grundlag kommer fremtidige projekter til at være genkendelige.



Hvad skal du bruge?

2 x NPN Transistor



8 x Søm



1 x 9V Batteri



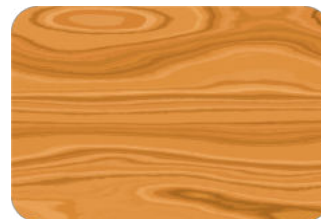
2 x 10k Ohm



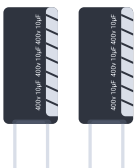
2 x 470 Ohm



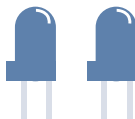
1 x Bræt



2 x Kondensator



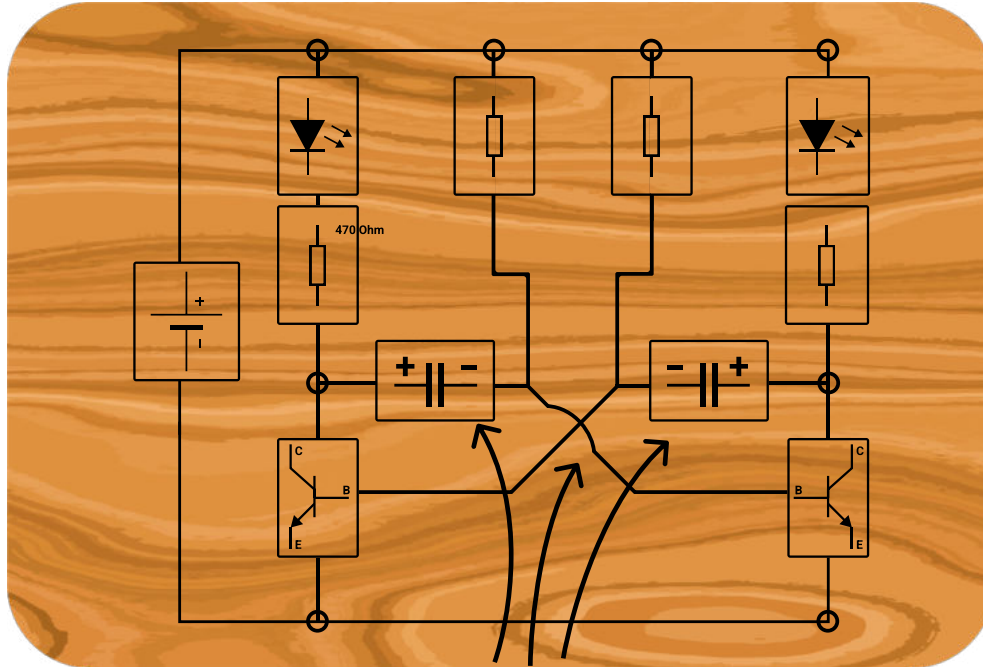
2 x LED



Masser af ledning

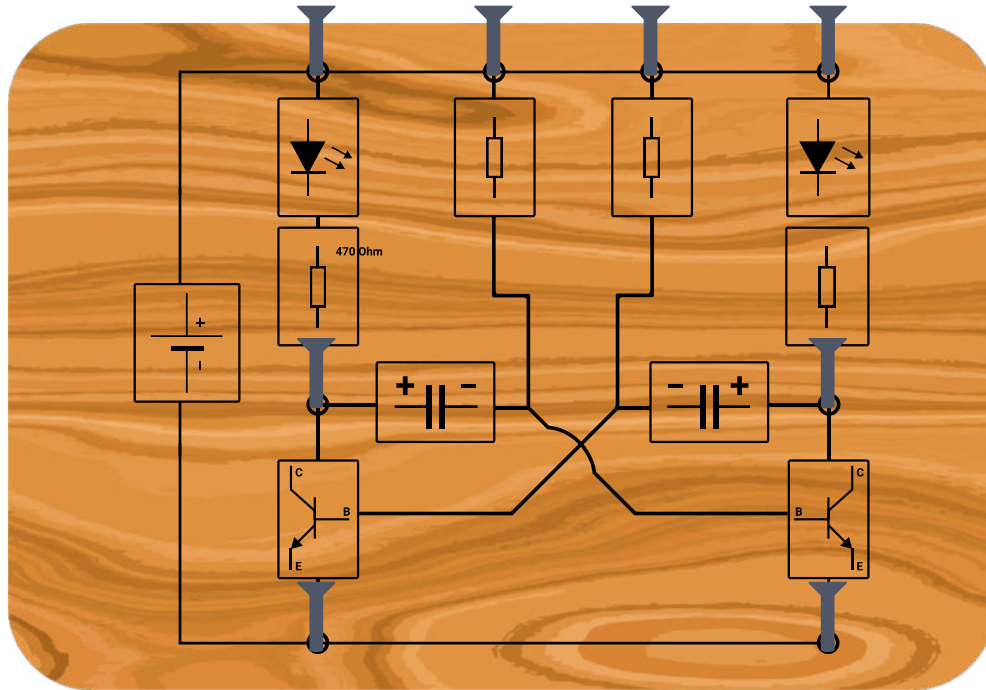


Ø: Tegn symbolerne på jeres bræt

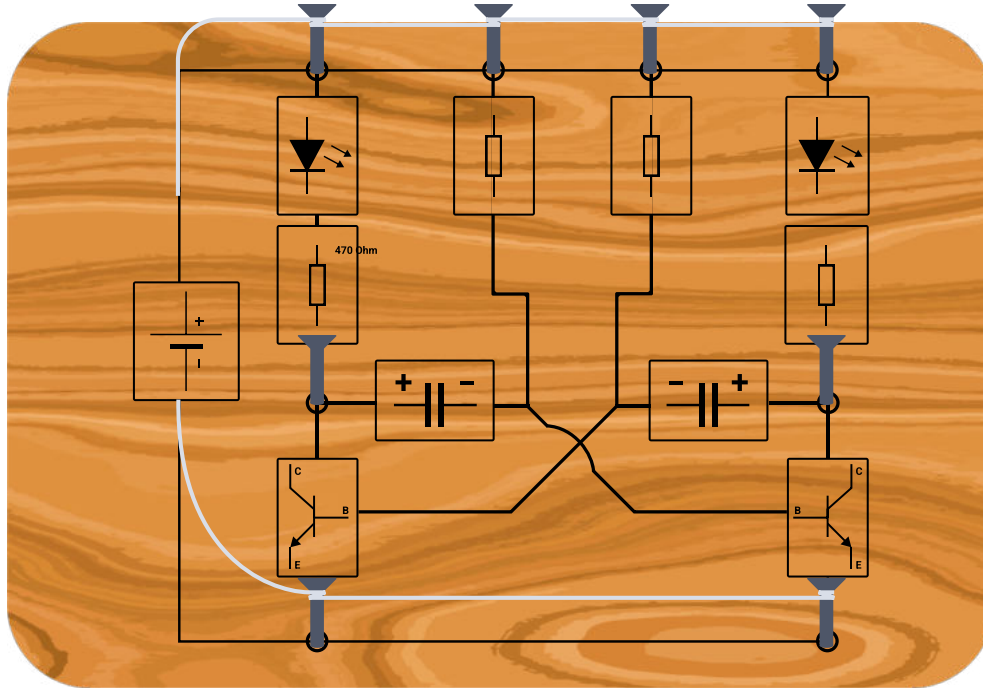


Det er vigtigt at i tegner, at ledningen krydser, og at i vender +/- rigtigt for kondensatorerne

1: Slå søm i de steder hvor der er cirkler



2: Lod de øverste og nederste søm sammen



I kan vikle tråden omkring eller lave en krog og lodde

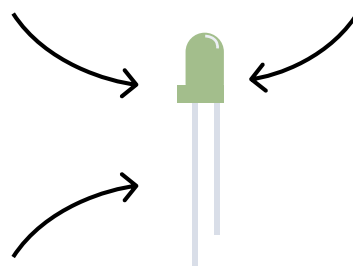
Modstande virker begge veje



LEDer skal vendes rigtigt

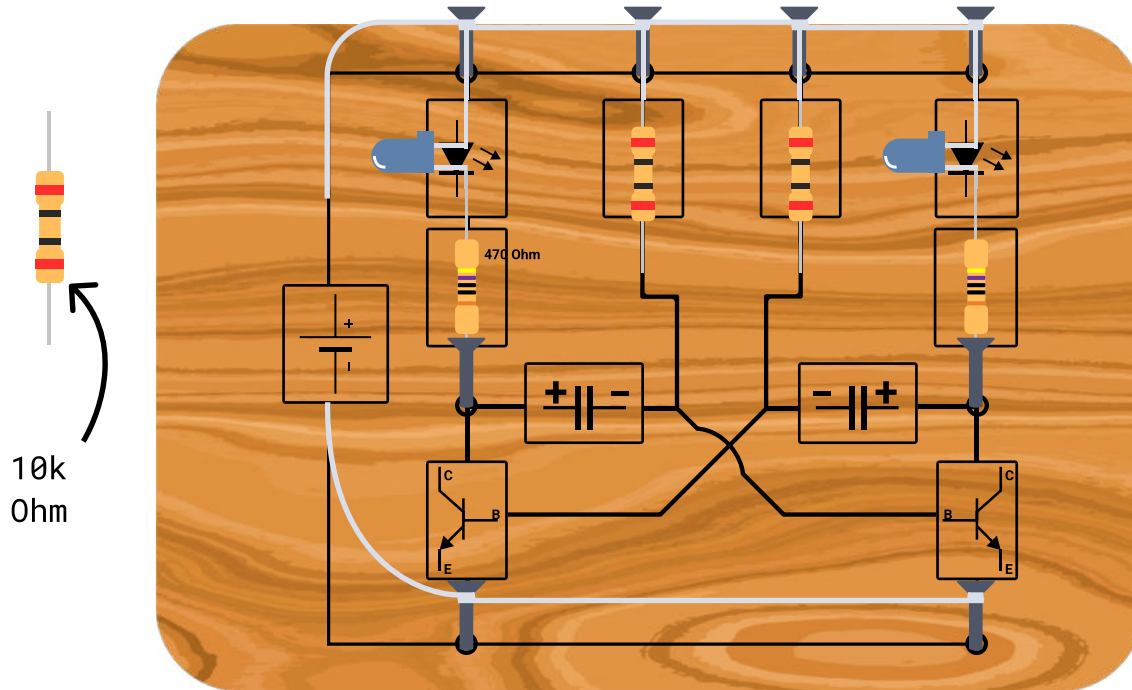
Den runde side er Anoden ("+"), den positive side

Den flade side er katoden ("-"), den negative side

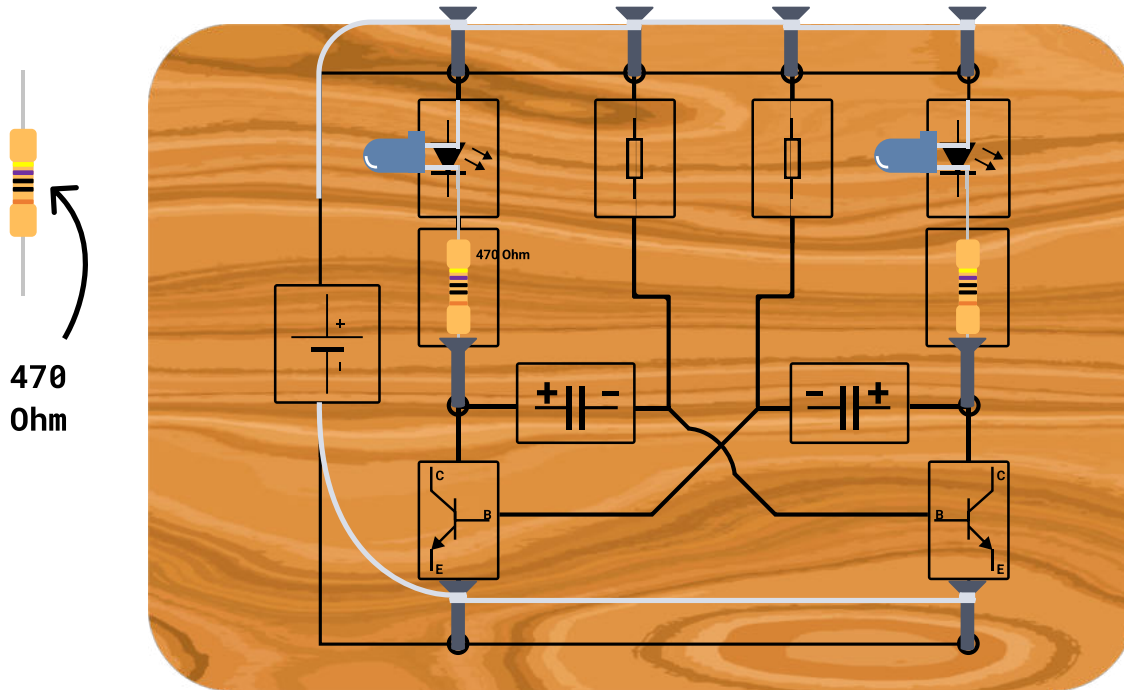


Det lange led på en LED viser også den positive side.

4: Lod nu vores 10k (12k) ohm modstande på



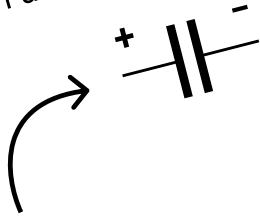
3: Lod LED'er og 470 ohm modstande



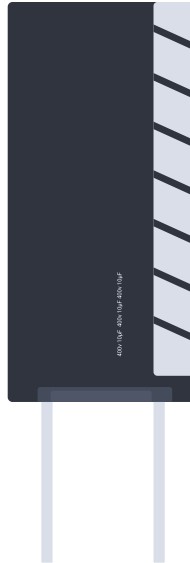
Husk retningen af LED'en

Kondensatorer SKAL vendes rigtigt

Strømmen skal løbe
fra + imod -

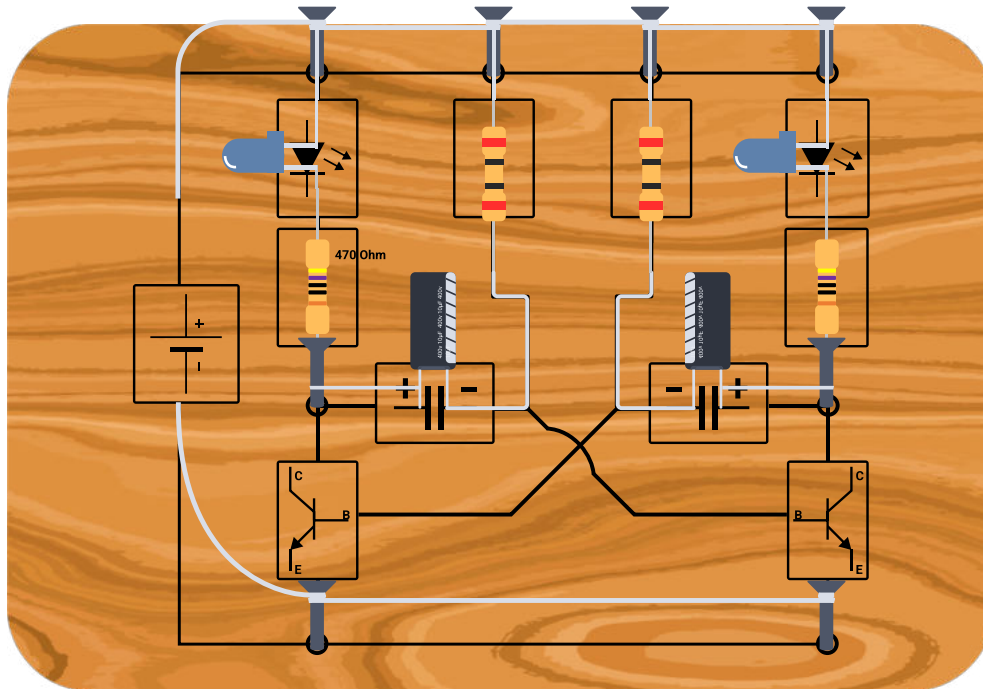


Hold øje med symbolet
på tegningen!



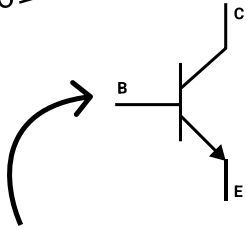
Kig efter en
markering på siden ("-")
eller et +/- symbol

5: Lod nu vores kondensatorer og de to 10k modstande



Transistorer skal også vendes rigtigt

Strømmen skal løbe
fra C imod E (med mindre den er
bi-polar)



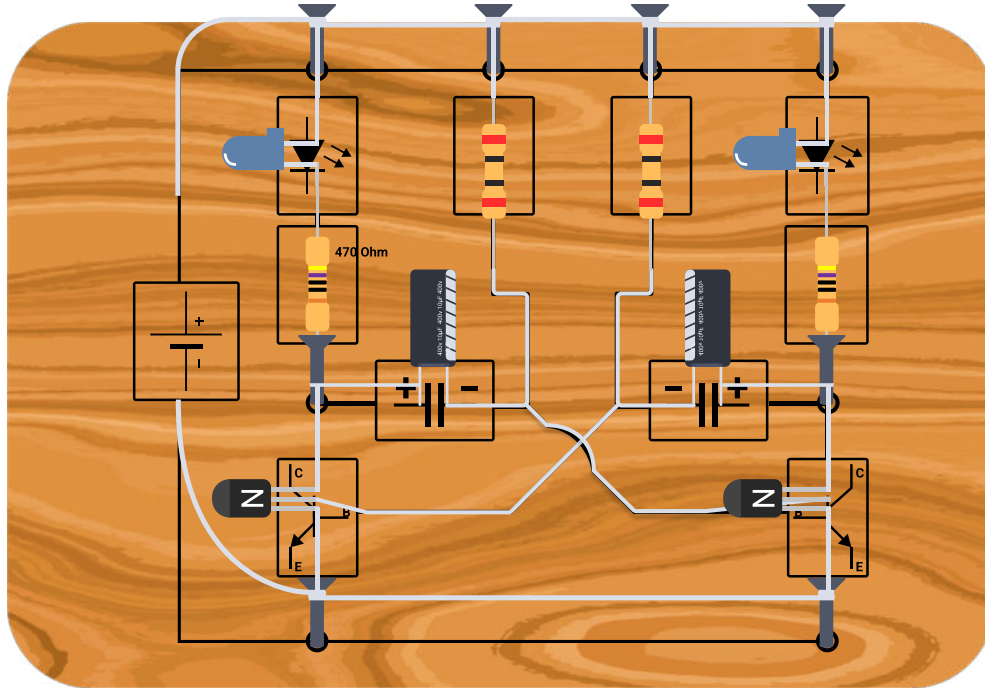
Kig efter hvilken
ledning der skal
loddet til B!



I skal slå jeres model
op og finde ud af hvilke
ben der er C, B og E

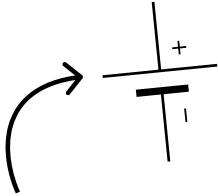
C B E

6: Lod til sidst transistorene på og afslut kredsløbet

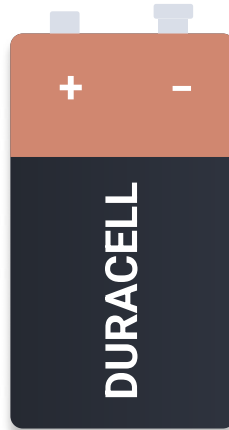


Strømforsyningen (batteriet) skal også vendes rigtigt

+ Skal være øverst
- Skal være nederst

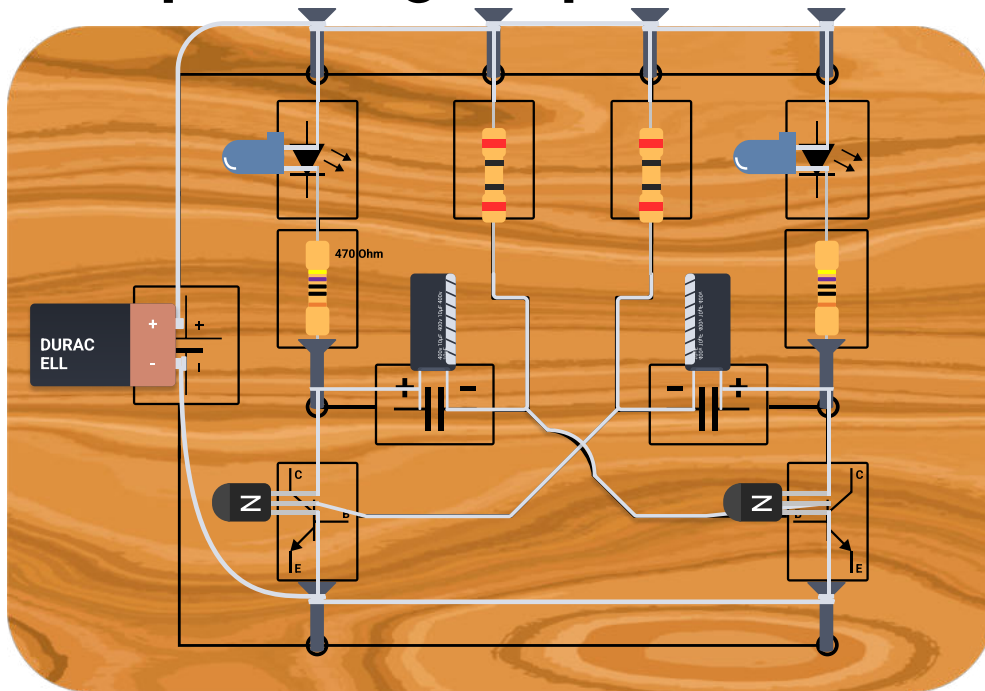


Kig på symbolet på tegningen



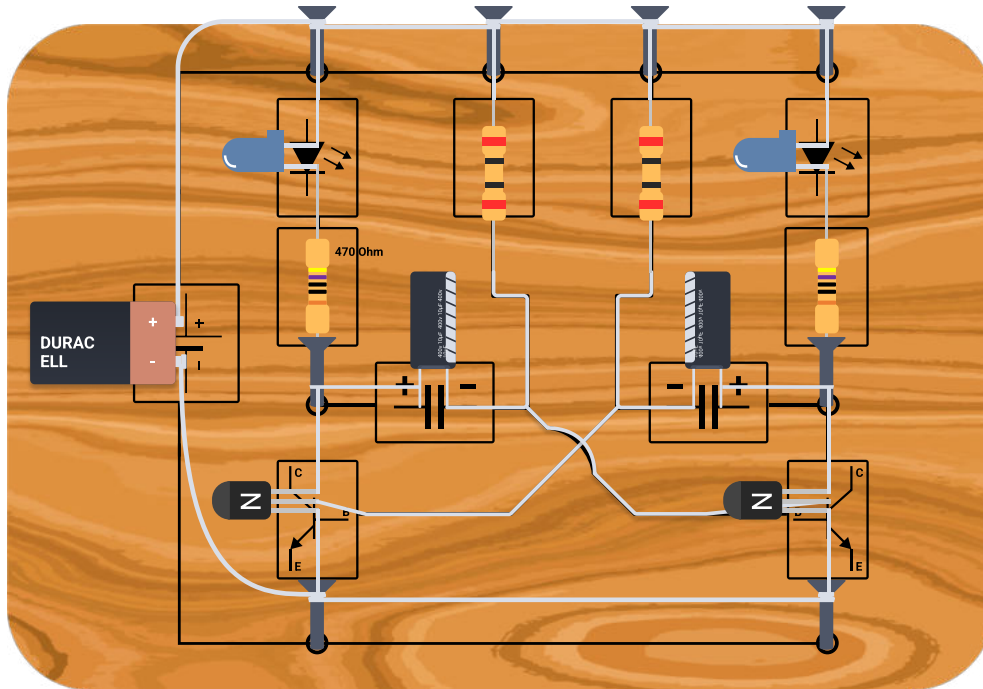
Jeres batteri burde have symboler på sig. Ellers kig på formerne!

7: Tilslut et batteri til jeres kredsløb ved at holde ledningerne på + og - polerne



Hvad i har lavet nu, bliver kaldt en astabil multivibrator!

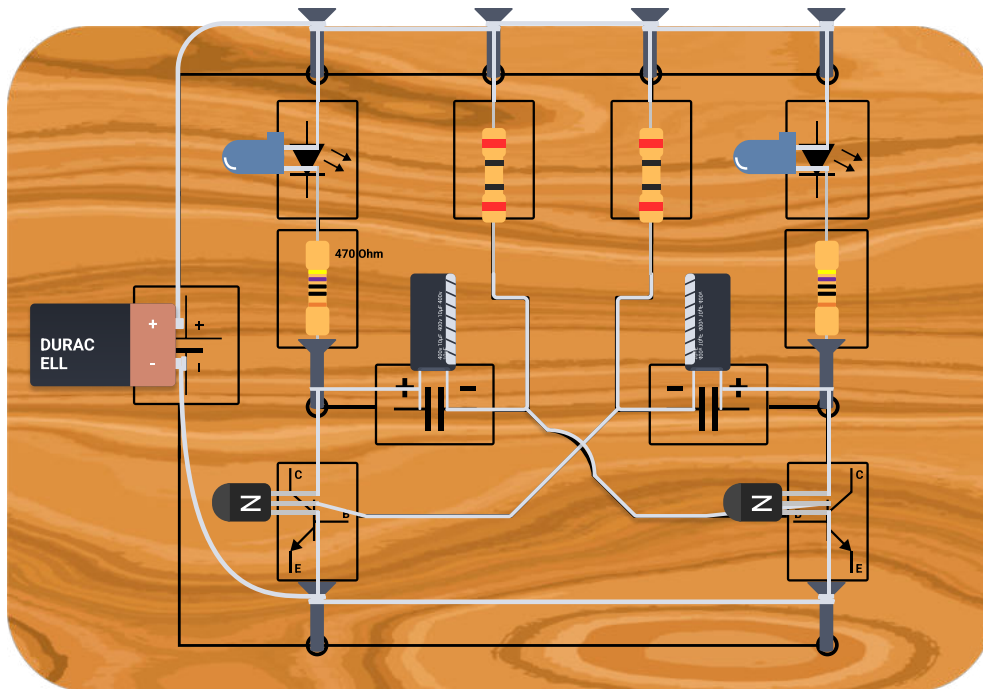
Strømmen løber rundt i systemet og tænder og slukker de to LEDer på skift. Med en større modstander eller kondensatorer, ville LEDerne blinke langsommere!



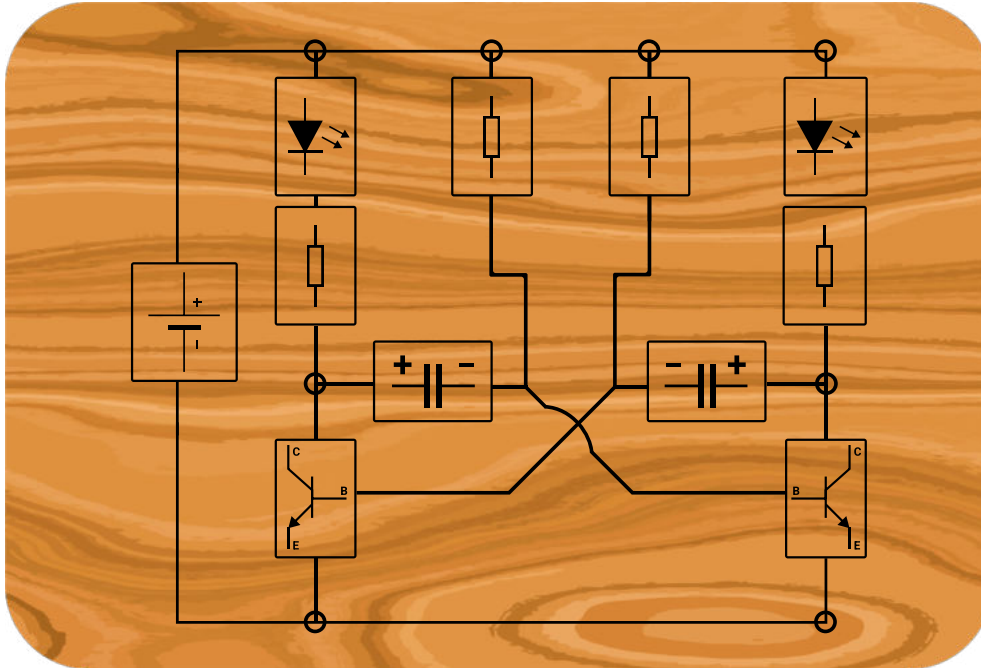
SØMBRÆT FORTSAT

**Endnu en fantastisk
sømbært øvelse!**

Lige nu ser jeres sømbræt sådan her ud

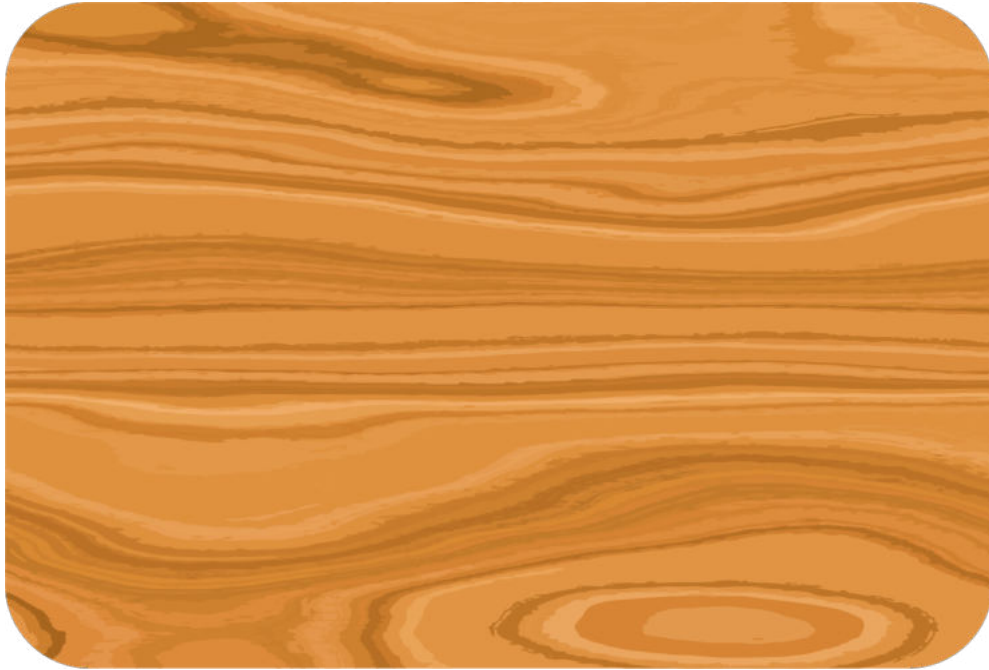


(Mulighed) Aflod Jeres dele, og fjern sømmene

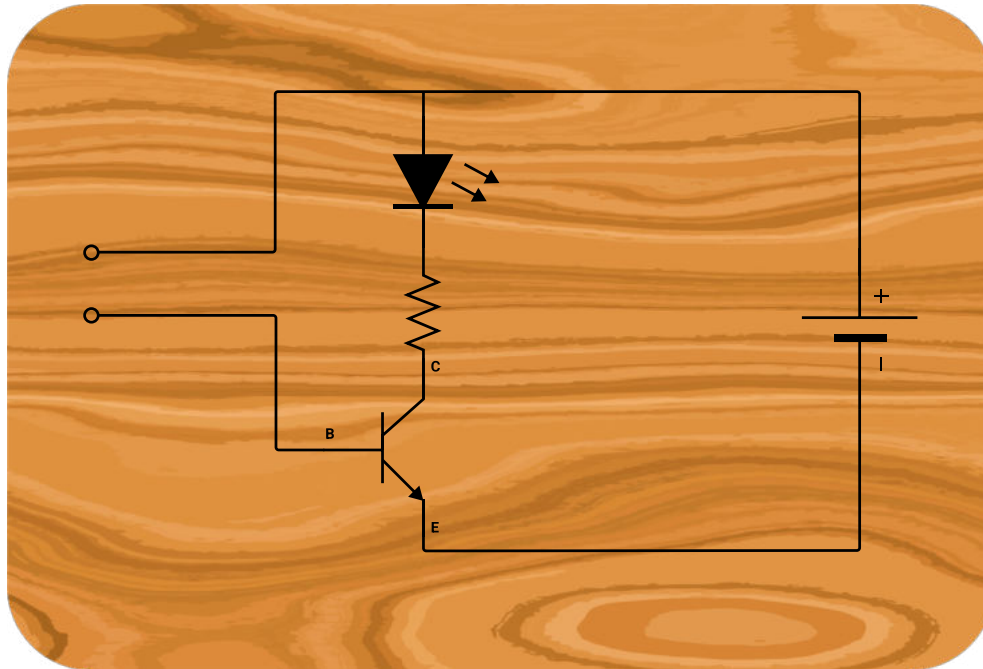


Prøv at sortere jeres dele, så i ved, hvor de forskellige modstande er.

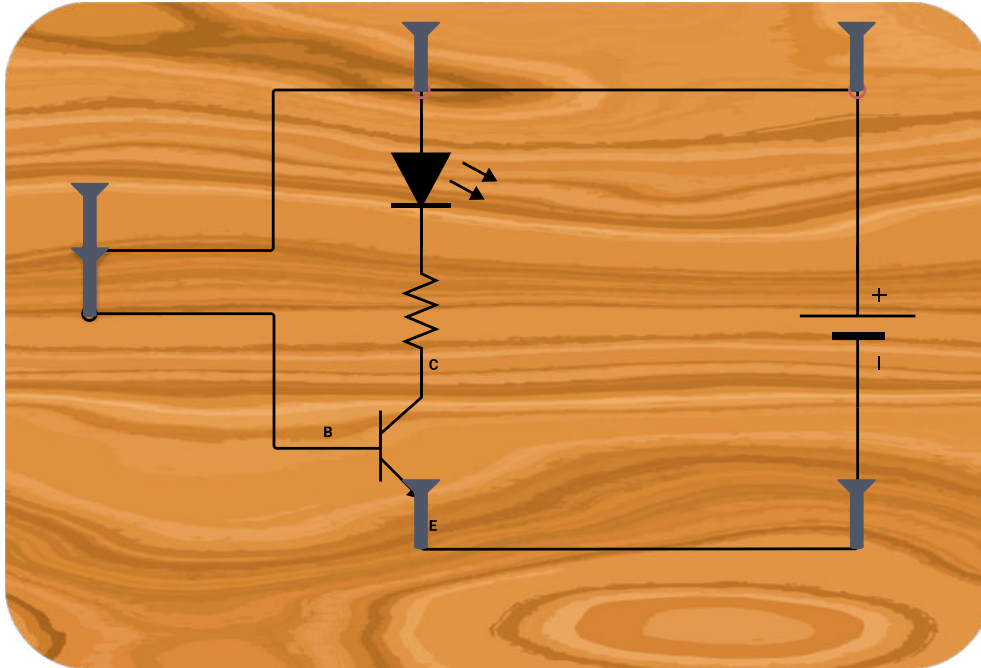
Vend jeres bræt om, og så er i klar!



Tegn symbolerne på jeres bræt/
papir

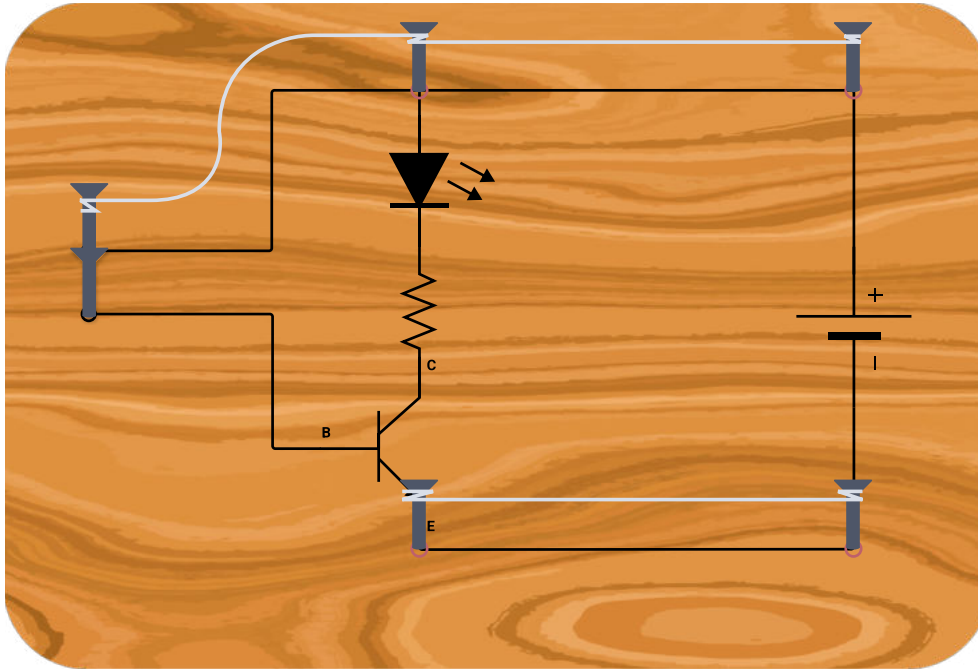


(Mulighed) Sæt søm i

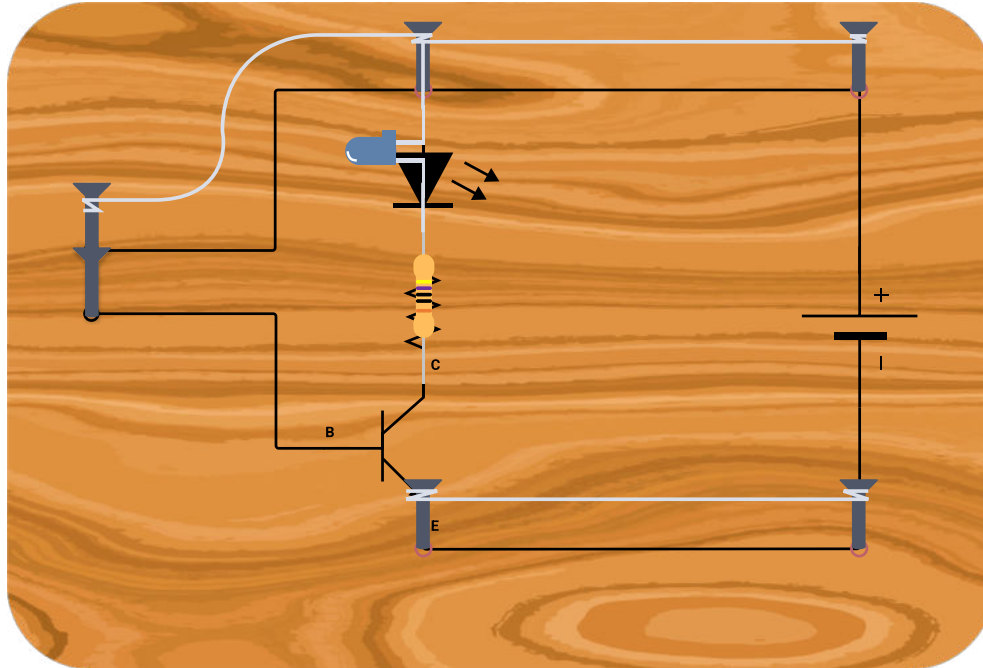


Det er op til jer selv at bestemme om i vil bruge søm som "loddestolper"

Lod ledninger på sømmene

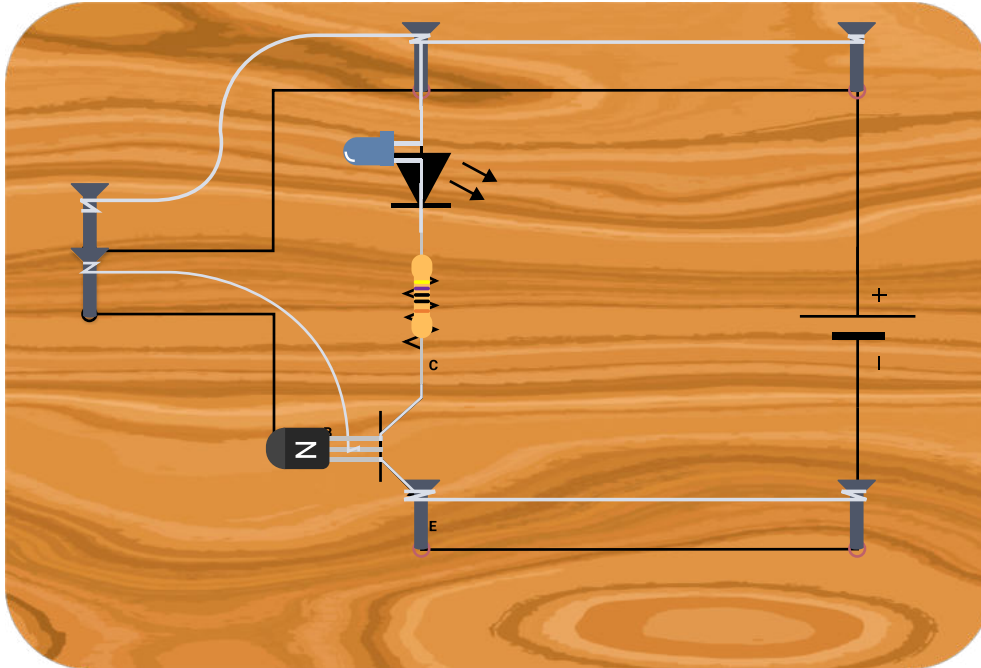


Lod LED og modstand på sømmet



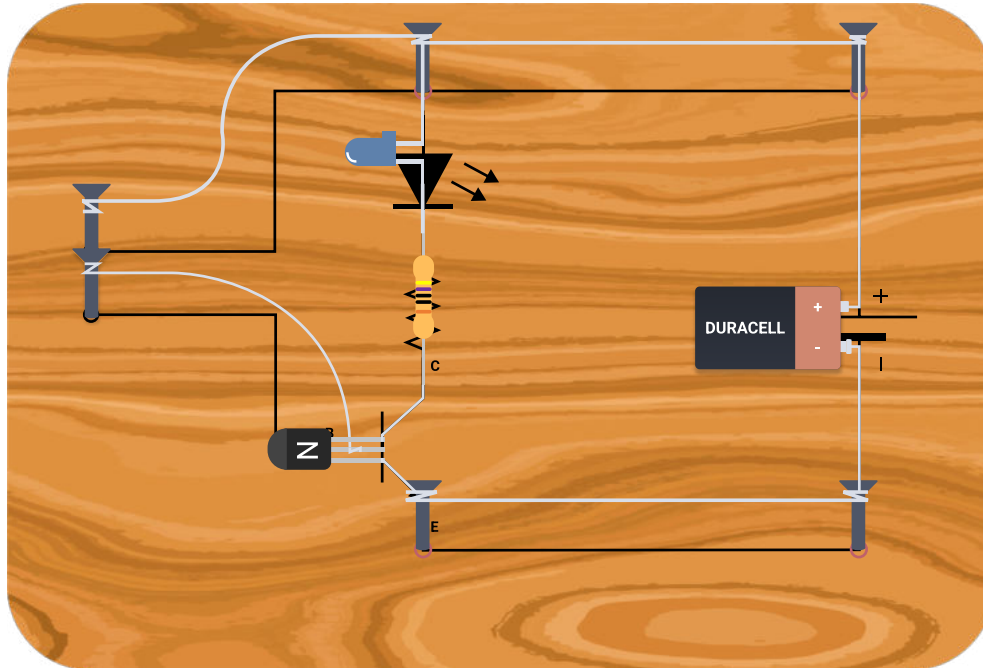
I kan evt. først lodde LED og modstand sammen, og så lodde LED'en på sømmet

Lod transistor til modstanden og sømmene



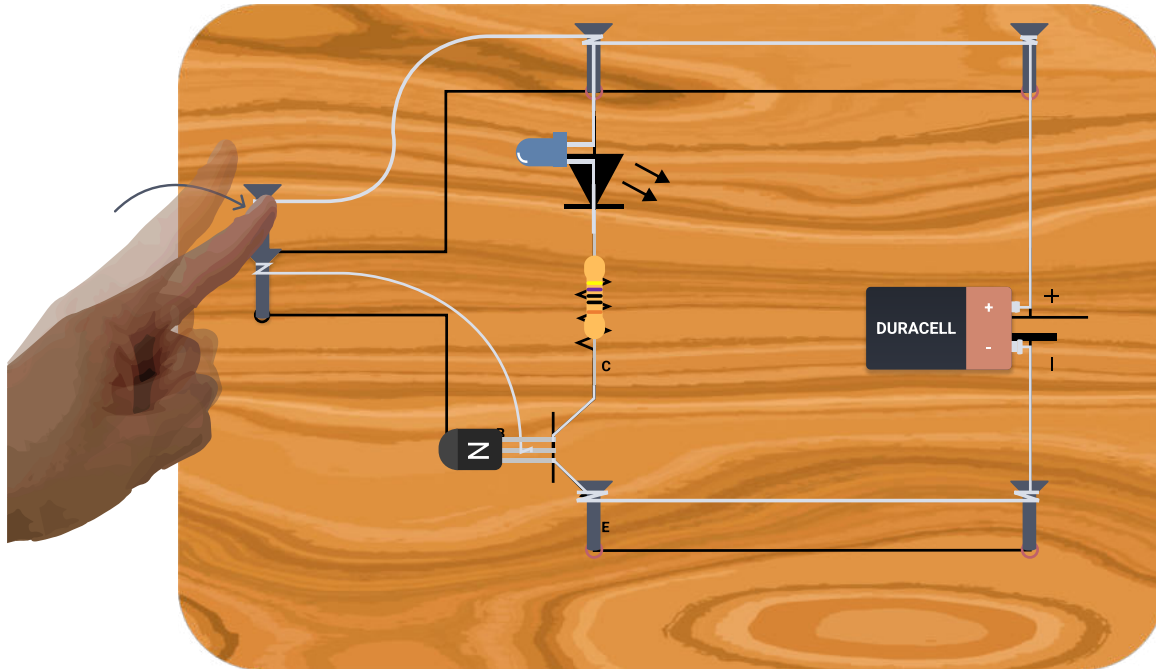
Det er vigtigt at transistorens ben er lodet til de rigtige søm/
modstand

Sæt til sidst batteriet til kredsløbet



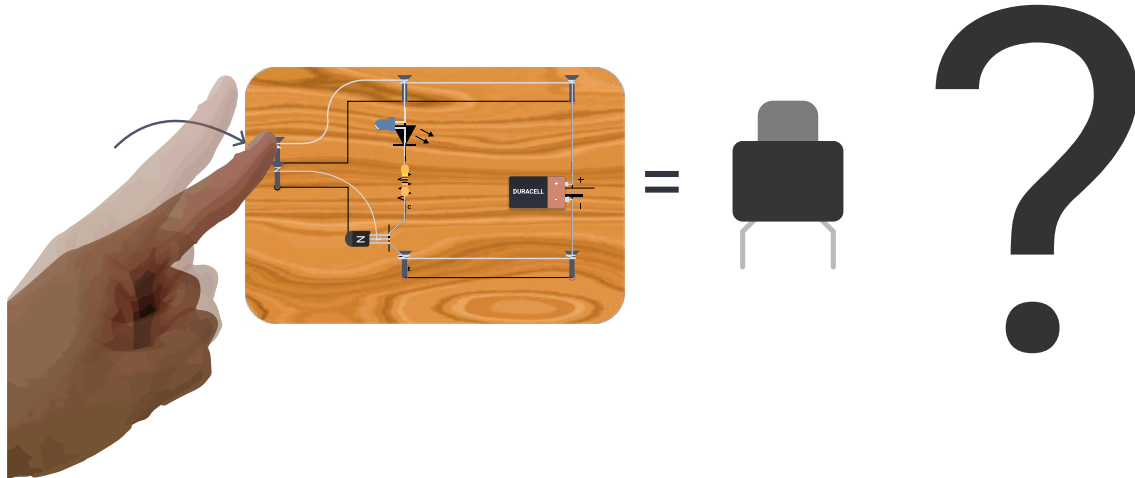
Husk + og - på batteriet!

Tænd og sluk for pæren ved at
ligge en finger over de to ydre
søm



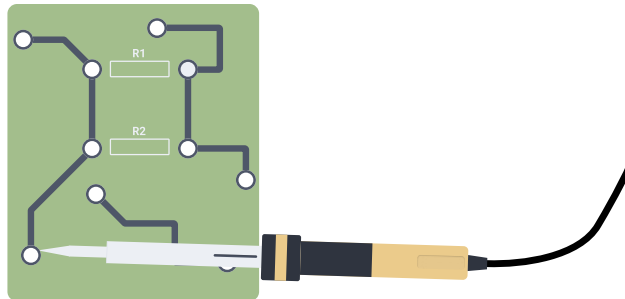
Dette simple kredsløb virker lidt som en kontakt, hvor i får lov til at tænde og slukke for en LED ved afslutte kredsløbet med fingeren.

Mennesker er nogenlunde elektrisk ledende, men prøv med andre materialer for at se om disse også virker!



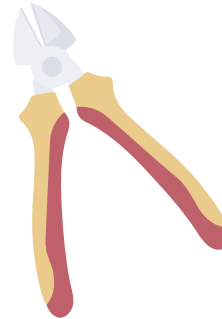
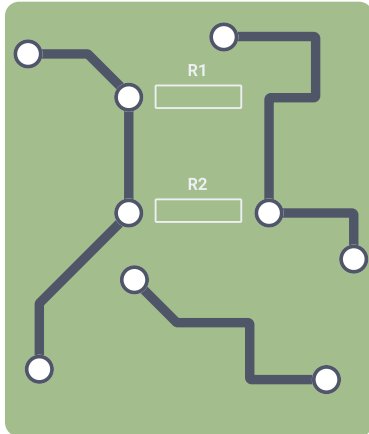
PCB LODNING

Hvordan gør man?



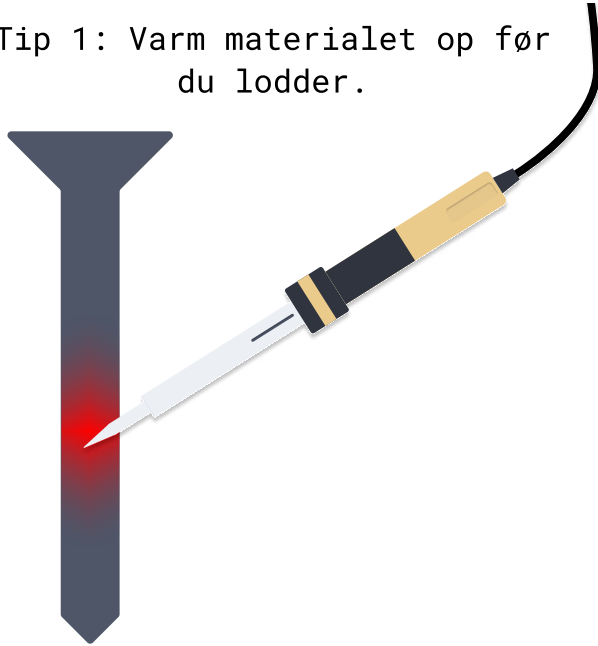
I dette eksempel vil vi gerne lodde en modstand på vores PCB

PCB
Printed Circuit Board
(Printplade)

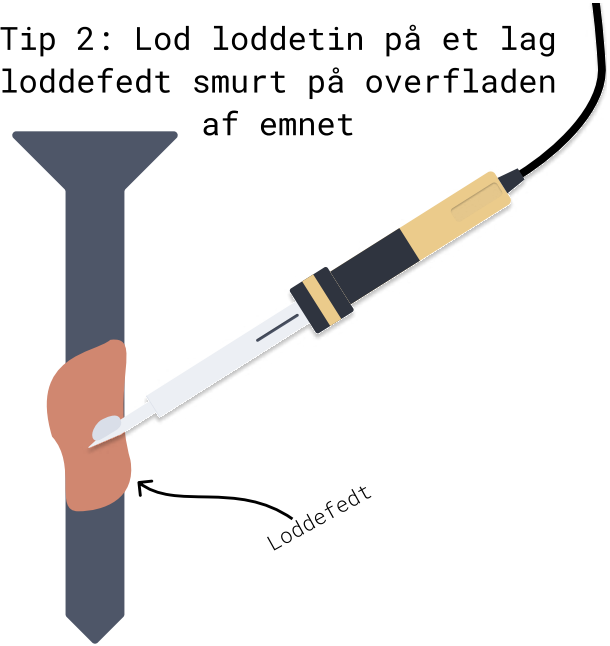


Generelle tips, hvis det er svært at lodde til en overflade:

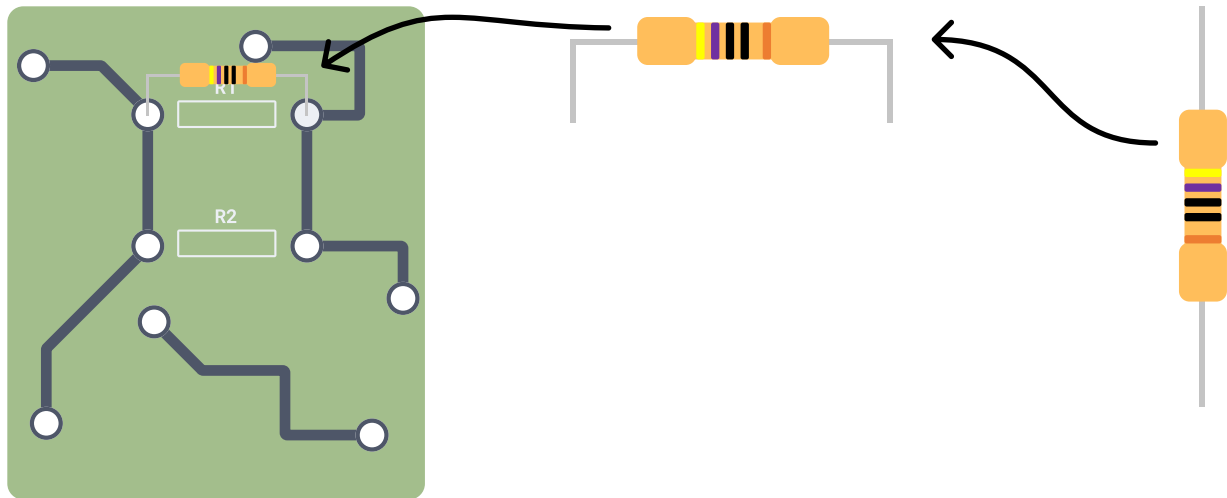
Tip 1: Varm materialet op før du lodder.



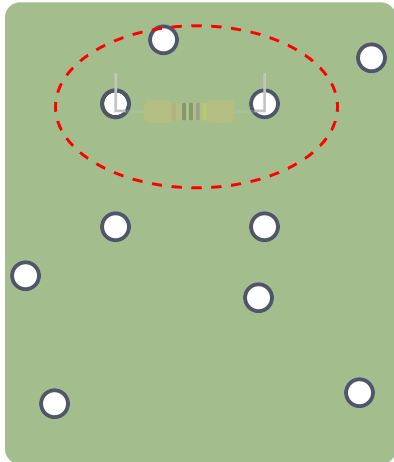
Tip 2: Lod loddetin på et lag loddefedt smurt på overfladen af emnet



Først, bøj enderne på dit
komponent så det passer til
hullerne

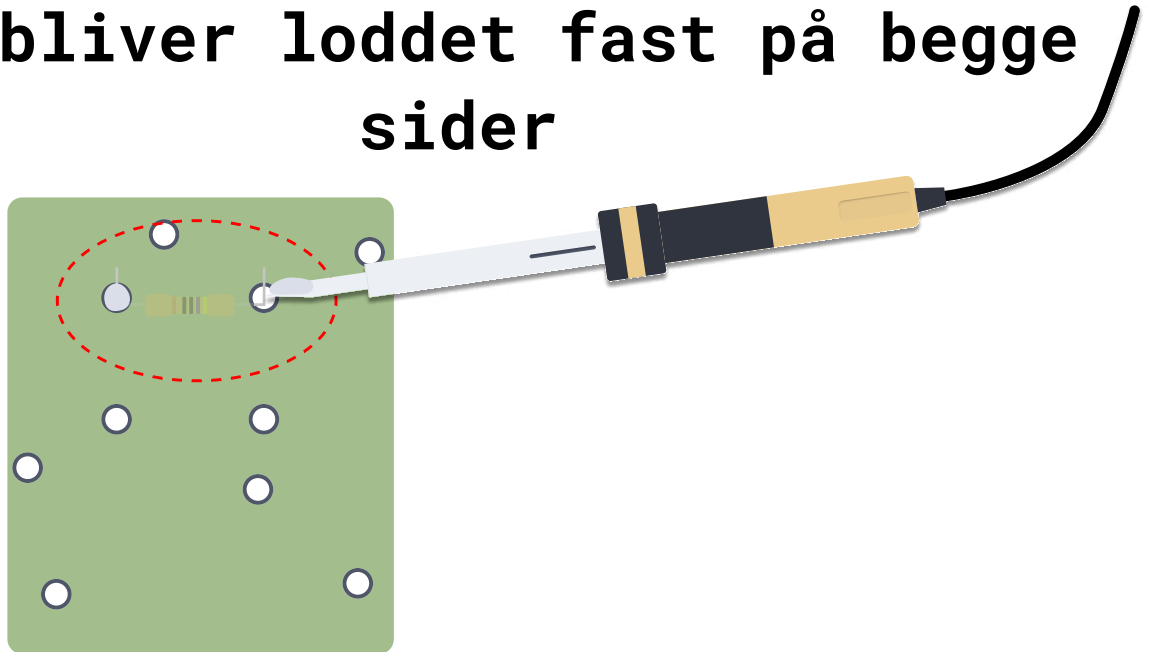


**Vend PCB'en om, således at
komponent benene vender ud mod
dig selv**

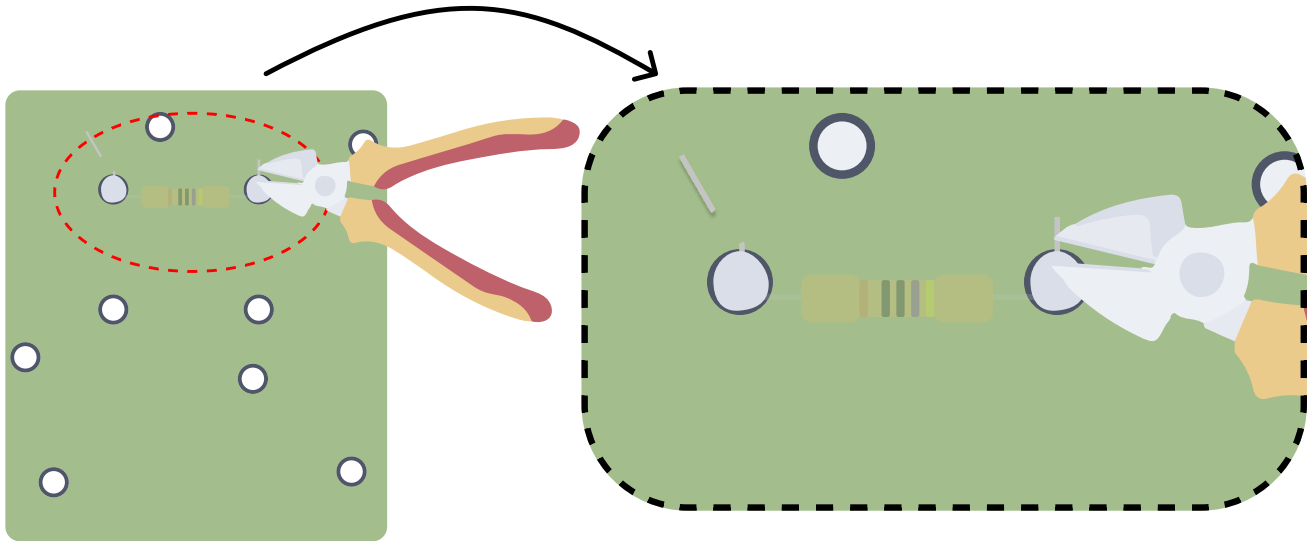


På denne måde er det nemmere
for os at lodde komponentet
fast senere!

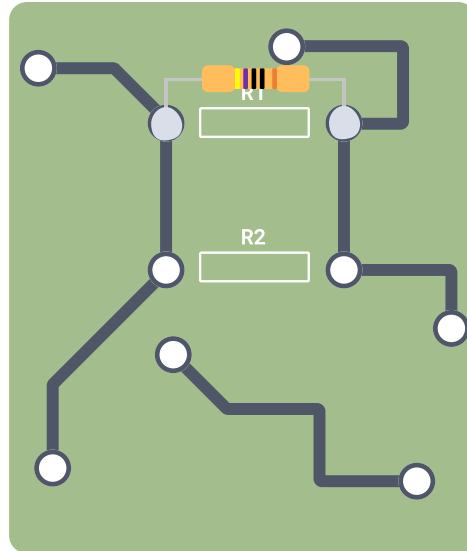
Lod nu komponent fast bagfra! Det er en god idé at lade noget loddetin løbe igennem hullet så den bliver loddet fast på begge sider



Efter komponentet er loddet fast
kan vi klippe endestykkerne af
benene!



Hvis du har gjort et godt loddearbejde, vil komponentet også være loddet fast på forsiden!

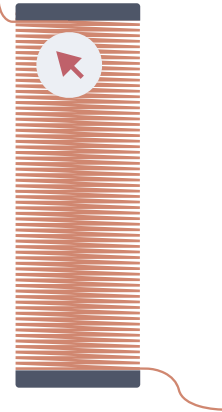


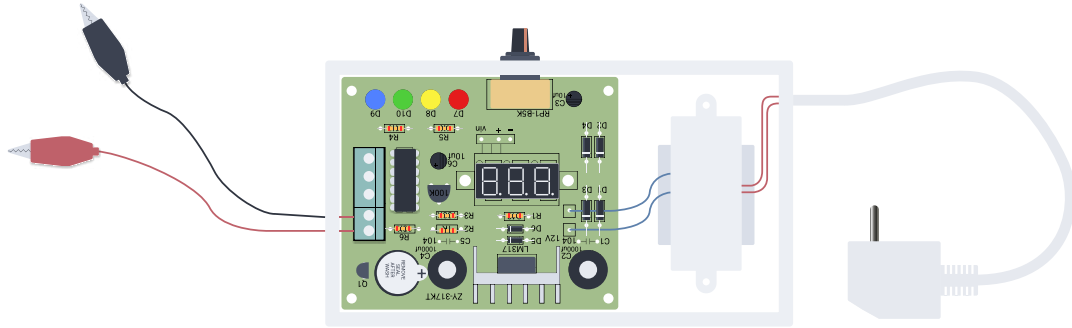
Det er en god ide at starte med de mindste komponenter først og de største til sidst, for at skabe bedre plads på printpladen mens du loder.

Lod som det første



Lod som det sidste





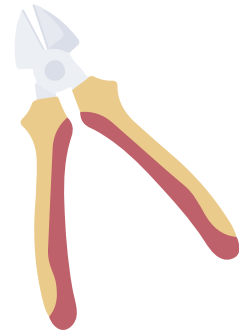
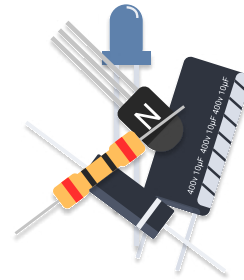
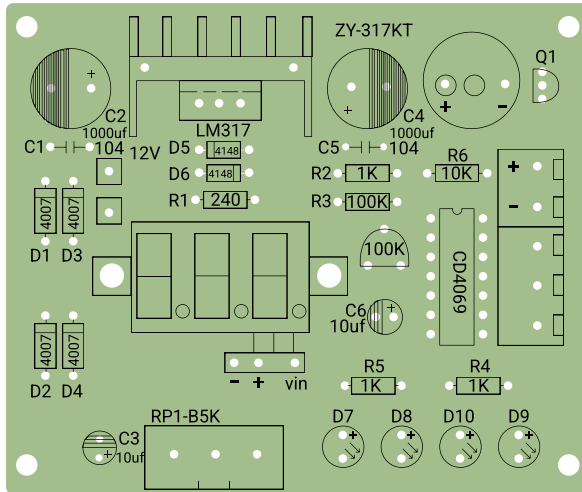
STRØMFORSYNING

Et fanstastisk

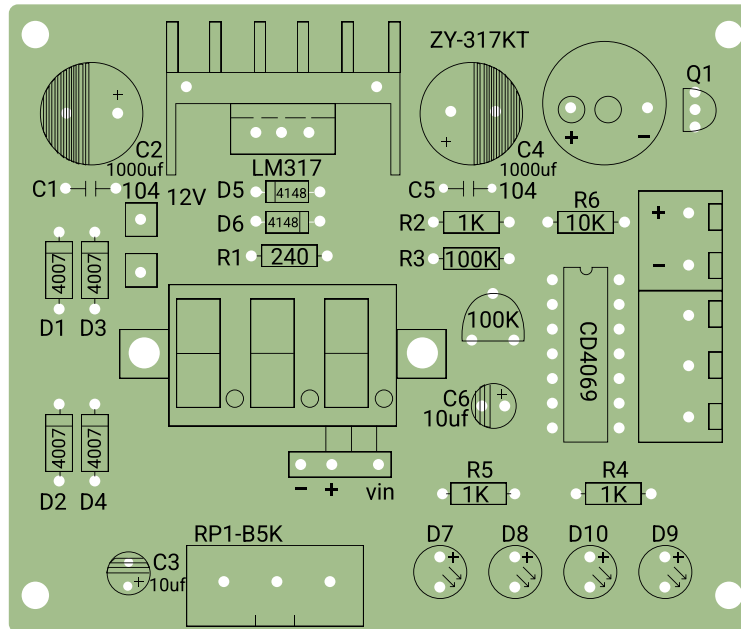
værktøj til

fremtidige projekter

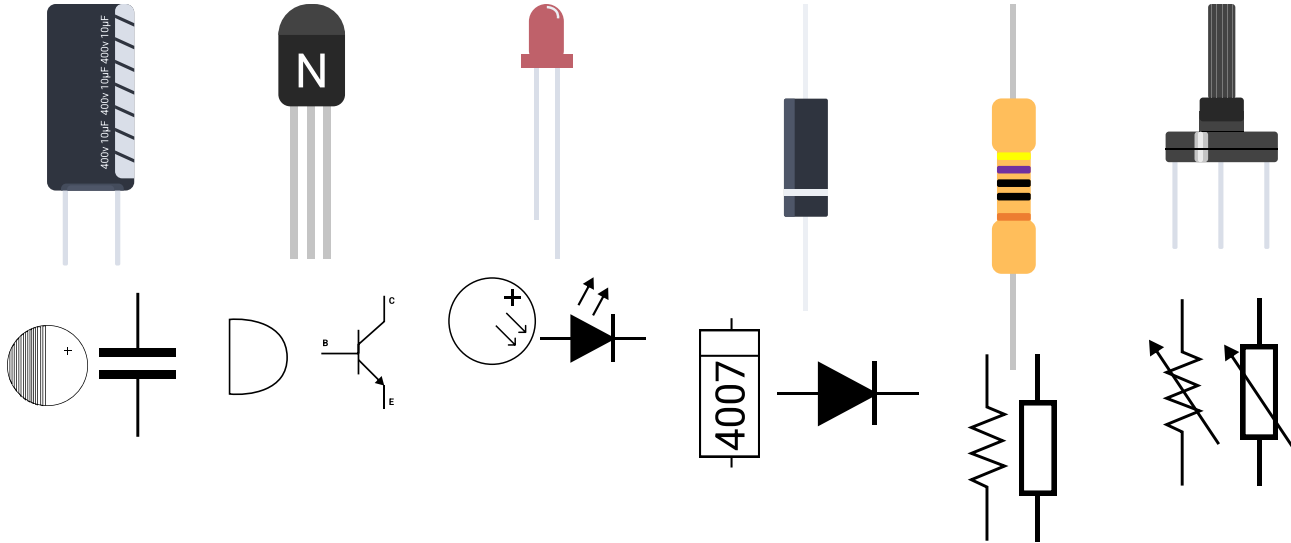
Jeres første store projekt: En Strømforsyning!



Der er meget der foregår på
printpladen, men bare tag tingene
én efter én



De fleste af delene har i allerede arbejdet med før!



Enkelte dele er nye, men dem skal vi nok få forklaret!

Transformer



“Jeg laver farlig vekselstrøm til en svagere vekselstrøm!”

Hex inverter
(Embedded Electronics)



“Jeg styrer hvordan elektronikken skal styres rundt!”

Spændingsregulator

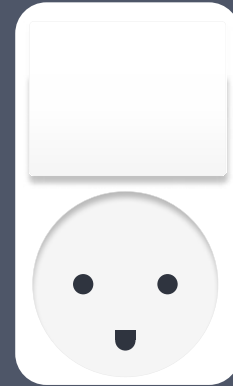


“Jeg kan skrue op og ned for jævnstrømmen!”

Husk: Jævnstrøm og Vekselstrøm

Strøm fra stikkontakten er vekselstrøm. Elektronerne bevæger sig frem og tilbage og når de gør det, bevæger energien sig igennem magnetfelterne.

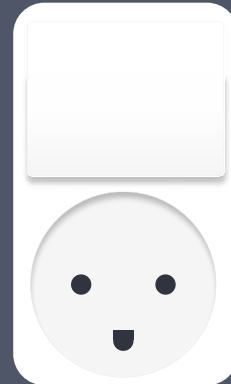
Et batteri skaber jævnstrøm, fordi en spændingsforskel (Volt) trækker elektronerne fra “-” mod “+”. Når de bevæger sig bevæger energien sig i magnetfelterne med lysets hastighed (næsten)



Pas nu på derude!

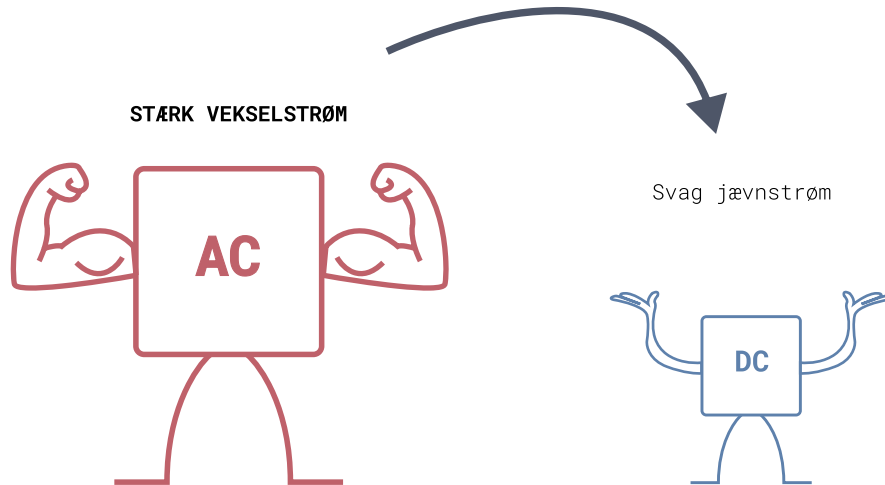
Vekselstrøm kan være farligt, så lad være med at "lege" med stikkontakten før i er færdige.

I skal ikke være urolige, men i skal bare tænke jer om.



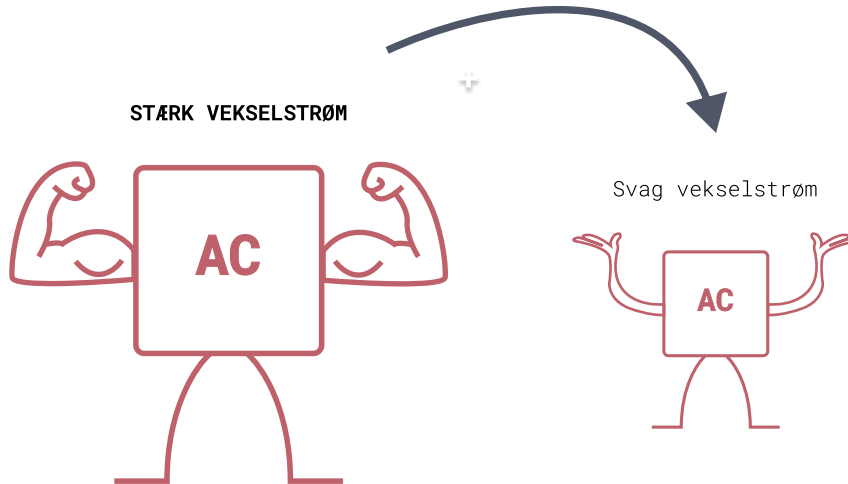
Hvad er det en strømforsyning gør?

En strømforsyning tager den stærke strøm fra væggen og laver det om til noget vi kan bruge i vores computere. Det sker over forskellige trin, og det er forskellige dele, som løser udfordringerne fra de forskellige trin.



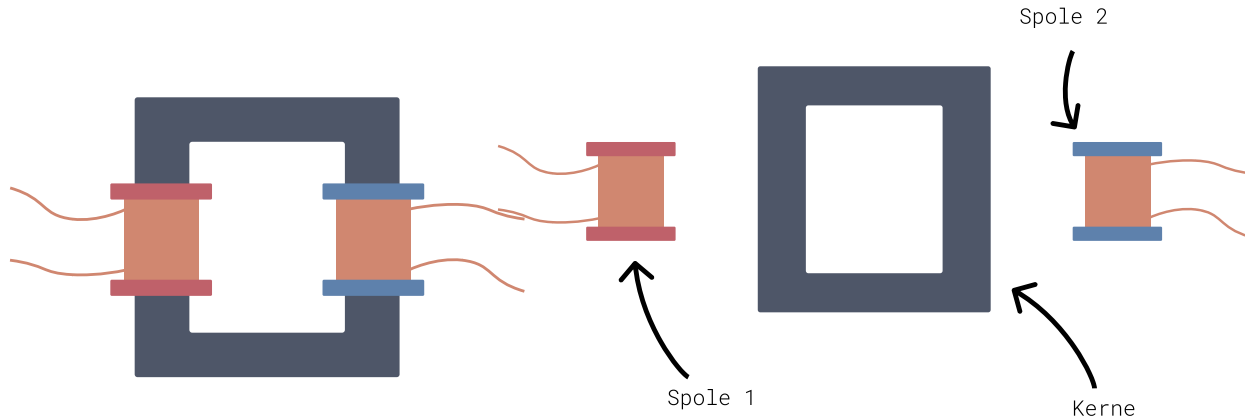
Trin 1: Transformere spændingen

Det første trin der skal løses er at reducere spændingen. Dette er vigtigt, fordi mange af vores dele ikke ville kunne holde til den spænding, der kommer fra stikkontakten. Så for at holde liv i vores teknologi får vi reduceret spændingen.



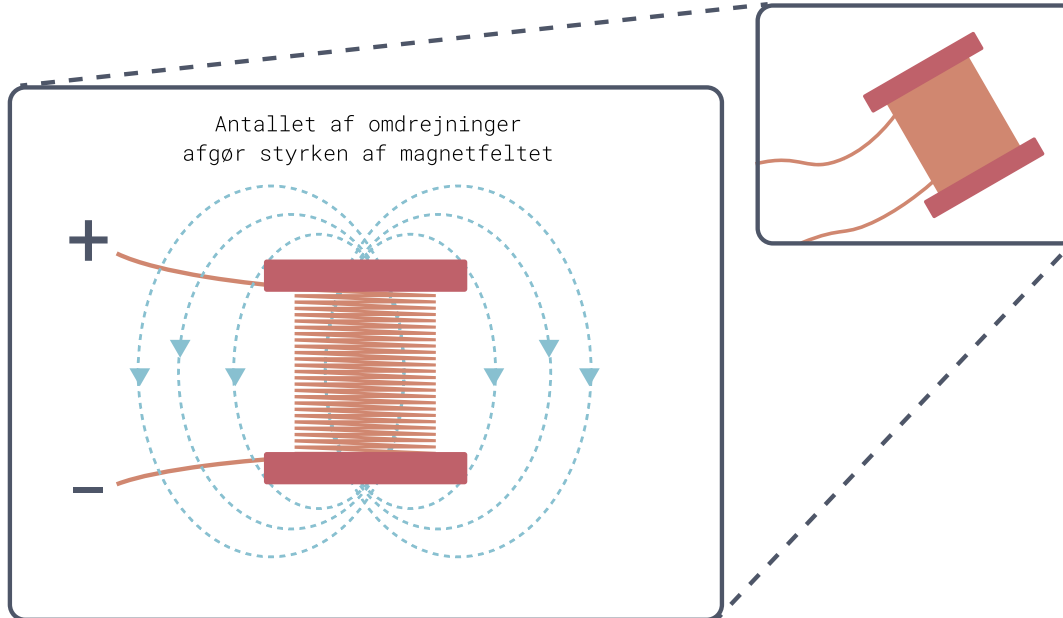
Transformatorer 1

Reduktionen i spændingen bliver skabt igennem en transformator. Som navnet indikerer, transformerer den spændingen. Transformatorens vigtigste dele er de to spoler. Det er dem der afgør hvor meget vi reducerer spændingen. Reduktionen passer med forholdet imellem antallet af omviklinger af kobbertråd i hver spole. Kernen forbedrer udvækslingen af energi.



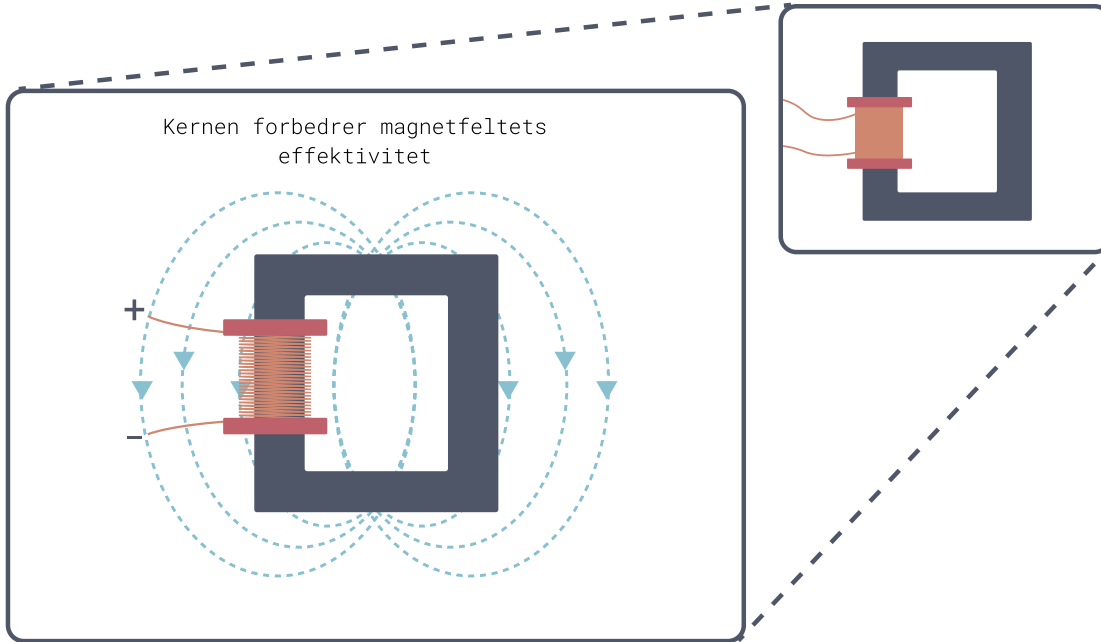
Transformatorer 2

Vi får altså følgende process - hvis vi følger strømmen igennem transformatoren. Først har vi vores input-spole.



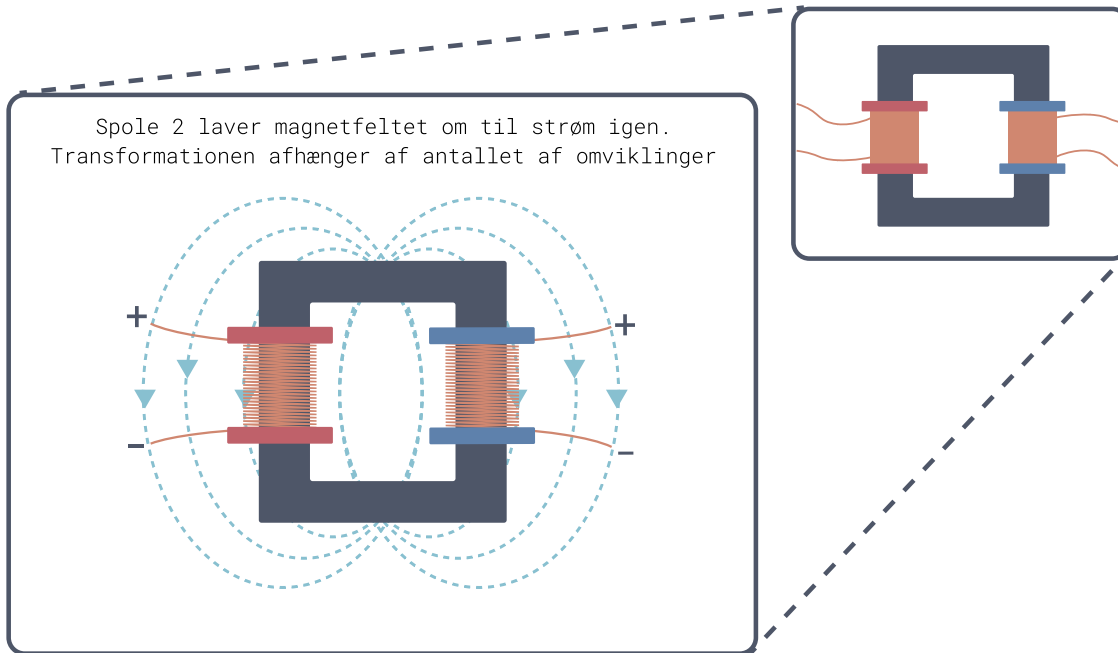
Transformatorer 3

Den sættes på kernen, som hjælper med at videreføre energien fra magnetfeltet.



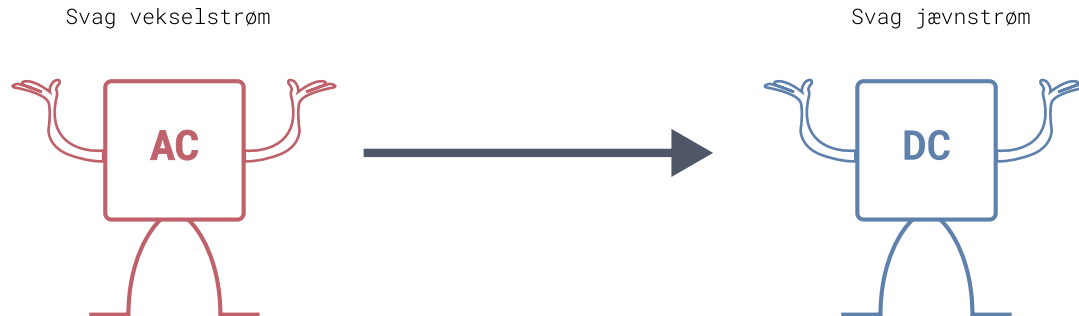
Transformatorer 4

Til sidst kører strømmen ud igennem vores output-spole. Hvis den har færre omviklinger, så bliver spændingen reduceret, hvis flere, så øges spændingen.



Trin 2: Skifte strømmen fra vekselstrøm til jævnstrøm

Nu, hvor spændingen er blevet reduceret, skal vi have lavet strømmen om fra vekselstrøm til jævnstrøm. Som vi har snakket om tidligere er mange elektroniske dele lavet til at virke med jævnstrøm - og derfor er vi nødt til at konvertere det.



Kan i huske dioder?

For at konvertere strømmen skal vi bruge en meget bestemt komponent - nemlig dioden. Strøm kan kun bevæge sig i en retning igennem en diode, og derfor kan vi bruge den til at ensrette strøm.



LED



Diode

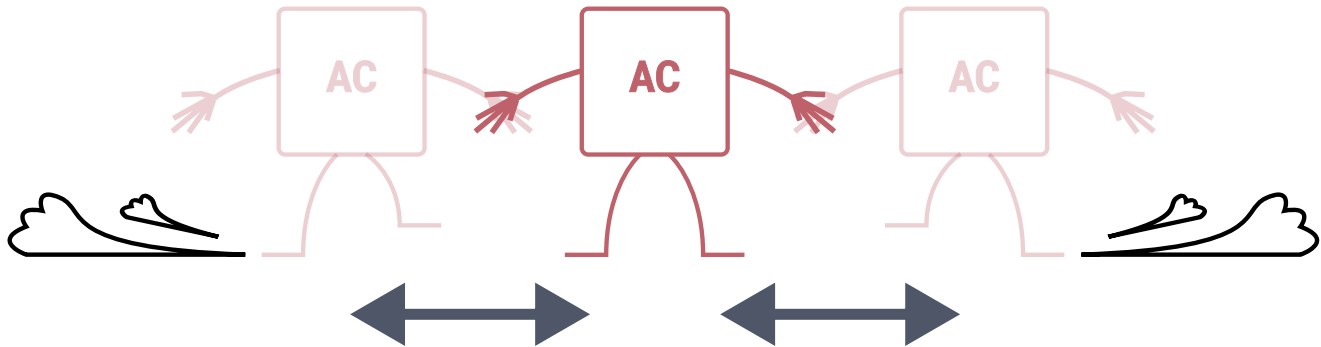


Strømmen kan kun bevæge sig i en retning ligesom vand igennem en ventil for eksempel

Kunsten at ensrette vekselstrøm

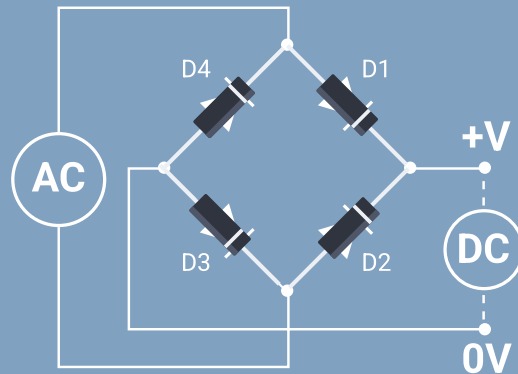
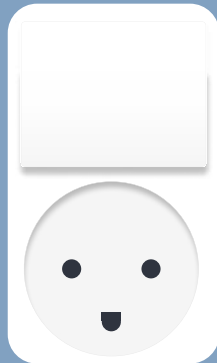
Men det er ikke nok bare at have en enkelt diode. For AC bevæger sig jo frem og tilbage hele tiden, og derfor ville det ikke virke med kun en enkelt. Det er der heldigvis nogen der har arbejdet på at løse!

Vekselstrøm bevæger sig frem og tilbage, ikke i én retning



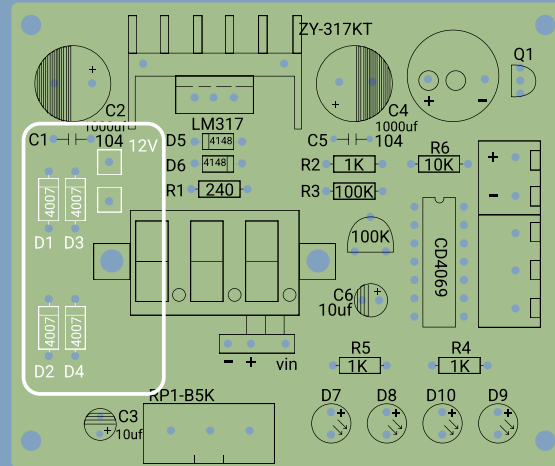
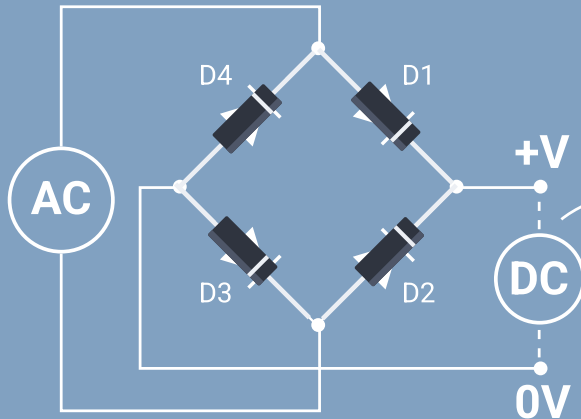
Sådan skifter vi fra AC til DC

Og løsningen er det her fancy kredsløb! Kredsløbet ensretter strømmen ved hjælp af fire dioder.



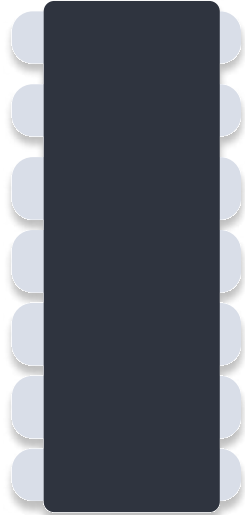
Den er her på jeres PCB

I kan finde ensretter-kredsløbet her på jeres PCB!



Med de andre komponenter i hus, hvad så med disse to:

Hex inverter
(Embedded Electronics)



Spændingsregulator



Spændingsregulatorer

Som navnet siger, kan vi bruge dem til at styre spændingen - ligesom med vores transformator. Men istedet for at få en bestemt værdi ud - kan vi justere hvilken spænding vi vil have.

Det skaber dog ret meget varme, og derfor har vi brug for at kunne køle vores regulator.



Vi bruger en køleplade til at sprede varmen



Spændingsregulatorer

I vores kredsløb bruger vi spændings-regulatoren til at skabe en bestemt spænding mellem 0 og 12 Volt. Det er utrolig smart, når vi gerne vil bruge vores strømforsyning til forskellige kredsløb, som kan tåle forskellige mængder spænding.

Man kan også regulere spænding ved hjælp af noget som heder Pulse Width Modulation, men den tager vi en anden gang.

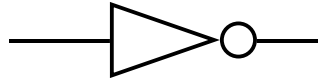


Hex-Inverter?

Betyder 6 på latin

Fordi den laver "+" om til "-"

Der er 6 "Not"-gates i en Hex-Inverter. De her porte bruges til at lave "+" om til "-". Altså, hvis den får 5 Volt ind, spytter den -5V ud.

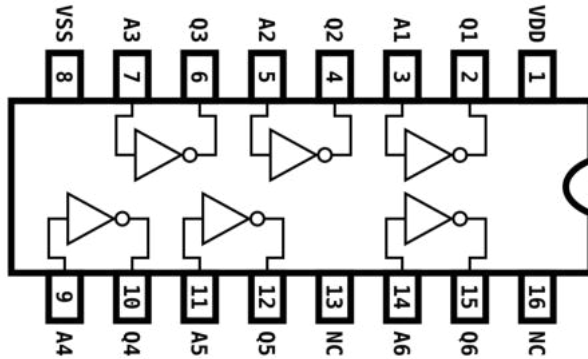
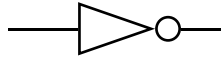


Gates er nogle kredsløb som man bruger til at skabe logik i elektronik. Det kan godt virke lidt abstrakt, og det er helt okay. Det her lærer man (nok) først på HTX eller Universitetet.

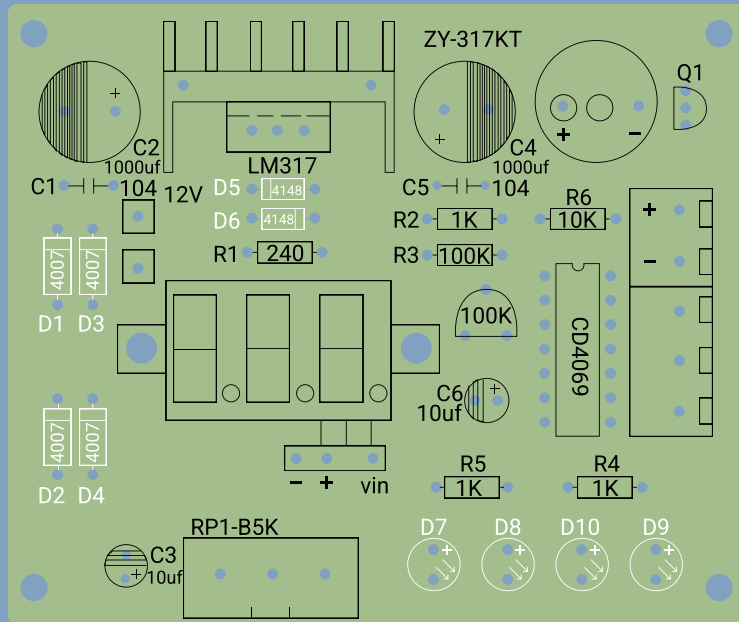
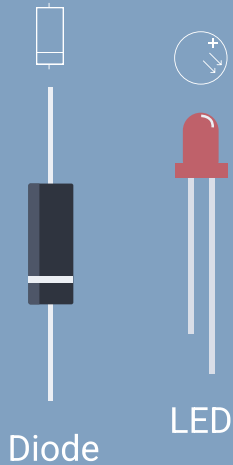


Hex-Inverter?

Man kan bruge en Hex-Inverter til mange forskellige ting: Stabilisere strøm, øge spænding, osv.



(Repetition) Hvor skal i sætte jeres dioder og LEDer?



(Repetition) Hvordan skal i sætte jeres dioder og LEDer?



Diode



Linjerne markere "-" siden

Den runde side hedder Anoden ("+")

Den flade er cathoden ("-")



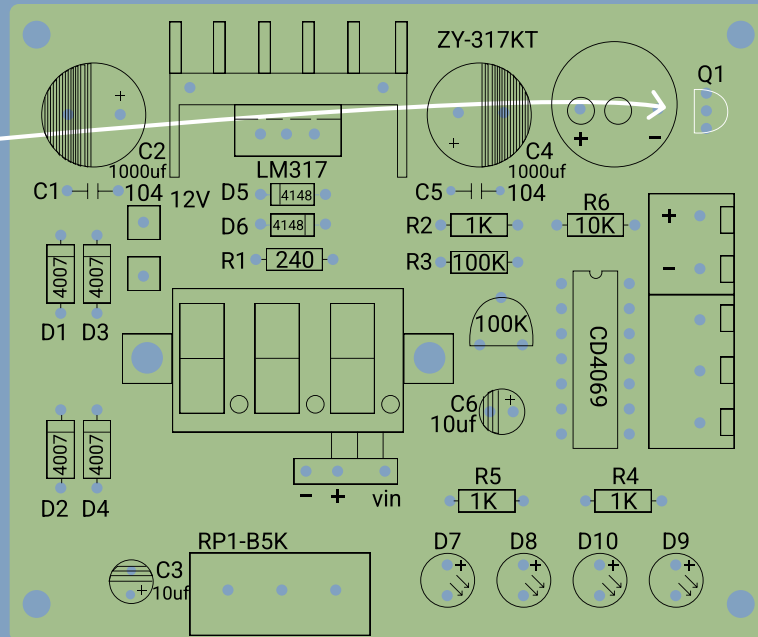
Kig efter det lange ben, for at finde "+"



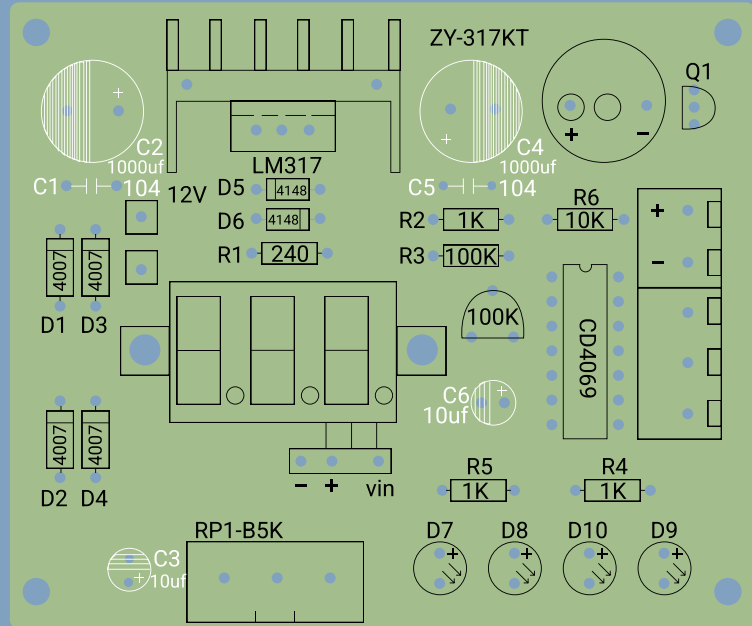
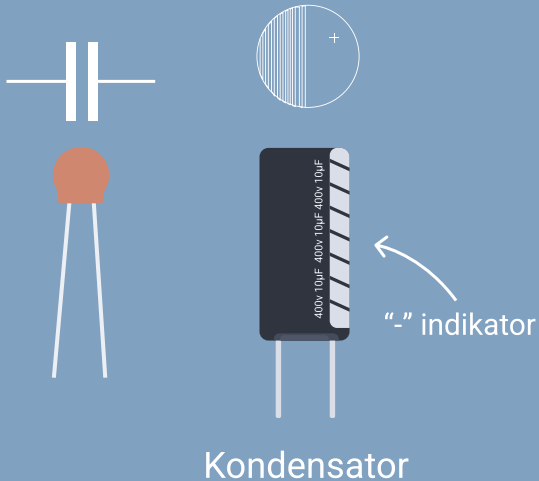
(Repetition) Hvor skal i sætte jeres transistorer?



Transistor



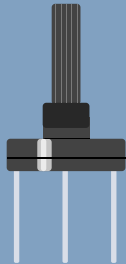
(Repetition) Hvor skal i sætte jeres kondensatorer?



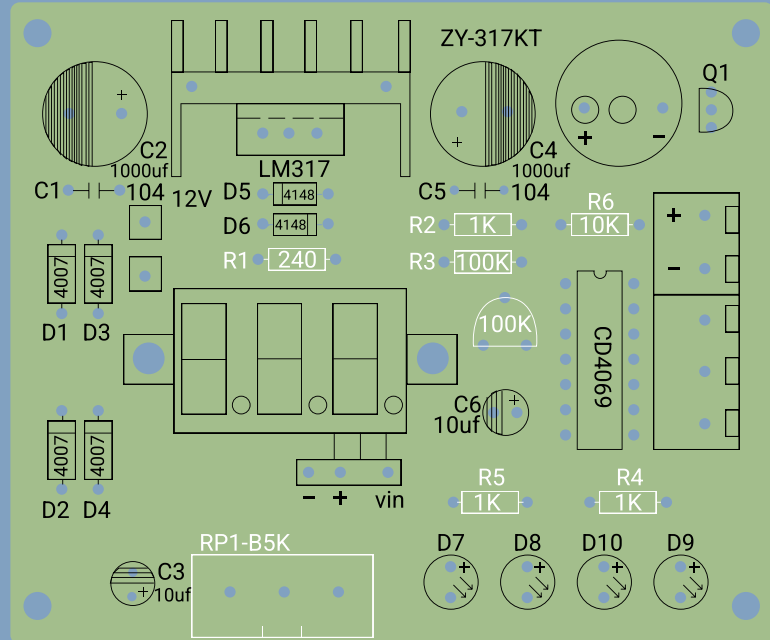
(Repetition) Hvor skal i sætte jeres modstande og potentiometre?



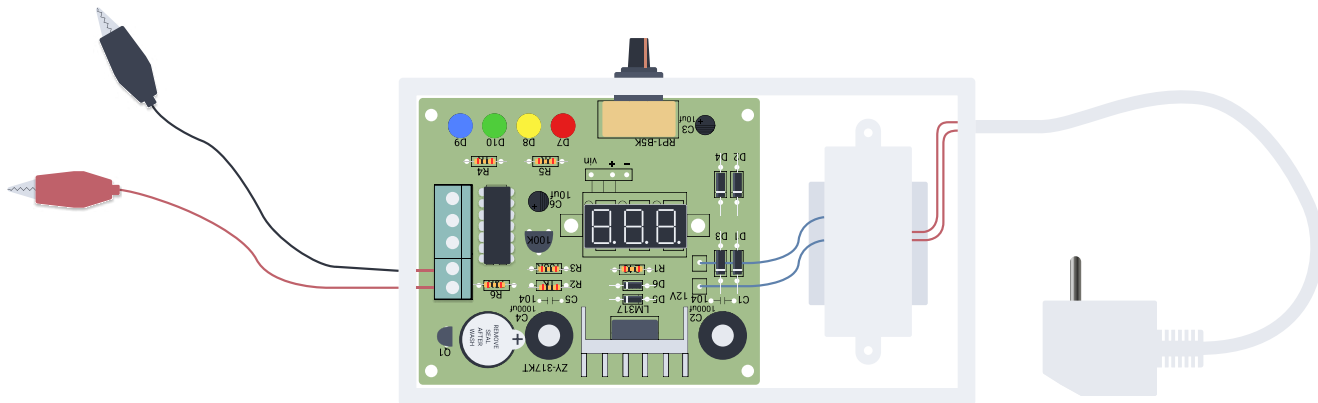
RP1-B5K



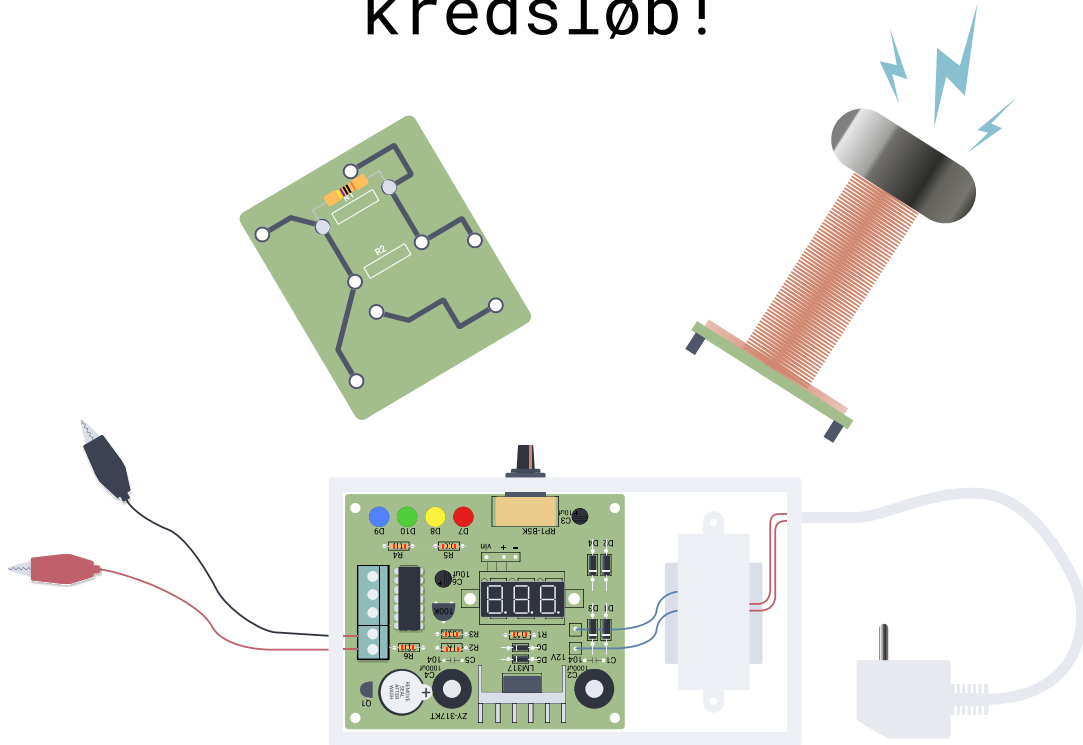
Potentiometer



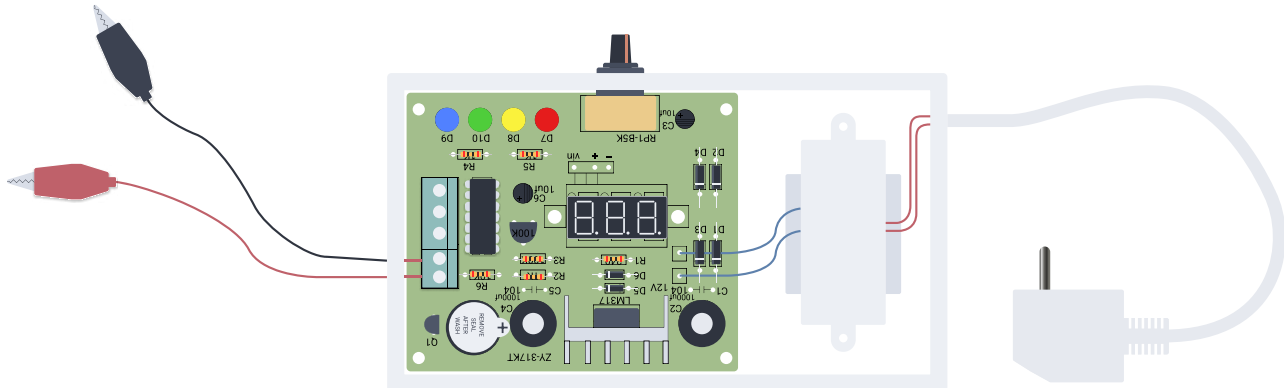
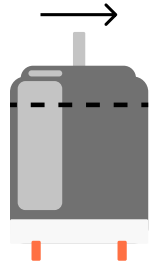
Strømforsyning: Hvad gør vi med den?



Vi kan.... Styre vores fremtidige kredsløb!

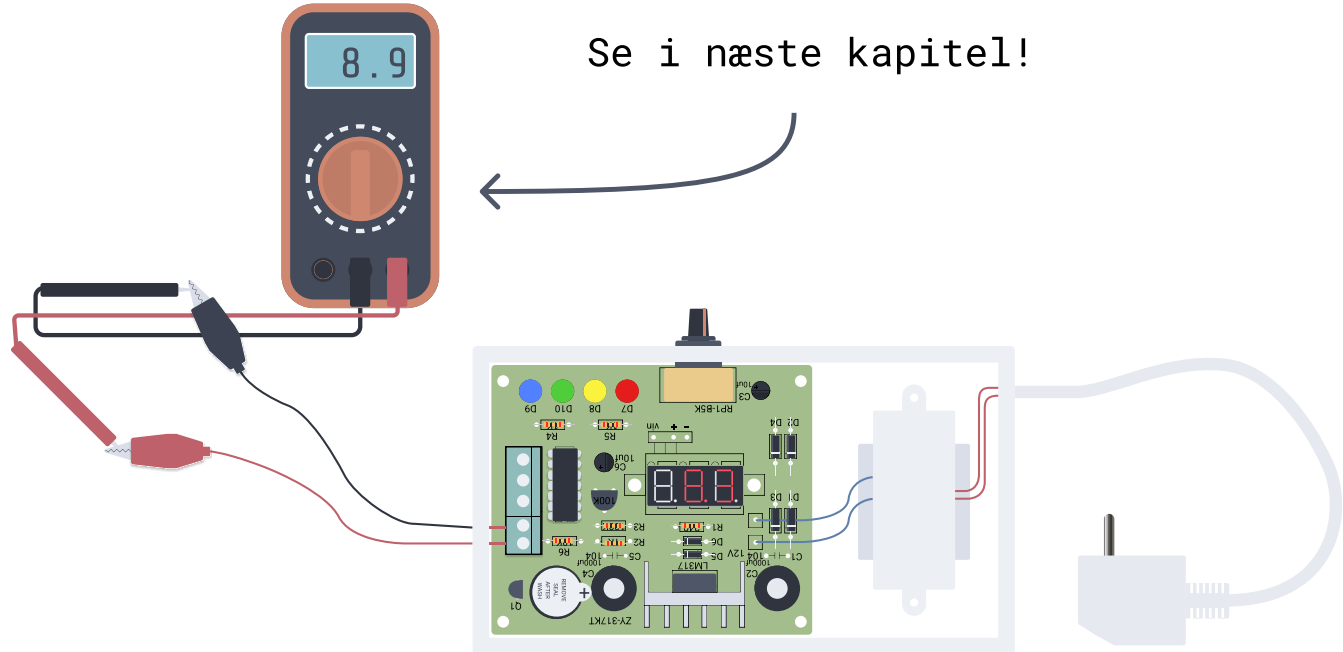


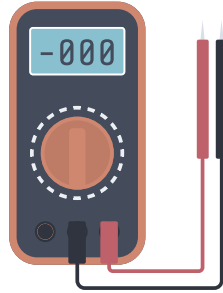
Vi kan... Teste om komponenter virker!



Når i har bygget færdigt...

Så prøv at mål hvor præcis jeres strømforsyning faktisk er..



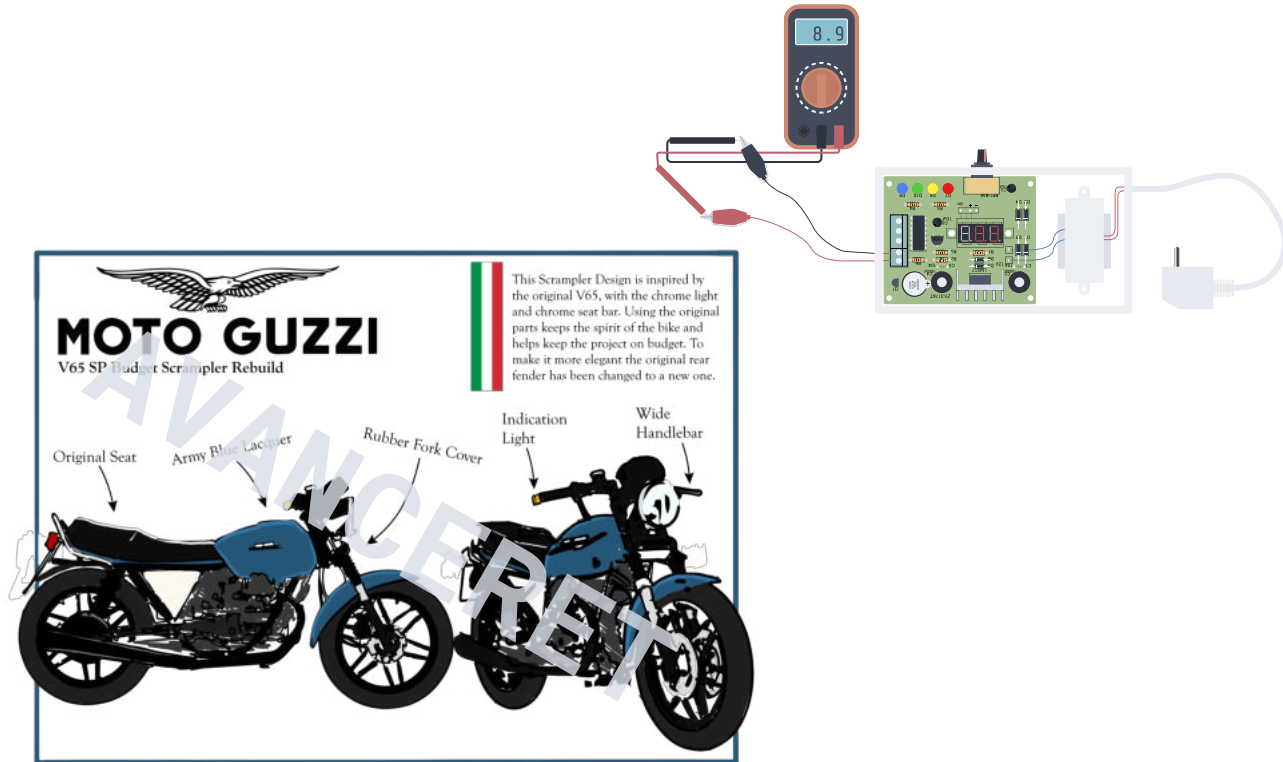


MULTIMETER

De 3 Hurtige

Hvad kan man bruge det til?

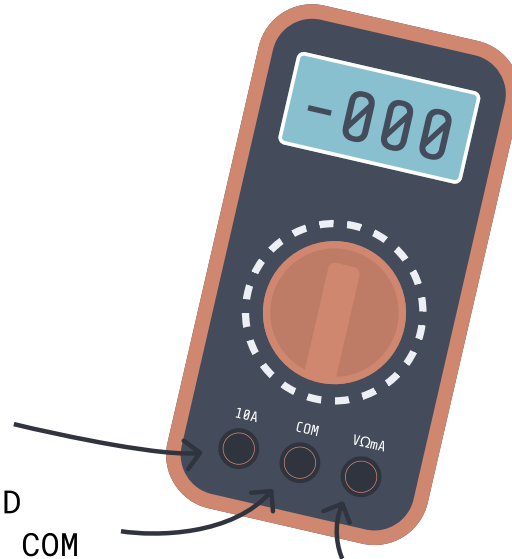
Fikse en gammel motorcykel, eller måle strømmen i en strømforsyning?



Eksempel 0: Hvor skal ledningen sidde?

Hvis i skal måles større mængder
ampere (A)
skal den røde ledning sættes i her

Den sorte ledning skal **ALTID**
sættes i den, hvor der står COM
(Jord)

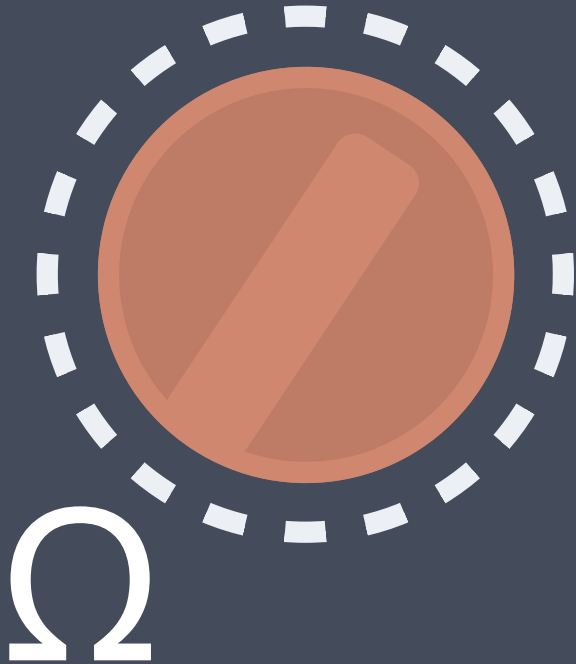


Hvis der skal måles spænding (V),
modstand (Ω) eller små mængder
ampere (A), skal den røde i her

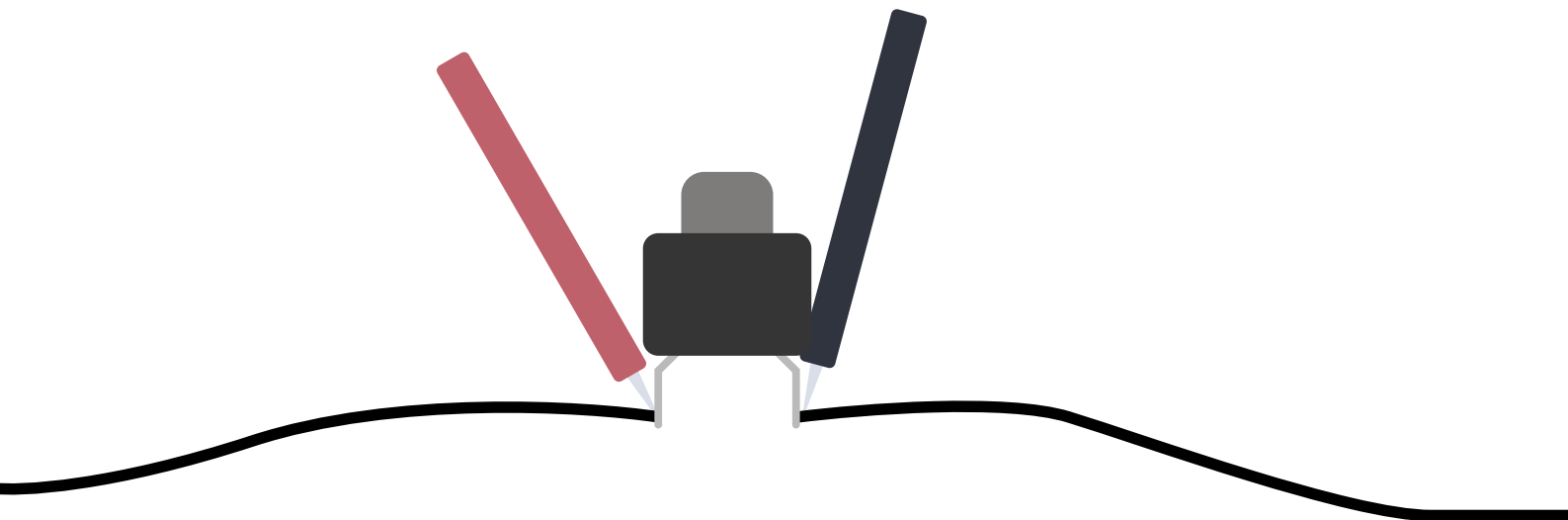
Eksempel 0: Hvor skal ledningen sidde?

Når man arbejder på gamle maskiner, som biler eller motorcykler, skal man ofte forholde sig til gamle knapper. Hvordan kan man tjekke om de virker?



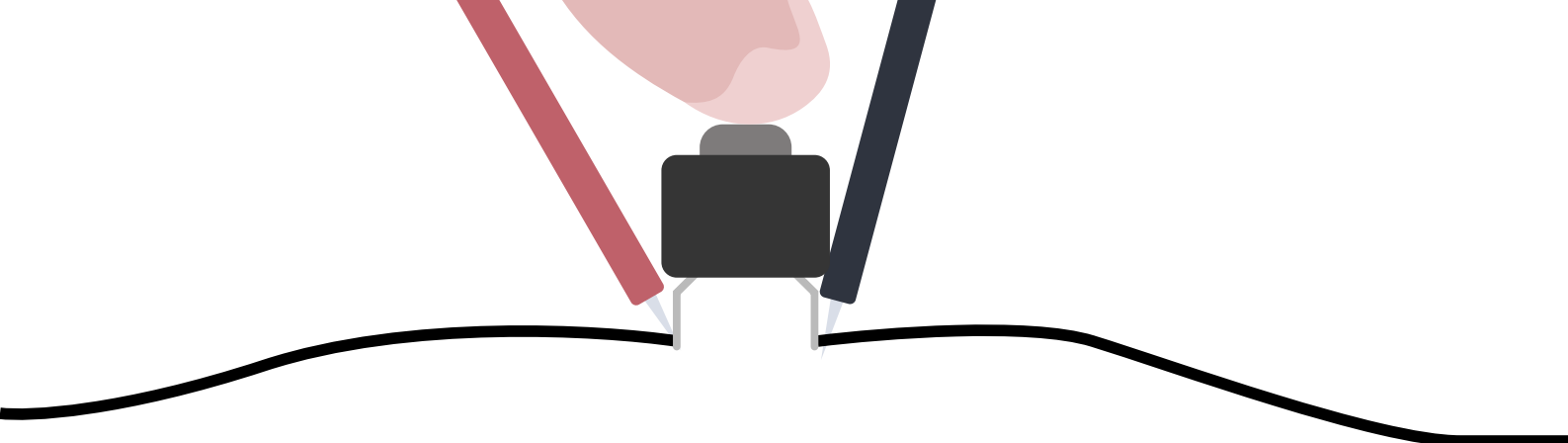


Drej jeres
multimeter over på
modstanssymbolet Ohm
 Ω . Det i tjekker for
er faktisk ikke
modstanden, men
manglen på samme.

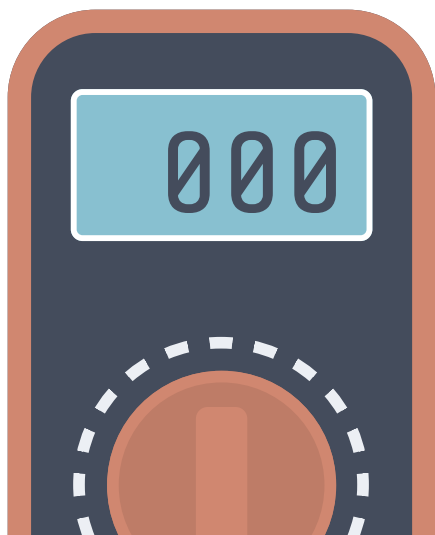


Når knappen ikke er trykket ned, så viser multimeteret en værdi, for eksempel 1.



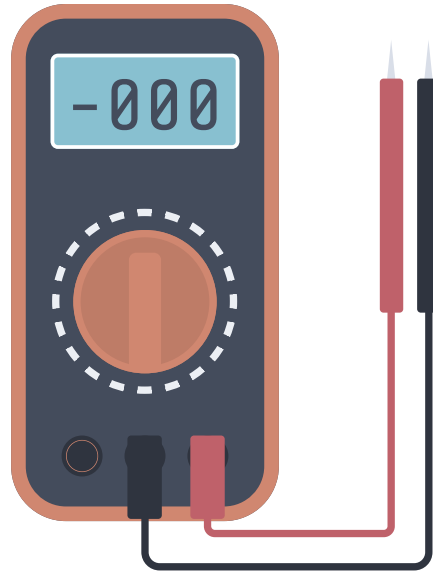


Tryk så på knappen og se, hvordan modstanden falder



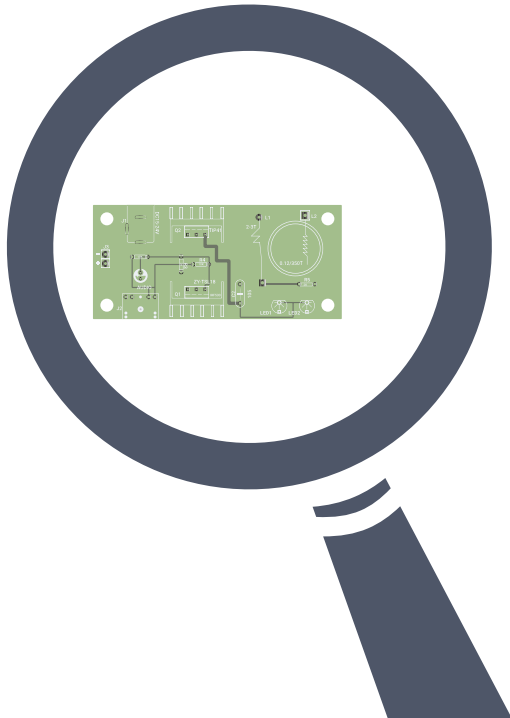
Eksempel 2: Har mit batteri den Rigtige spænding (V)?

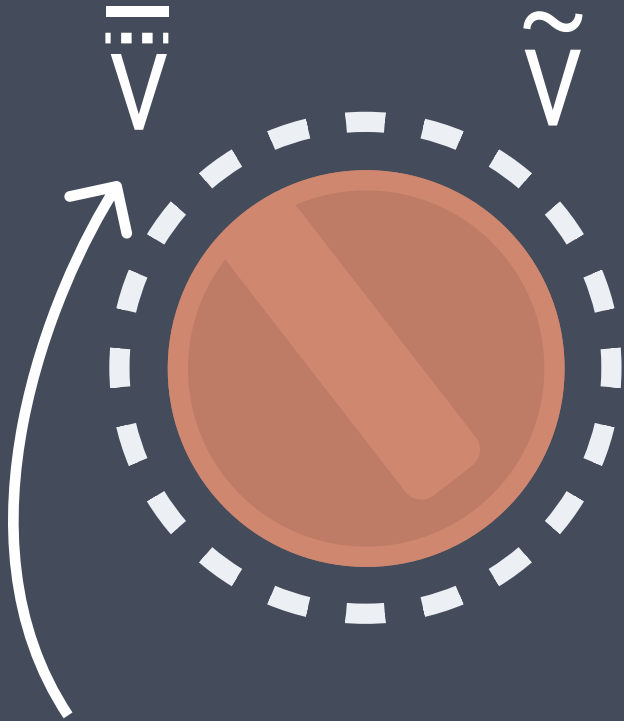
Det er ikke noget værre end at have brugt lang tid på at lodde noget sammen for så at ødelægge det hele med for høj spænding.



“Check Twice, Power Once”

“I min situation var jeg i gang med at lodde en højtalermikrochip. Den er meget lille og det tog lang tid. Da jeg satte strøm til begyndte den at ryge, og virkede ikke... Det var en ommer”
- Anton (der kom til at give den for meget spænding).



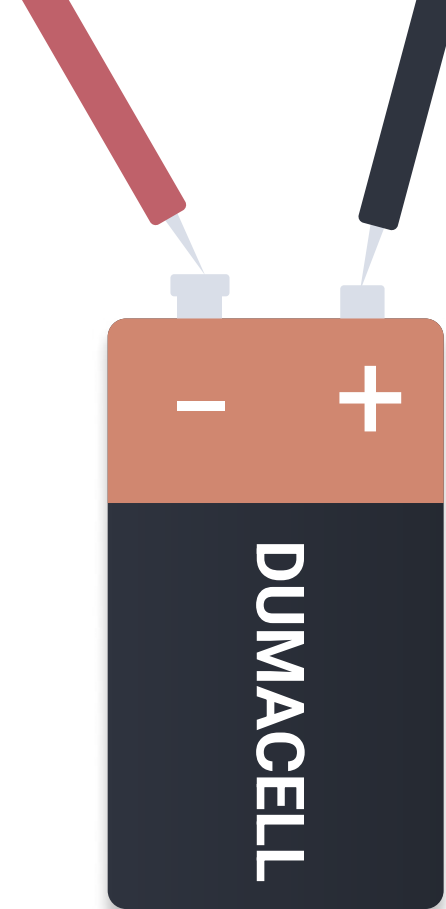


Det her symbol

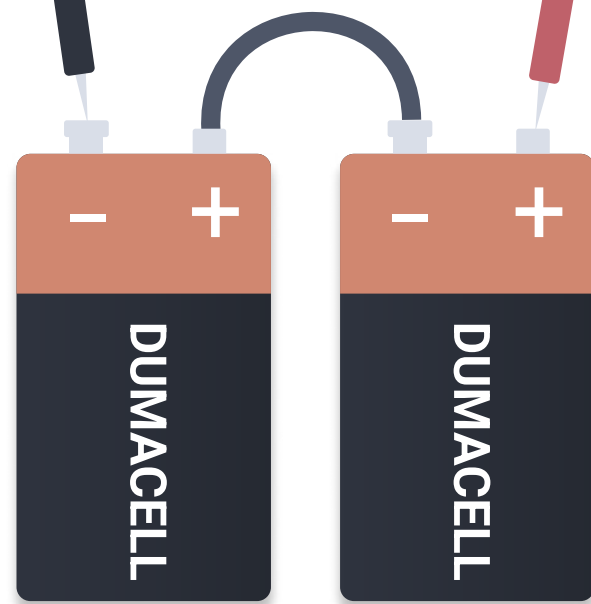
For at undgå at vores hårde arbejde går op i røg, skal vi tjekke spændingen på vores kredsløb. Et batteri udleder jævnstrøm, og derfor skal vi bruge Volt symbolet V med en hel og stiplet streg

Mål på de to poler på batteriet
og se en spænding.

Hvis den er negativ, så prøv at
vende batteriet.



Specielt hvis man bruger flere batterier, så er det virkelig en god ide at tjekke!

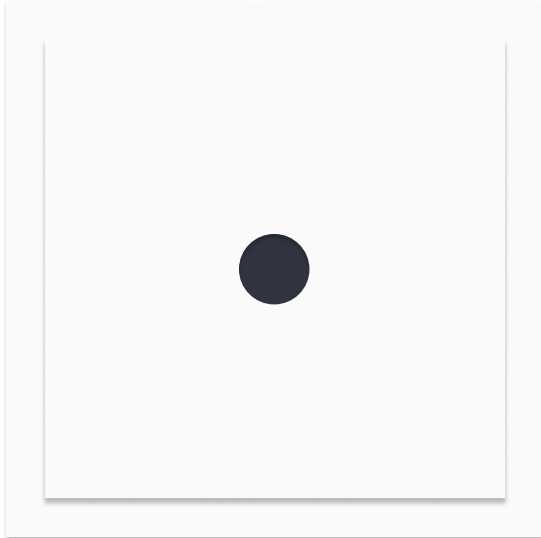


Hvis man sætter to batterier sammen med deres '+' og '-' poler, så får man deres sammenlagte spænding ud. Dette kaldes en serie forbindelse. Hvis man i stedet tilkobler '-' med '-', og '+' med '+', så skaber man en parallel forbindelse der sammenlægger batteriernes levetid. På denne måde kan man både lave parallelforbindelser og serieforbindelser tilpasset til ens projekt.

Eksempel 3: Er der strøm i stikket?

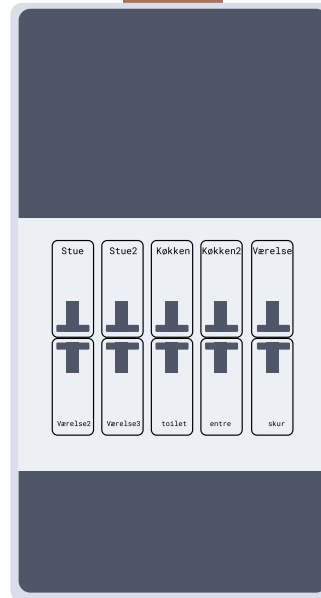


Hvis i er i gang med at renoverer et hus eller hænge en lampe op, så er det her nok vigtigt at vide, hvordan man gør!



Lampeudtaget er et af de steder,
de fleste hobby-eletrikere kommer
til at møde på et tidspunkt.

Sikringskab



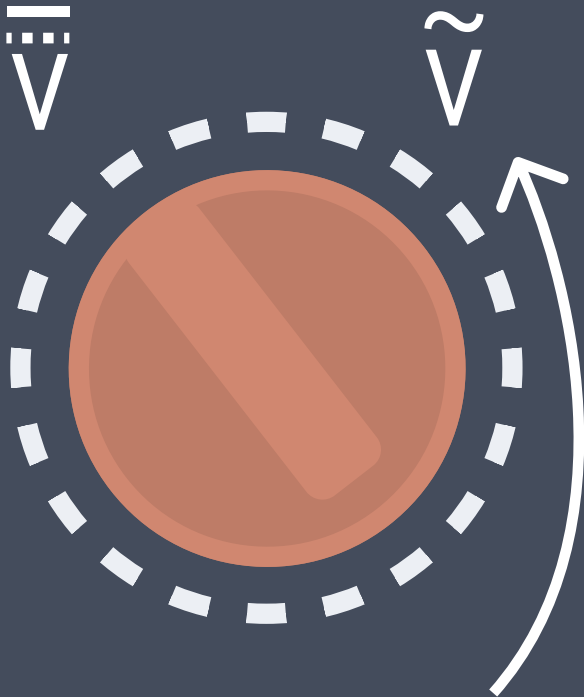
Først skal den relevante sikring slås fra! Eller i hvert fald den du tror det er. Nogen gange er de ikke helt logiske! Det er derfor vi tjekker efter.



Inde i lampeudtaget finder man flere forskellige udtag. M1 og M2 betyder tændledning. Dem regner vi som "+". N og jord \perp symbolet regner vi som "-". L kan også være en fastfaseforbindelse

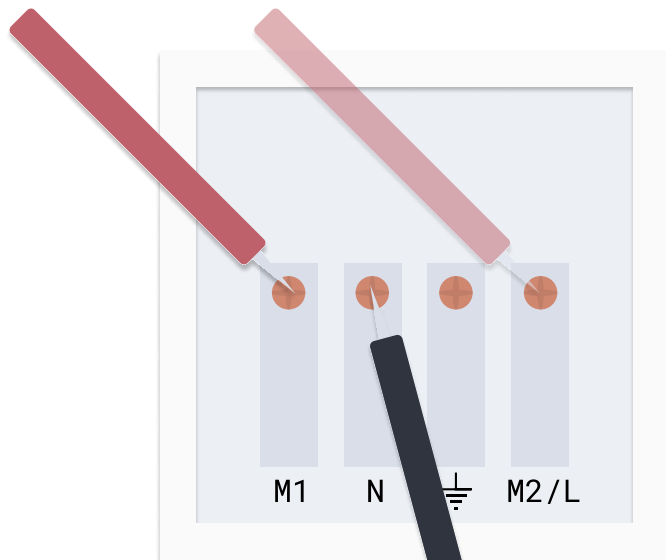


Når vi måler er det M1 og M2 og N, som er de vigtigste. Vi skal tjekke om der er spænding, for spændingen i stik kontakter er nemlig farlig!

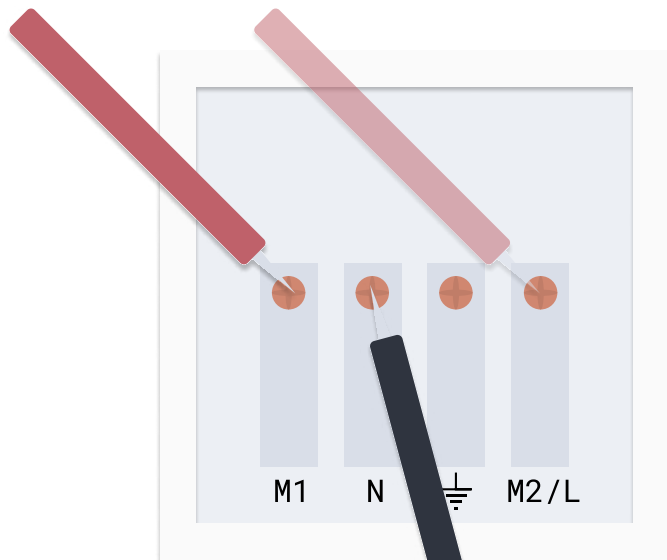


Det her symbol

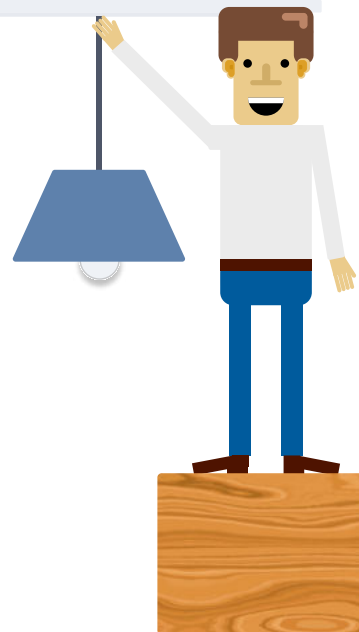
For at sikre at vi ikke lige pludselig får strøm sætter vi vores multimeter over på V med den lille bølge ~ (tilde) over sig.

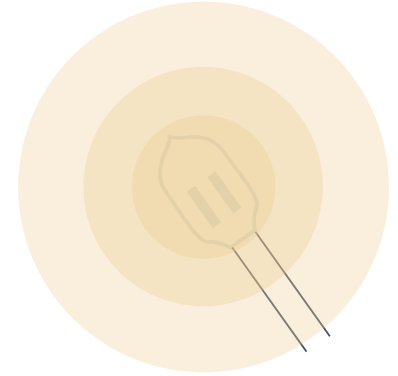


Sæt den sorte på N og tjek derefter M1 og M2. Der må ikke være spænding nogen af stederne!!! Hvis der er, så prøv at slukke nogle flere sikringer.



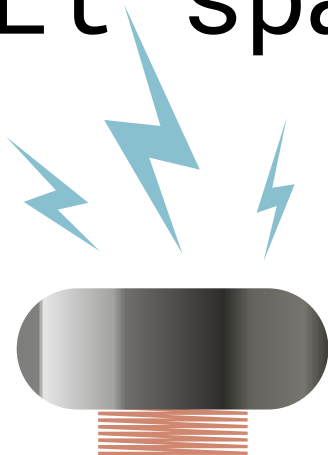
Hvis der ikke er noget
spænding, så kan man
sikkert sætte sin lampe op!





TESLA SPOLEN

Et spændingsfyldt
show!



Remember this face!

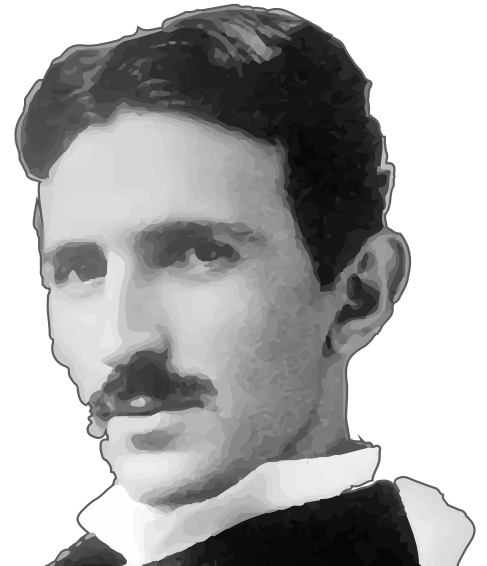
Nikola Tesla 1856-1943



Nikola Tesla

Nikola Tesla er uden tvivl en af de største fysikere nogensinde. Han har opfundet induktionsmotoren og været ansvarlig for udbredelsen af vækselfølgende strøm. Derudover opfandt han teknologien som vi bruger til radiobiler. Han opfandt selvfølgelig også Tesla-spolen, og en masse andet.

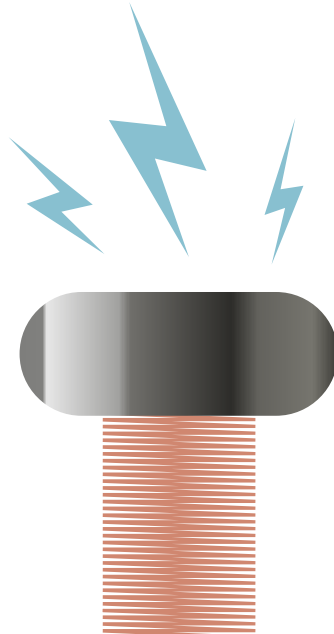
Our boy Tesla was real Smart!



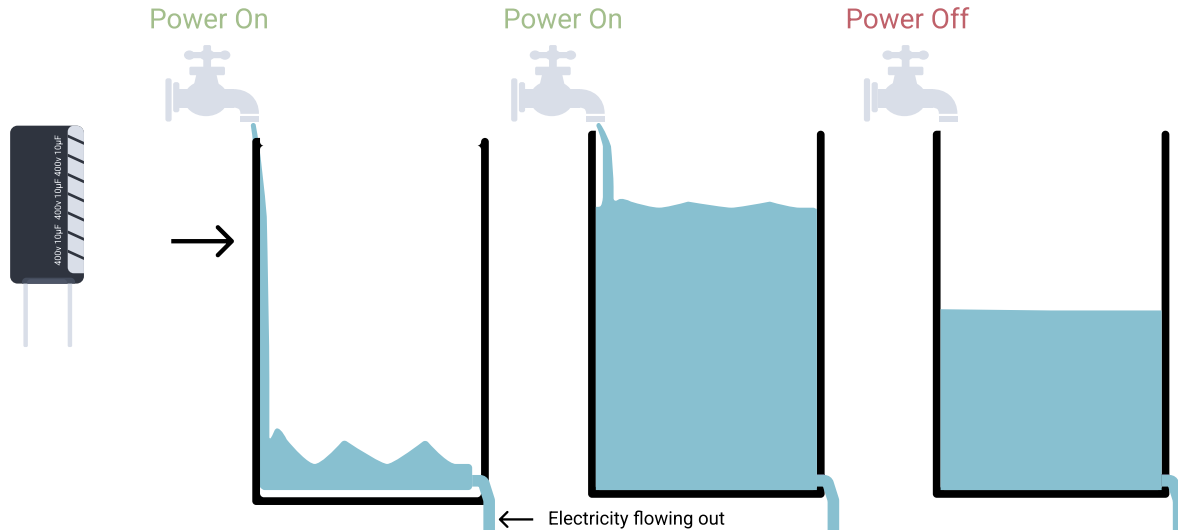
Tesla Spole (Tesla Coil)

Tesla Spolen er faktisk et koncept til at vise, at energi kan bevæge sig igennem luft. Tesla håbede på at udbrede luftbåren energi til hele verden, hvilket vi i dag ved ikke er fysisk muligt (med den teknologi i hvert fald).

Teslaspolen er dog stadig en stor opfindelse, men hvad består den af?



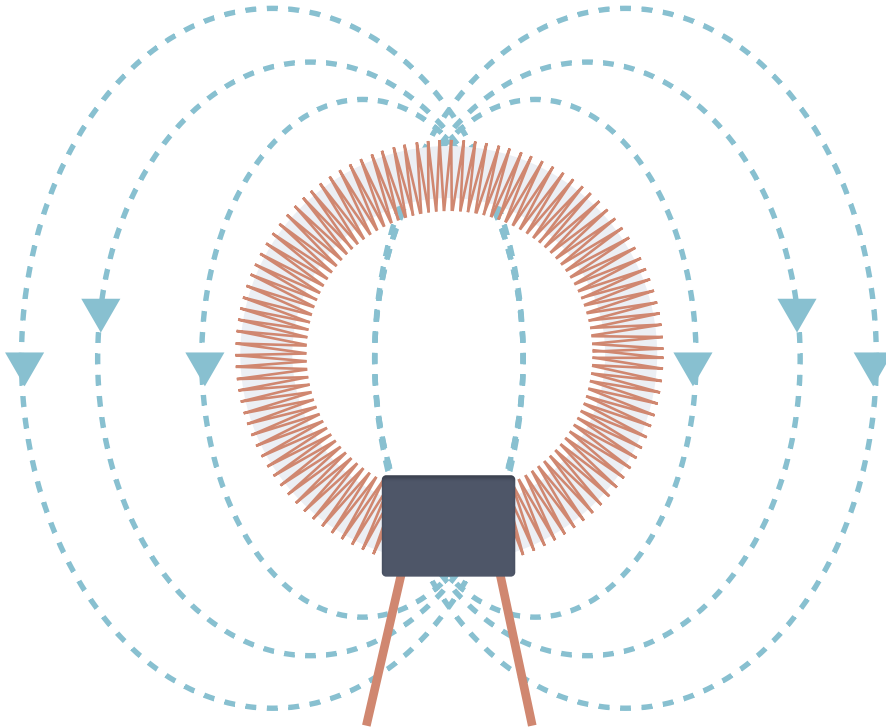
Kondensator



Som i måske kan hukse, så virker kondensatoren som en form for "åbent batteri", hvor ved slukning af strømmen, så vil der stadigvæk løbe noget elektricitet ud af kondensatoren. Dette er næsten det samme princip en "Induktor" fungerer med.

Induktor

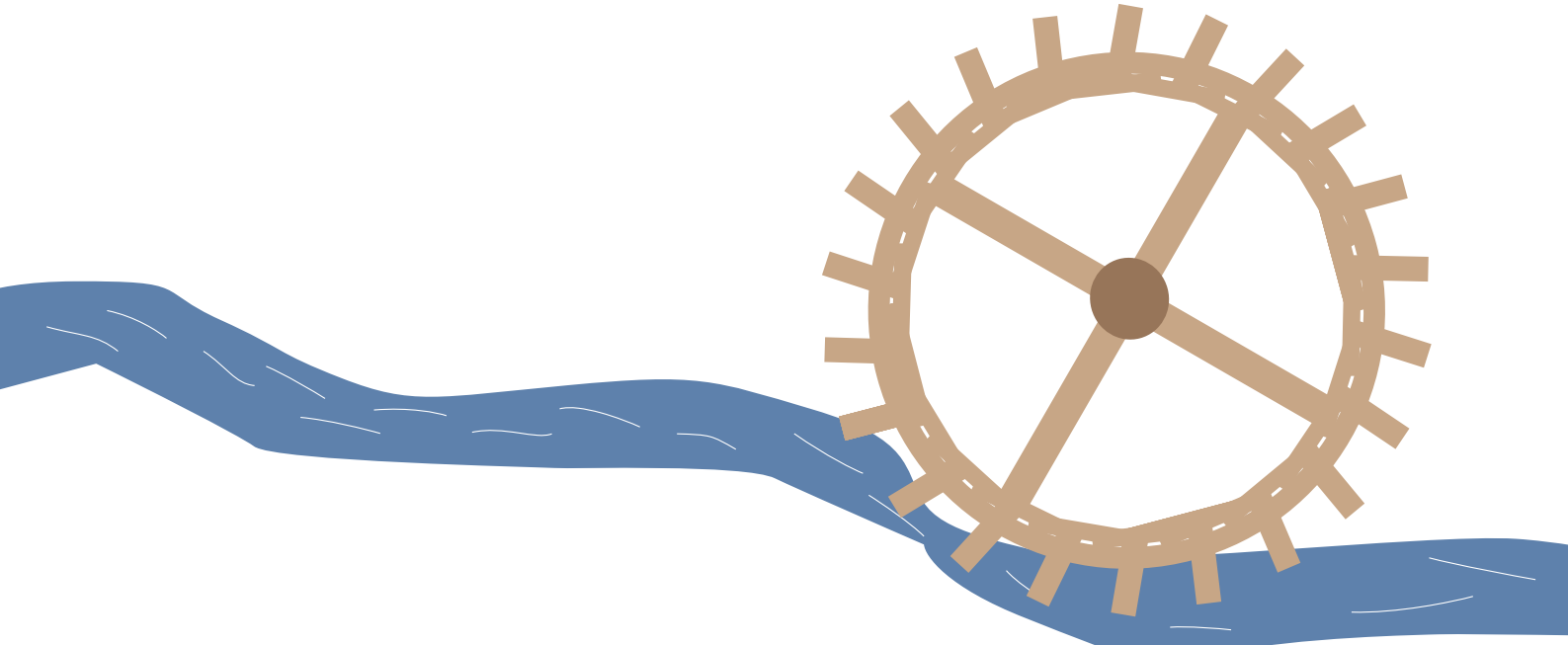
Induktoren er lavet til at holde energi i et magnetfelt - lidt ligesom kondensatoren.



Induktor

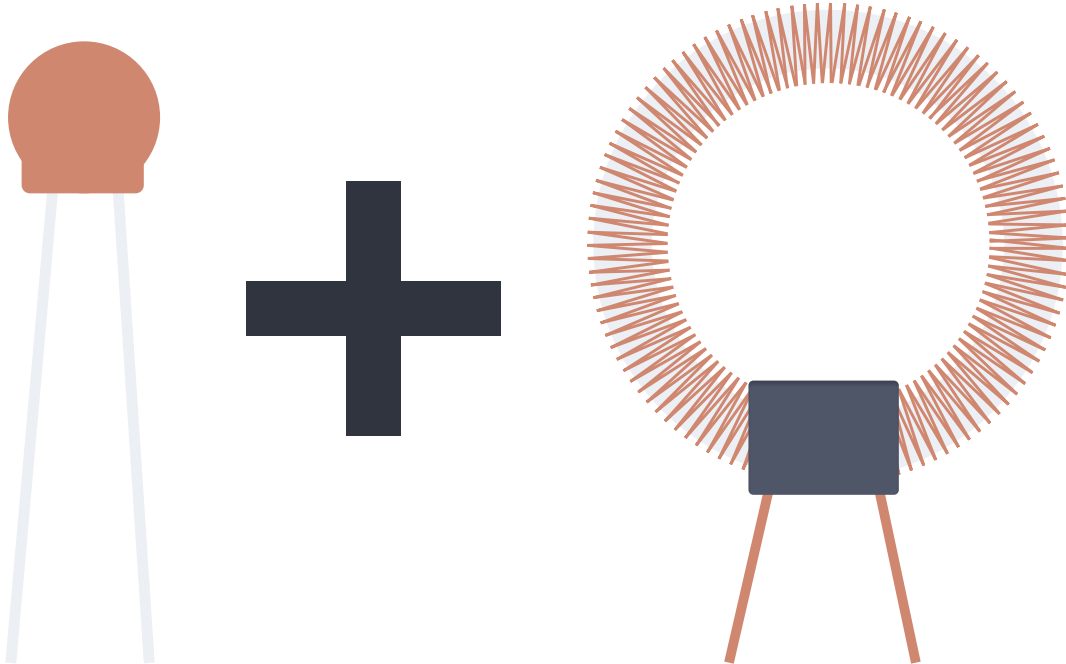
Tag et kig på mølehjulet til en vandmøle for eksempel. Selvom vandet stopper med at bevæge sig, vil hjulet blive ved med at køre rundt lidt endnu.

Det er sådan en induktor fungerer, men istedet for et hjul der drejer rundt, så er det et magnetfelt, som aftager over tid



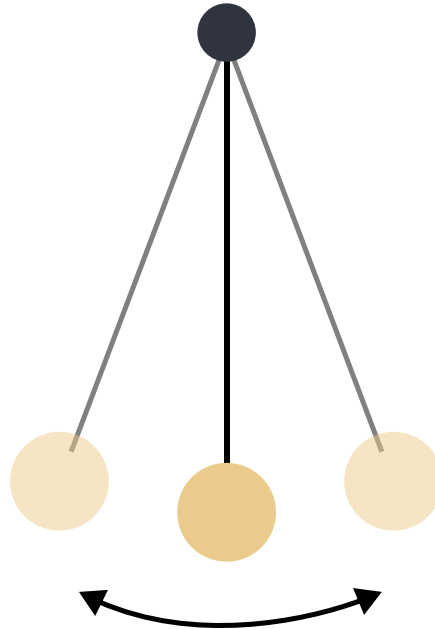
Svingningskredsløb

Men induktoren på plads kan vi begynde at snakke om det svingningskredsløb som er inde i en Teslaspole.



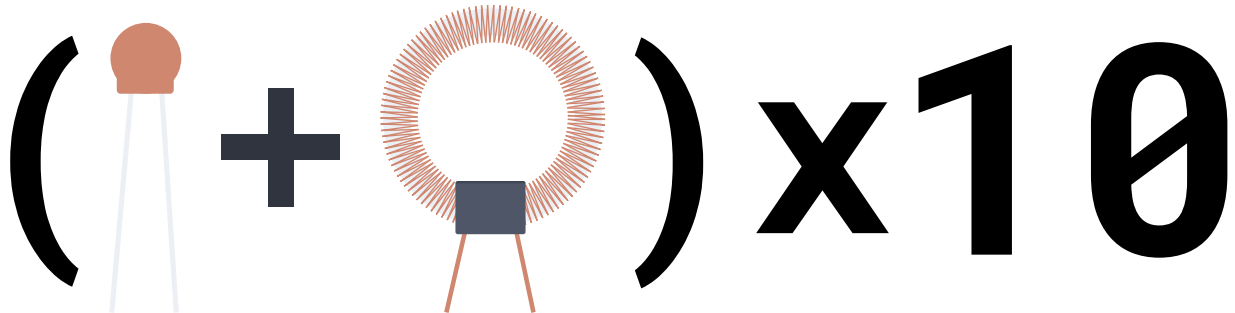
Svingningskredsløb

Kredsløbet virker som et pendul der svinger frem og tilbage. Svingningen skyldes, at energien bølger frem og tilbage mellem de to energibeholdere (induktor og kondensator).



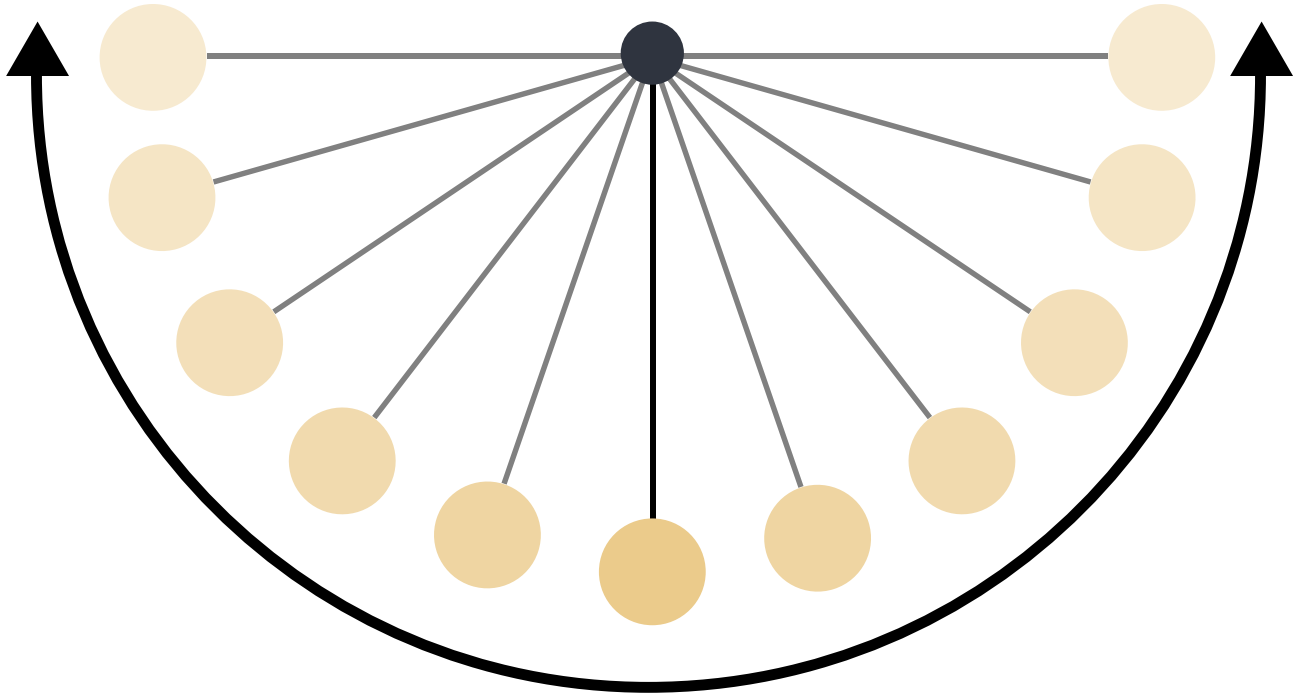
Svingningskredsløb

Hvor meget 'pendulet' svinger frem og tilbage afgøres af hvor mange af kredsløbende der er - eller hvordan de er bygget.



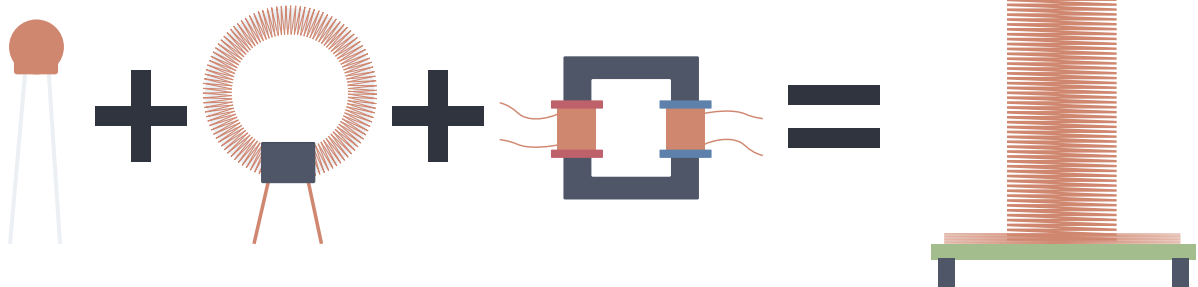
Svingningskredsløb

Med flere kredsløb forstærkes udsvingene i vores magnetfelter.

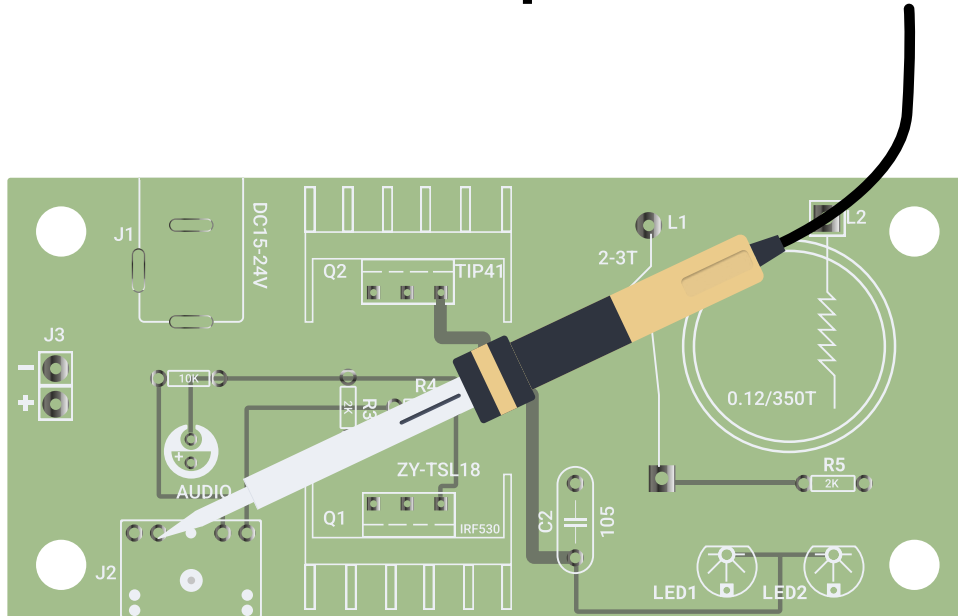


Tesla Spole (Tesla Coil)

Med svingningskredsløbet på plads skal vi bare gøre spændingen høj nok til at energien bliver luftbåren. Dette gør vi med en transformator, som bare kører ud i luften. Og så skal der bare sættes strøm til - sådan virker en Tesla Spole!

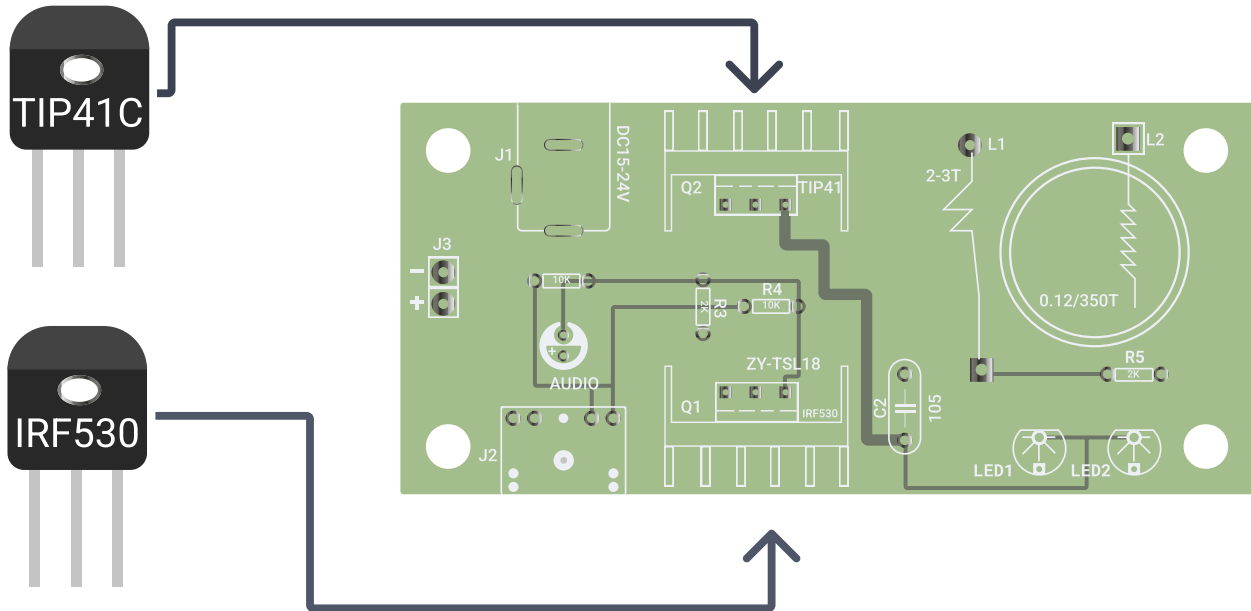


Tips of Tricks når i samler jeres Teslaspoler



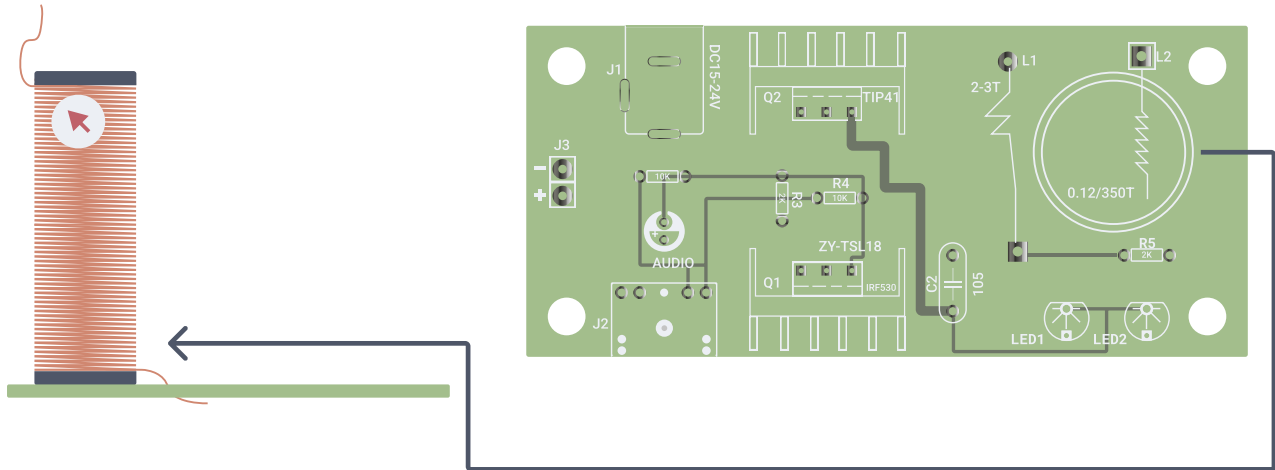
Tip 1: Vend jeres transistorer rigtigt!

TIP41C i toppen, IRF530 i bunden!



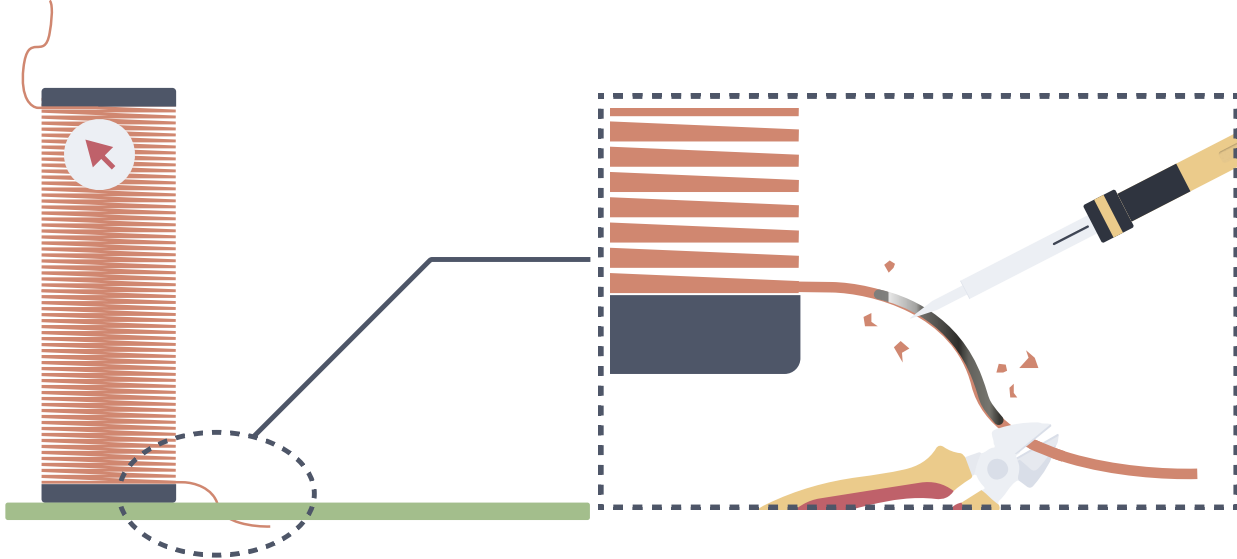
Tip 2: Vend jeres kobber spoler rigtigt!

Pilen på spolen indikerer hvilken ende der skal vende op!



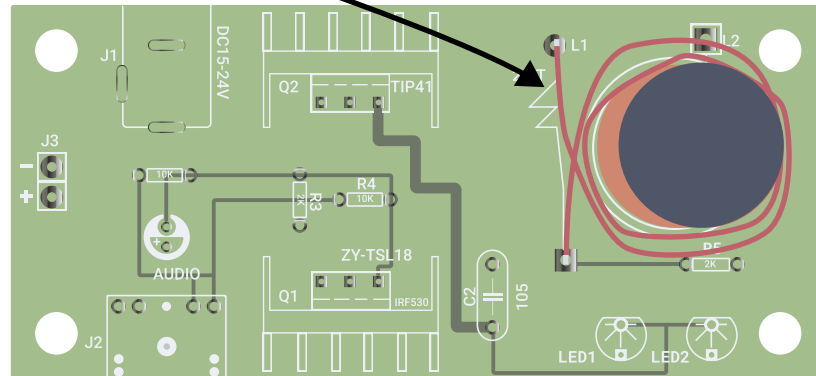
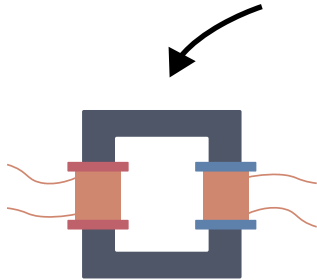
Tip 3: Fjern det beskyttende lag på kobberet der hvor i skal lodde!

Ved at bruge en hobbykniv eller en lighter kan i forsigtigt fjerne det beskyttende, hvilket gør det muligt for jer at lodde ledningen fast!

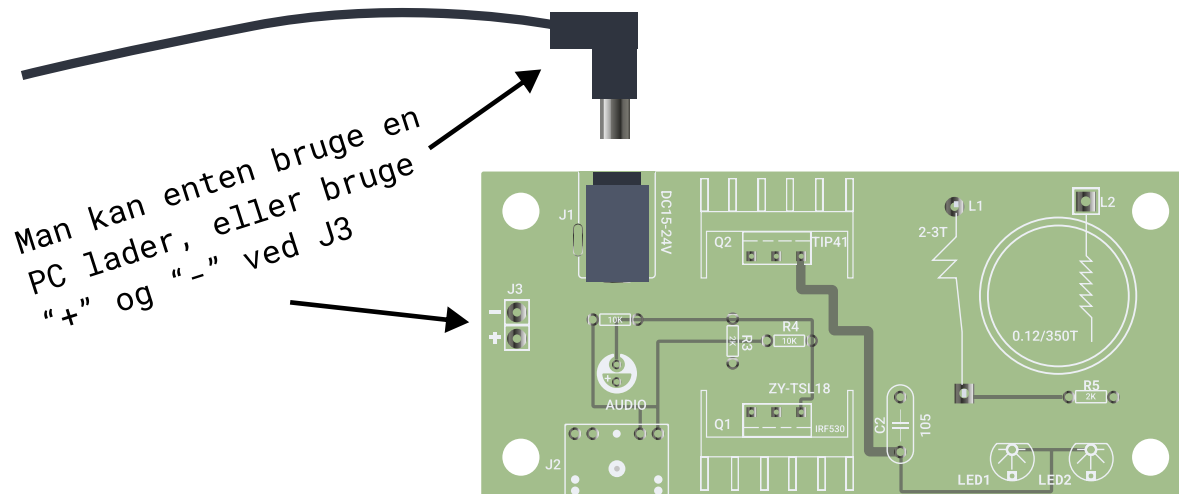


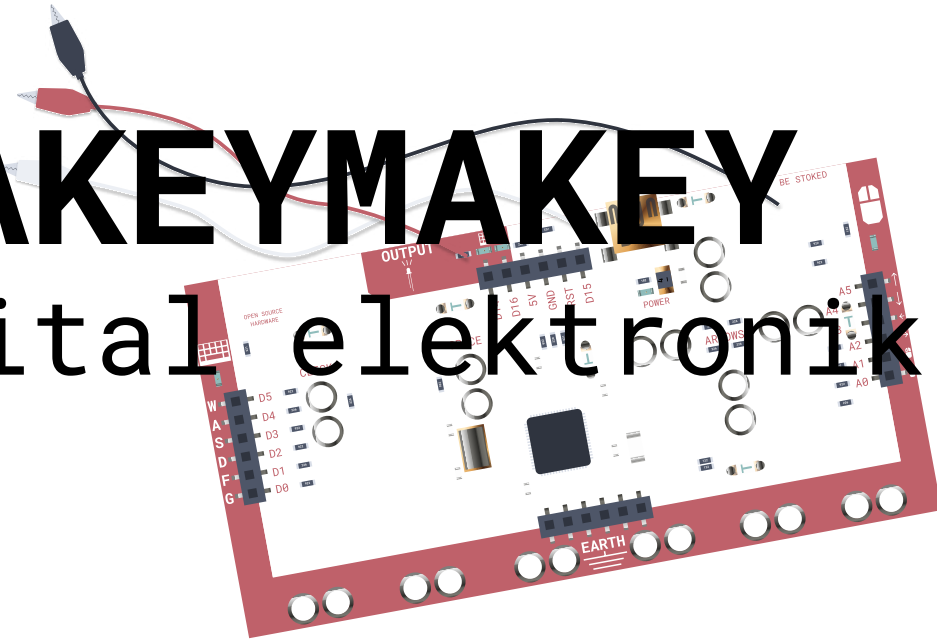
Tip 4: Kør Ledningen 2-3 gange rundt om spolen

I bygger en transformer, lidt ligesom med vores strømforsyning!



Tip 5: Det er flere steder at give den strøm



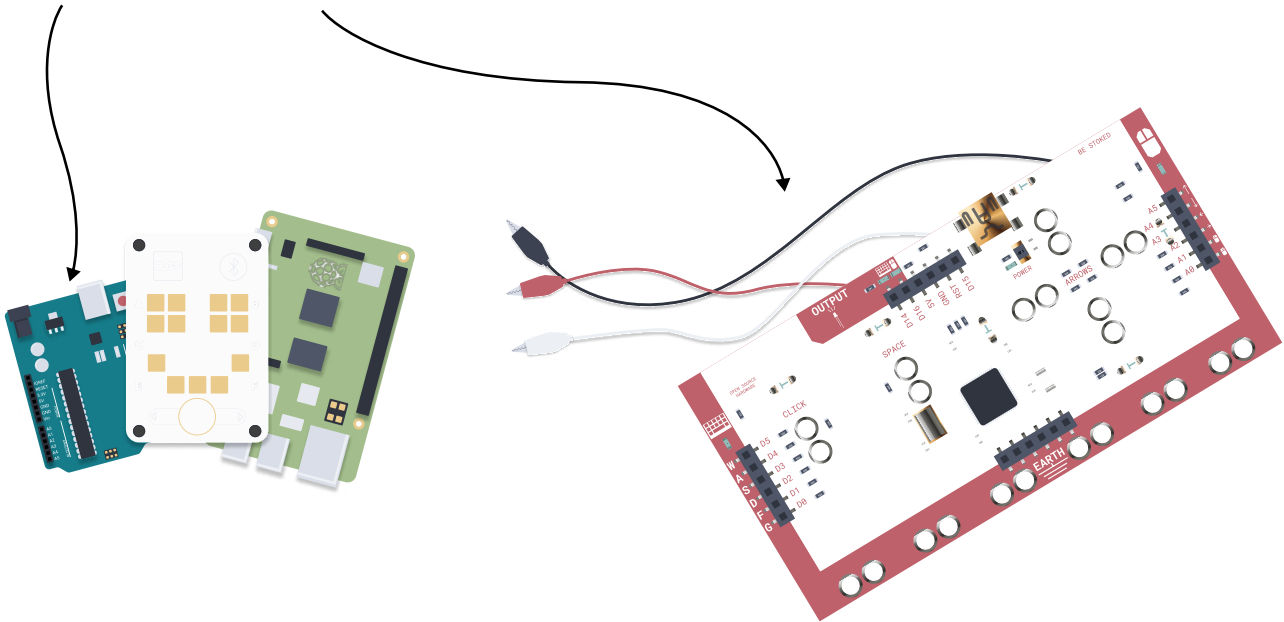


MAKE YMAKEY

Digital elektronik

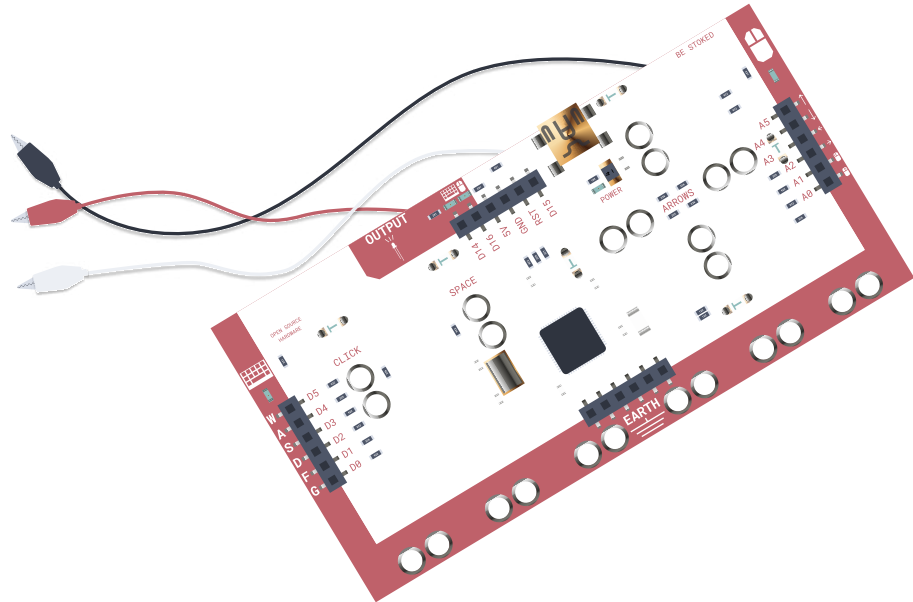
Platforme for digital elektronik

De næste kapitler vil være med fokus på digital elektronik. Vi bruger ordet platforme til at beskrive bestemte PCB'er og deres tilhørende software og hardware komponenter. Vi kommer til at arbejde med Arduino og MakeyMakey.



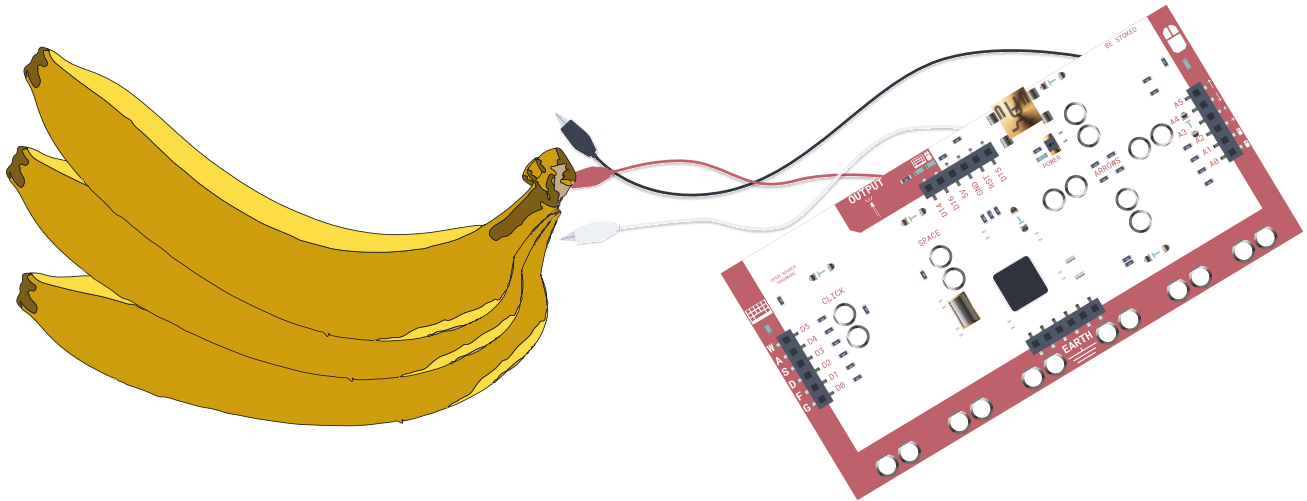
Hvad er MakeyMakey?

MakeyMakey er et værktøj til at lære alle at være 'opfindere'. Den er opfundet af MIT-uddannede ingeniører, som havde en vision om at hjælpe interesserede i at forstå elektronik. Det 'den' kan, er at være et tastatur, hvor vi designer knapperne.



Basicly, så er den et tastetur-PCB, som ikke har nogen knapper.

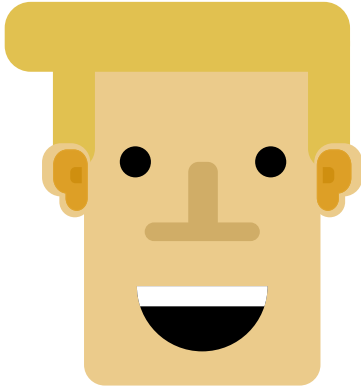
Tilføjer vi programmeringssproget Scratch kan vi lave mange sjove lege og øvelser som handler om elektronik. Men hvad er Scratch og hvad ved vi om kodning og programmering?



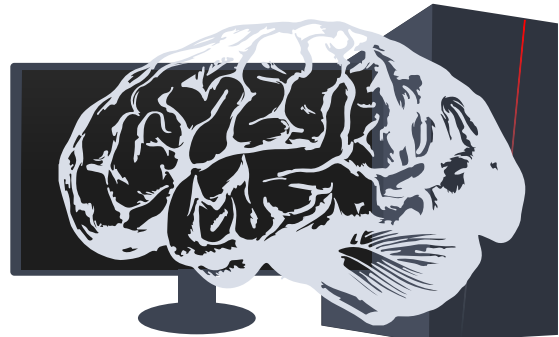
Kodning 1-0-1

Vi styrer computere igennem kode. Kode dækker over mange forskellige sprog, metoder, niveauer, men handler grundlæggende om at kunne få computere til at gøre nogle bestemte ting fx styre en robotarm, sende emails, osv.

“Hello there!”

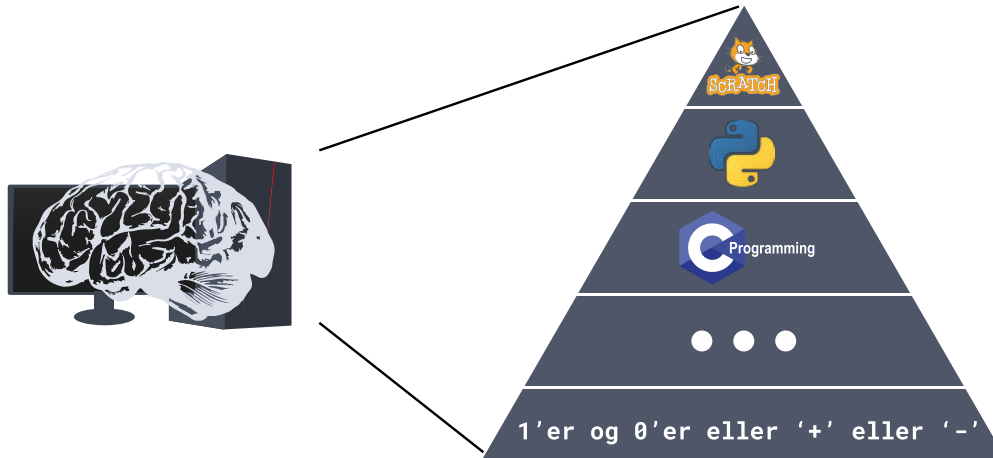


“General Kenobi”



Hvad mener vi med sprog og niveauer?

I gamle dage brugte man virkelig lang tid på at lære at styre computere, da man dengang kun kunne kommunikere med computere igennem positiv og negativ strøm skrevet i 1-taller og 0'er. Tænkt det som en slags morsekode for computere. I dag er det meget nemmere, da andre har udviklet avancerede sprog, som computere forstår.



Links til jer, som gerne vil lære mere om det her!

Hvis man er meget interesseret i kodning og programmering anbefaler vi at man hopper ind på de følgende hjemmesider!

[khanacademy.org](https://www.khanacademy.org/) - har mange videoer omkring computing, datalogi og programmering. Og alt er gratis!

<https://www.instructables.com/> - forum med projekter fra alt mellem himmel og jord

<https://codecombat.com/> - et spil hvor man lærer python eller JavaScript

Alternativt kan man selvfølgelig bruge 'youtube.com' eller bruge Coding Pirates (Coding Pirates koster dog penge)

Målet for i dag er at komme i gang med MakeyMakey og Scratch!

<https://scratch.mit.edu/>



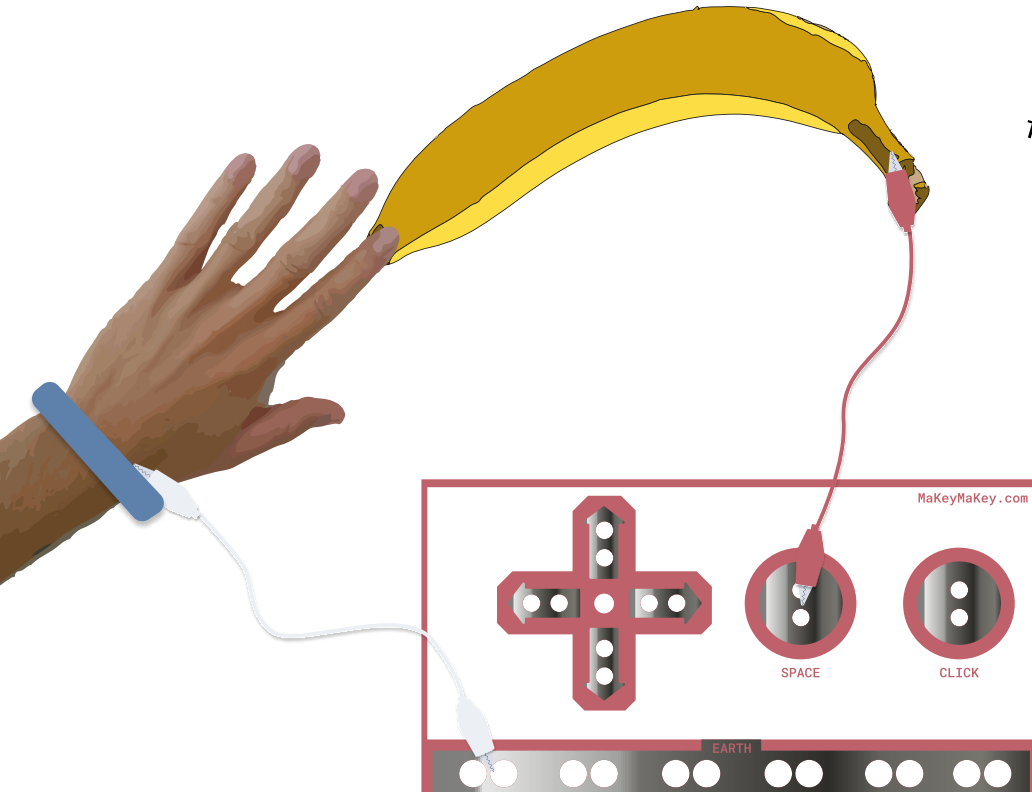
Vi skal løse 2 Opgaver i dag!

Først, skal vi lave et kreativt klaver! Når vi har spillet lidt musik skal vi lave vores første program på Scratch. Derefter vil der være tid til at lodde de sidste projekter færdige eller kaste sig ud i nye MakeyMakey udfordringer!

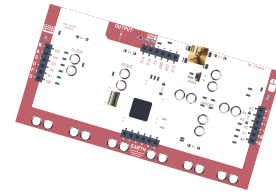
"You better do
this, or else!"



Det kreative klaver!



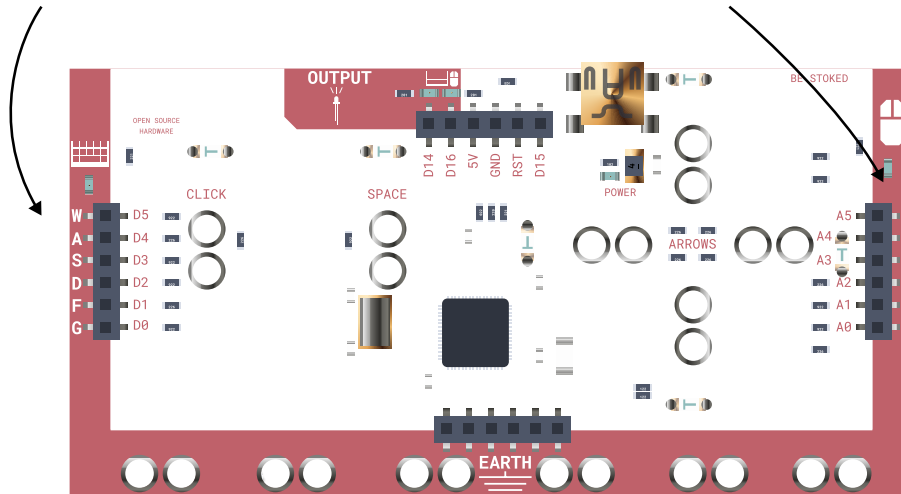
På den her side er der flere muligheder!



Det kreative klaver!

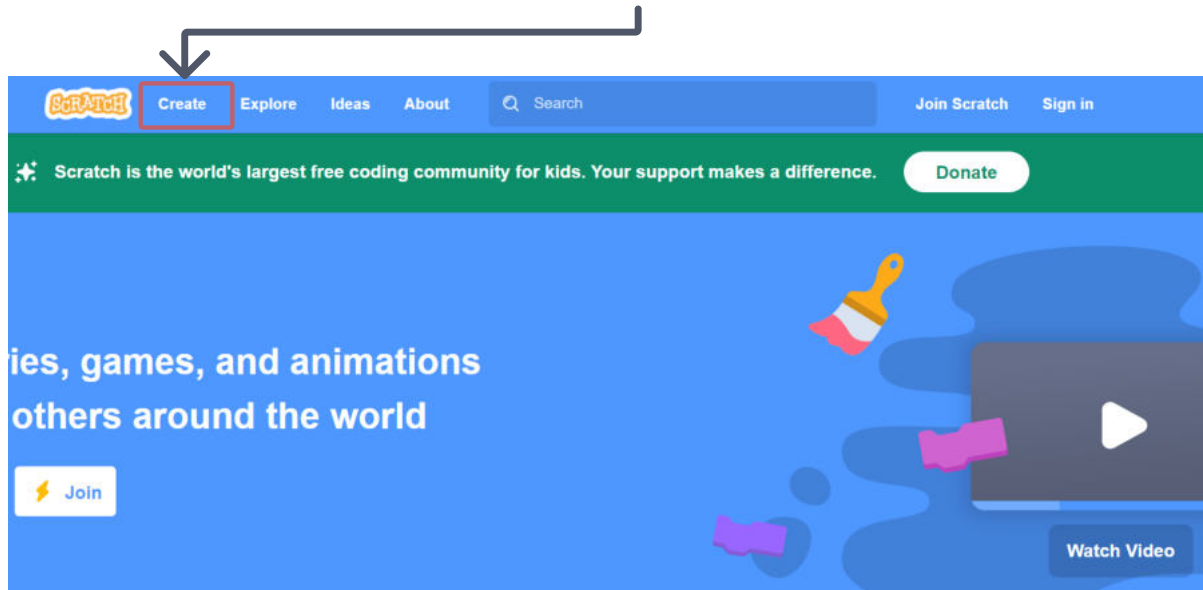
<https://apps.makeymakey.com/piano/>

Det står på siden hvad den enkelte betyder

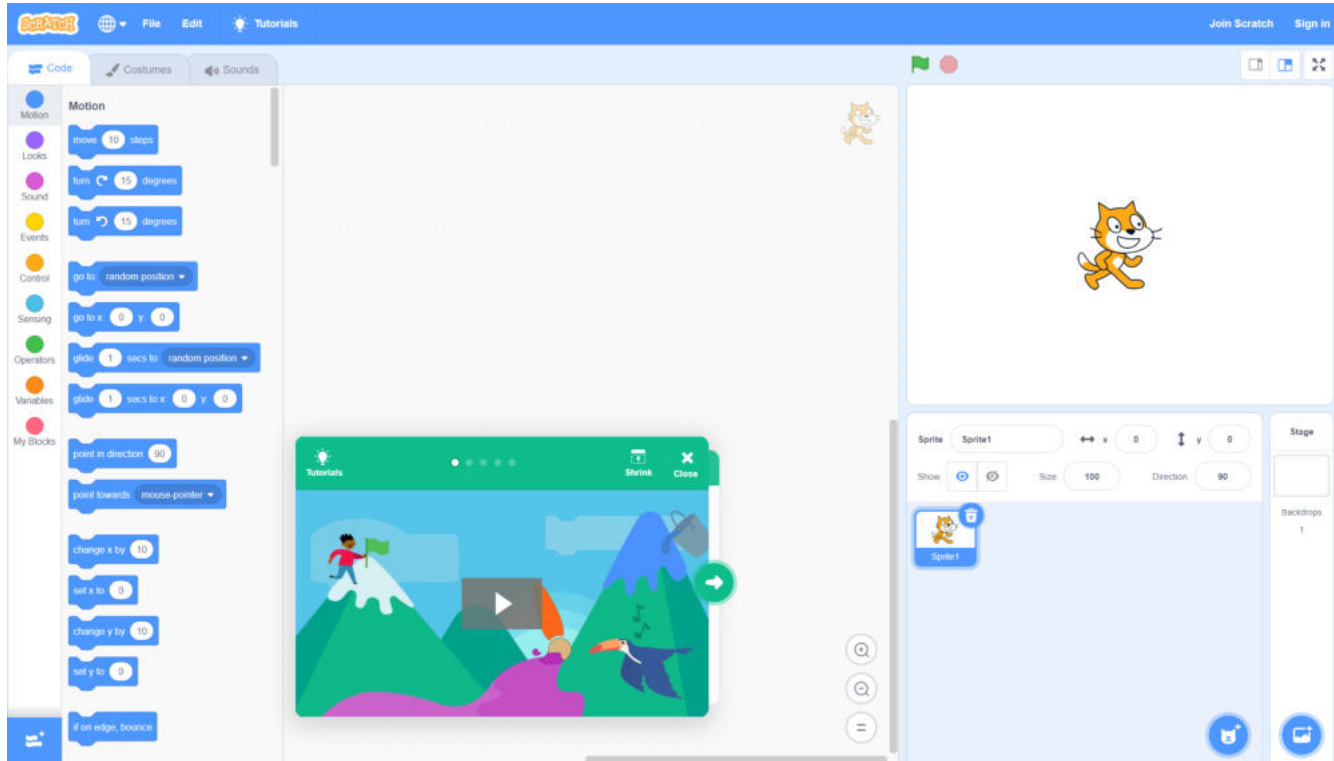


Målet for i dag er at komme i gang med MakeyMakey og Scratch!

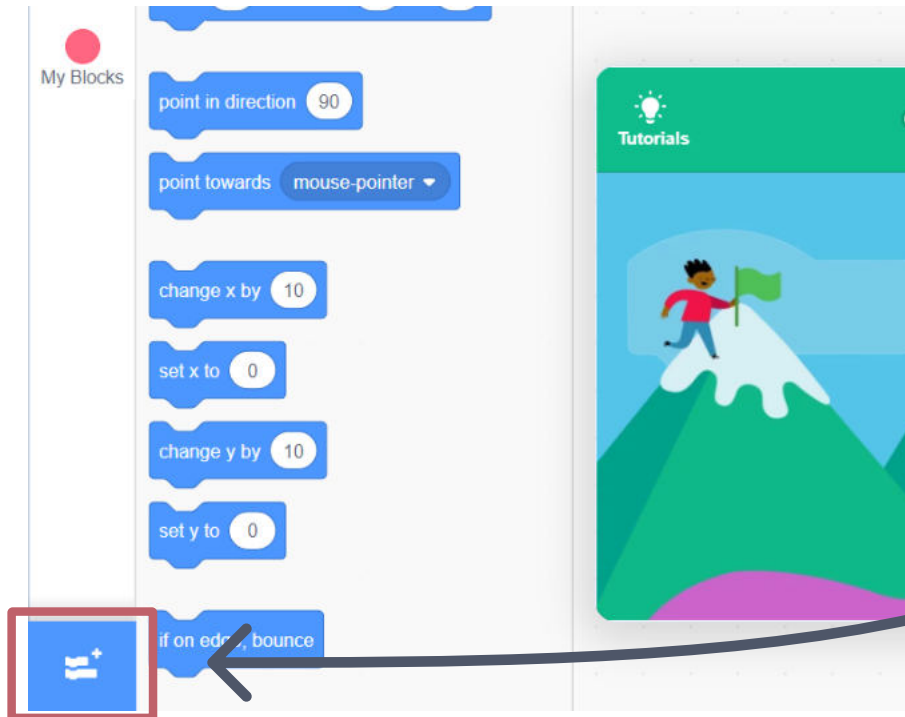
Gå ind på hjemmesiden: <https://scratch.mit.edu/>












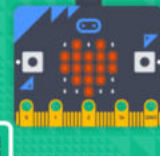





Dette vil være jeres “legeplads”



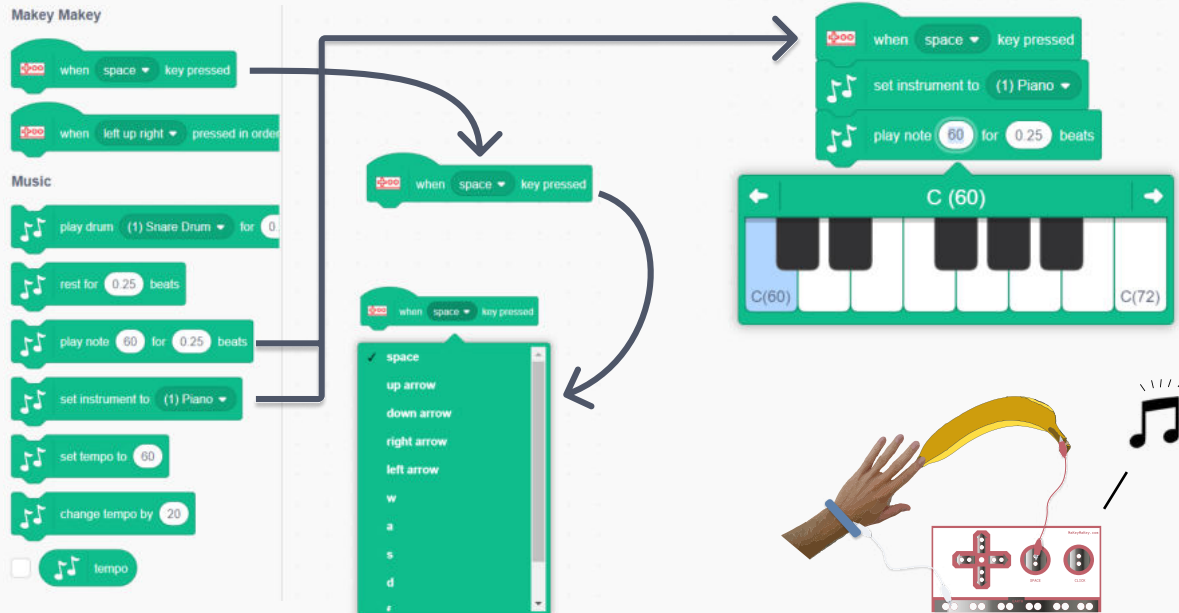
Vi skal først tilføje vores MakeyMakey blocks



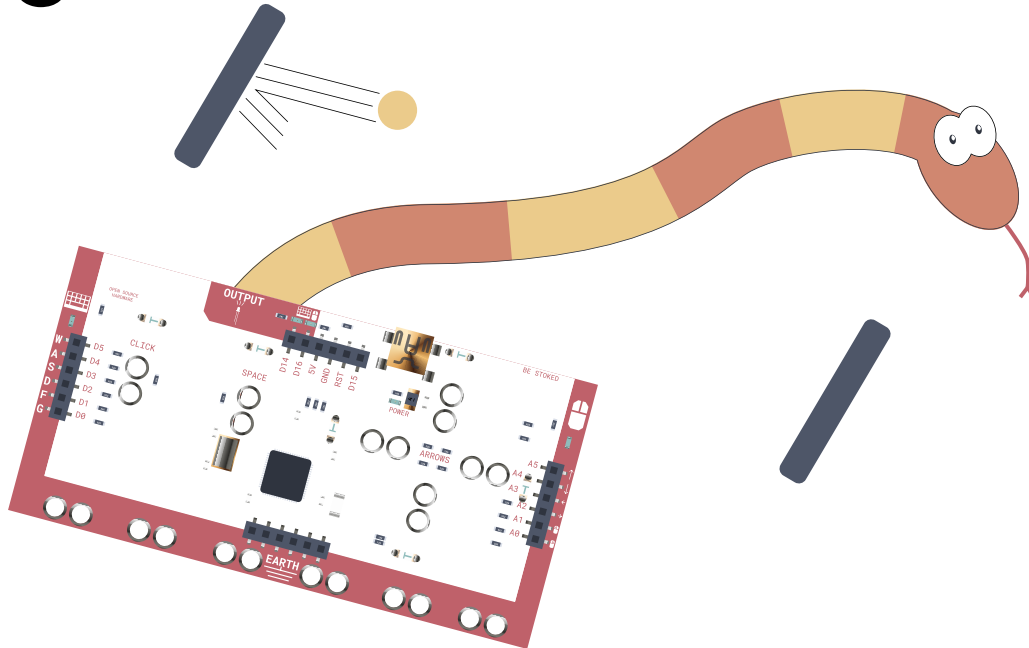
I kan tilføje både “Music” og “Makey Makey”

 <p>Music Play instruments and drums.</p>	 <p>Pen Draw with your sprites.</p>	 <p>Video Sensing Sense motion with the camera.</p>	 <p>Text to Speech Make your projects talk.</p> <p>Requires  Collaboration with Amazon Web Services</p>
 <p>Translate Translate text into many languages.</p> <p>Requires  Collaboration with Google</p>	 <p>Makey Makey Make anything into a key.</p> <p>Requires  Collaboration with JoyLabz</p>	 <p>micro:bit Connect your projects with the world.</p> <p>Requires   Collaboration with micro:bit</p>	 <p>LEGO MINDSTORMS EV3 Build interactive robots and more.</p> <p>Requires   Collaboration with LEGO</p>

Når i har tilføjet jeres blokke, så kan i trække de funktioner ind i har brug for!



MakeyMakey og digital elektronik 2

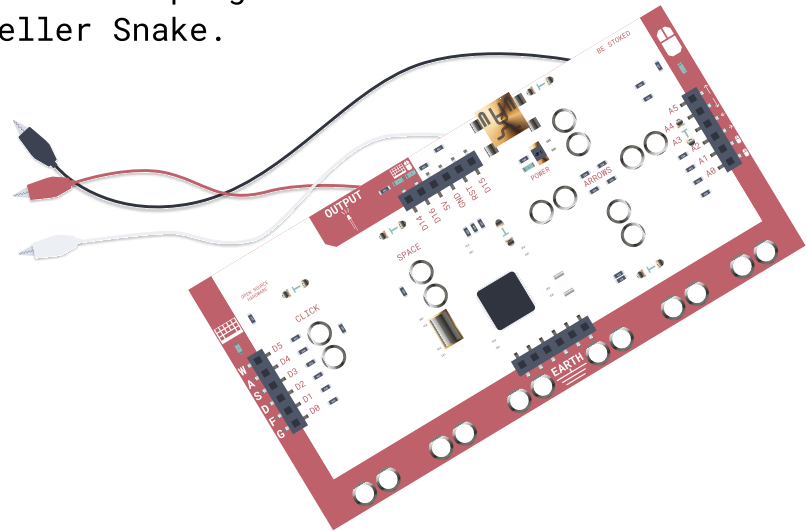
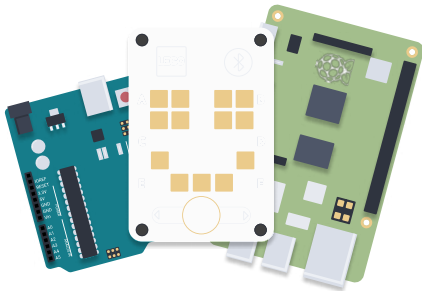


Platforme for digital elektronik

Vi kommer til at arbejde videre med MakeyMakey, men nu med fokus på at kode noget lidt mere avanceret.

Kodningen kommer stadig til at foregå i Scratch, hvor der skal udforskes nogle af de forskellige kodningsbegreber.

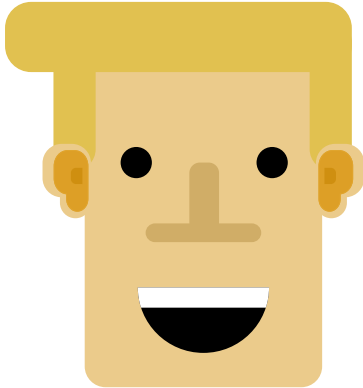
Målet er at alle lykkedes med at programmere et arkadespil, her enten Pong og/eller Snake.



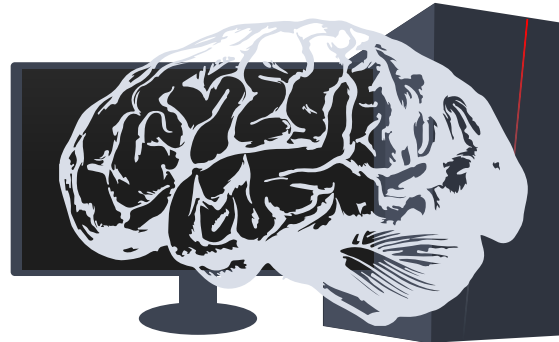
Kodning 1-0-1

I forrige kapitel snakkede vi om at computere forstår bestemte 'sprog'. I dag kommer vi til at kigge lidt nærmere på nogle af de generelle strukturer, som størstedelen af alle programmeringssprog indeholder.

"Hello there"



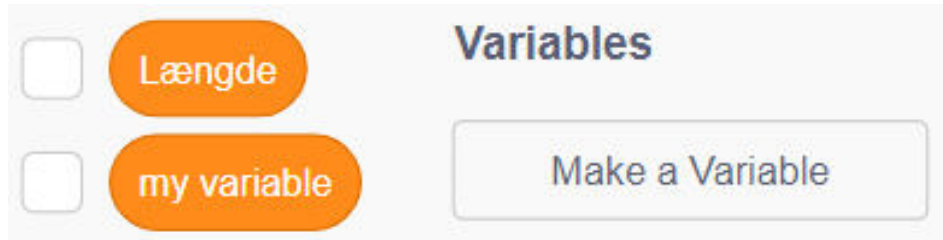
"General Kenobi"



Variabel

Begrebet Variabel dækker over noget computeren skal huske eller arbejde med. Dette kan fx være et tal, et ord eller et sæt koordinater. En variabel har et navn og én eller flere værdier som computeren husker.

I scratch ser en variabel sådan her ud!



Statements ("Betingelser")

Statements, eller betingelser, er bestemt udtryk i koden, som vi bruger til at få bestemt ting til at ske.

I dag skal vi fx teste om vores slange kommer til at bide sig selv eller om den spiser et æble.

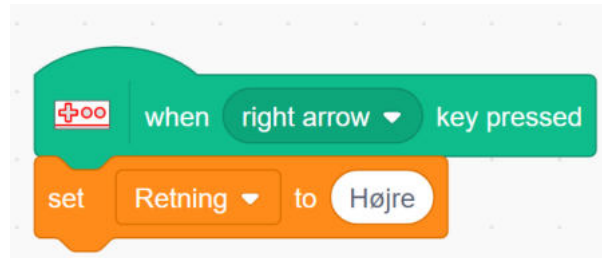
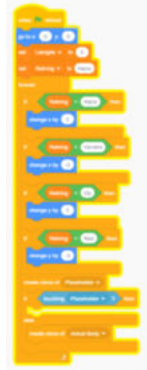
Det her er et if-statement (en "hvis" betingelse, altså, hvis "dette" sker, så skal "det her" ske)



Funktion

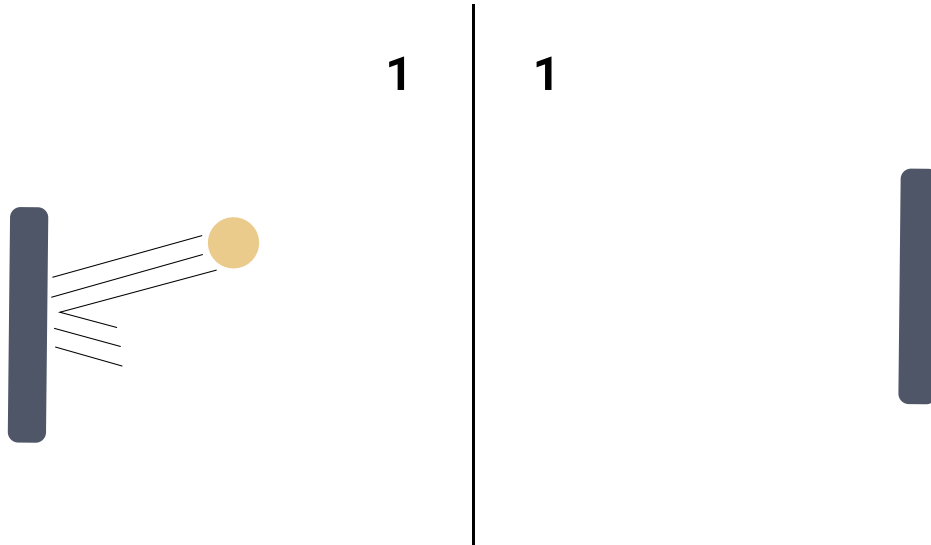
En funktion er en eller flere linje kode, som får computeren til at gøre noget. Funktioner arbejder med variable, og kan fx indeholde matematiske udtryk. Nogle gange skal vi selv skrive funktionerne, og andre gange kan vi udnytte at der er nogen som har gjort det for os.

Funktioner ser sådan her ud i Scratch

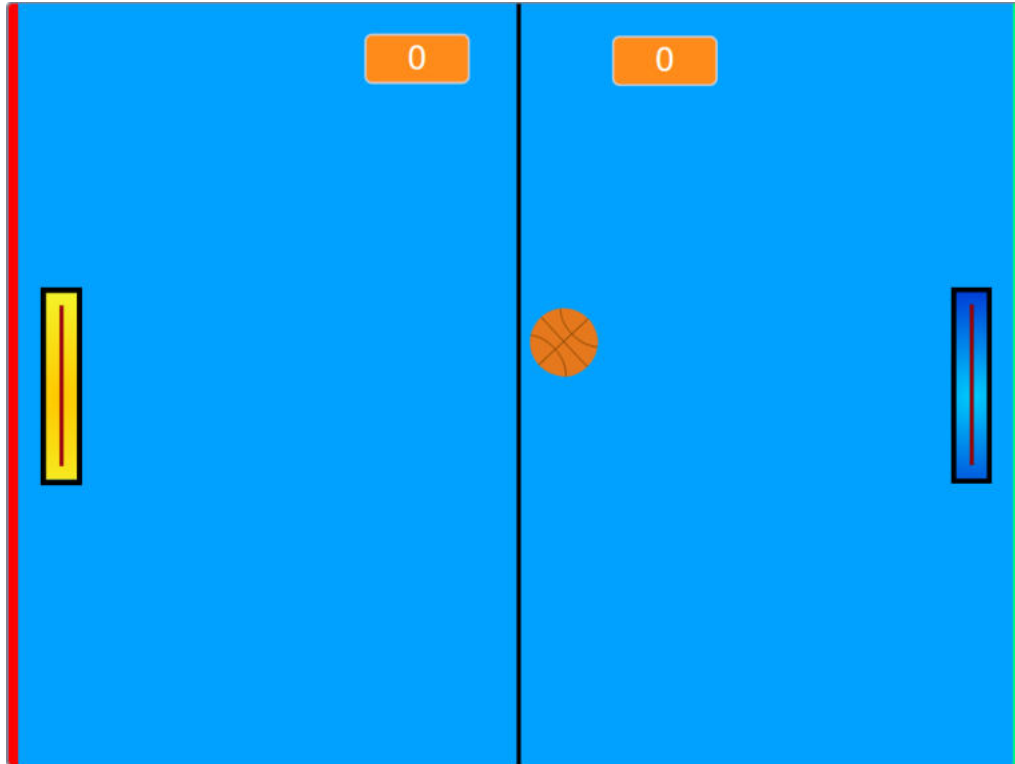


Pong!

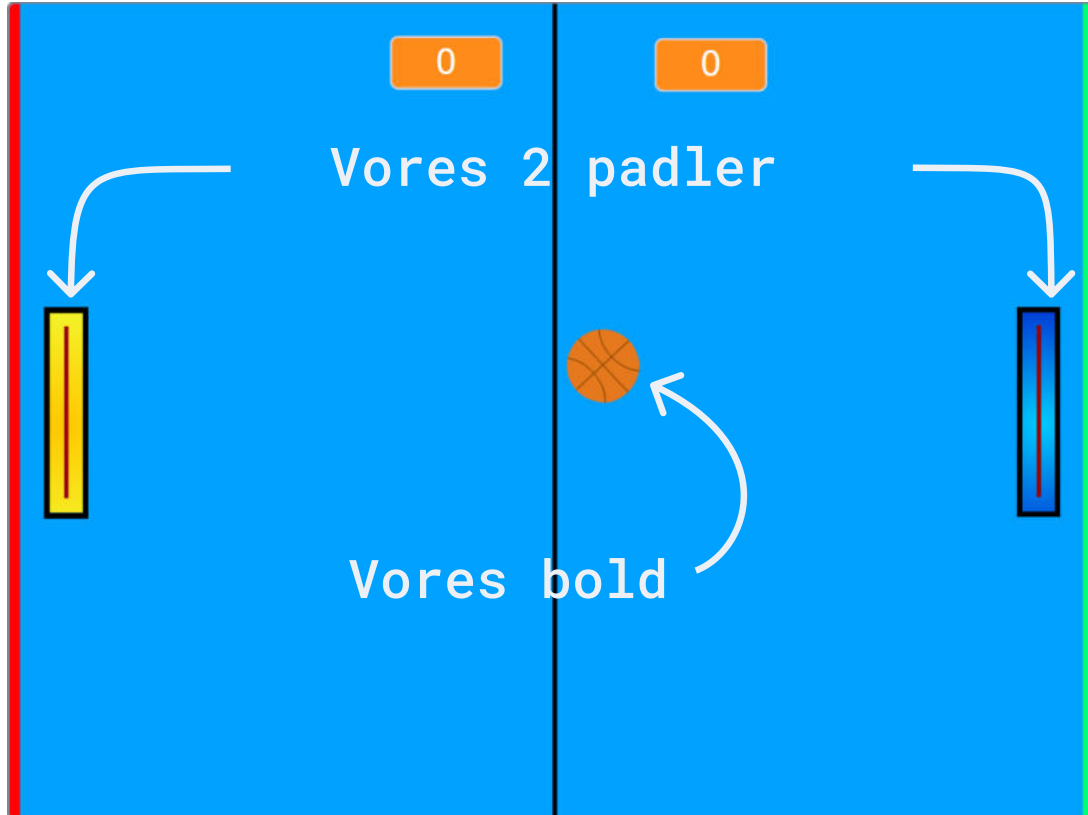
Det første Arkadespil vi skal kode i scratch er PONG. Spillet er fra 1972 og blev udviklet af Atari. Spillet er kendt som et af de første arkadespil.



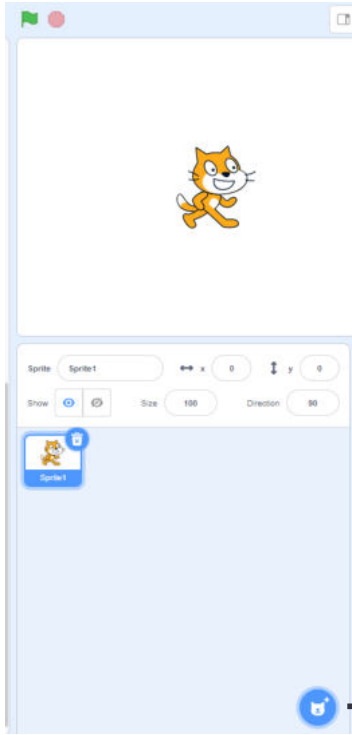
Slut-resultatet vil se sådan ud!



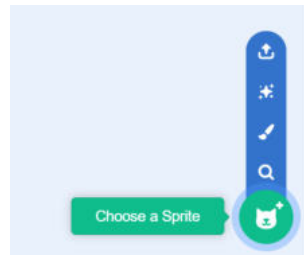
Der skal programmeres 3 ting:



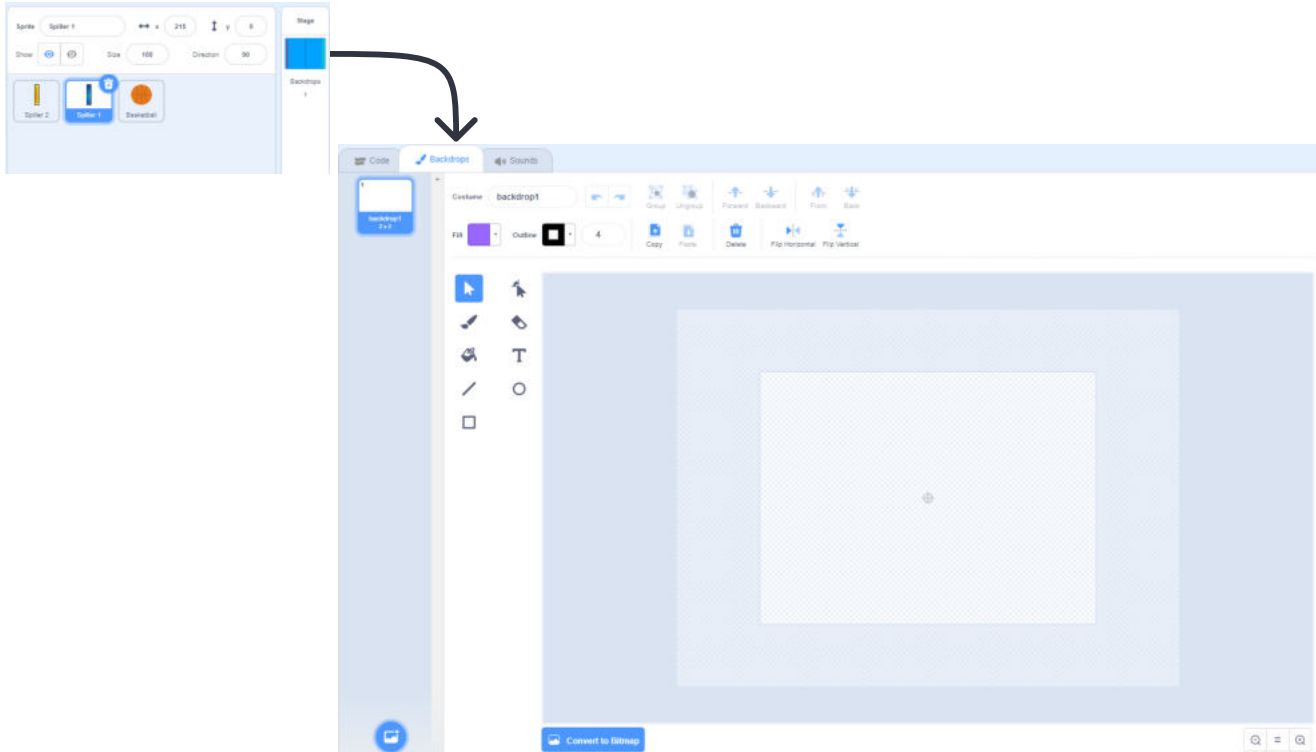
Start med at finde vores figurer!



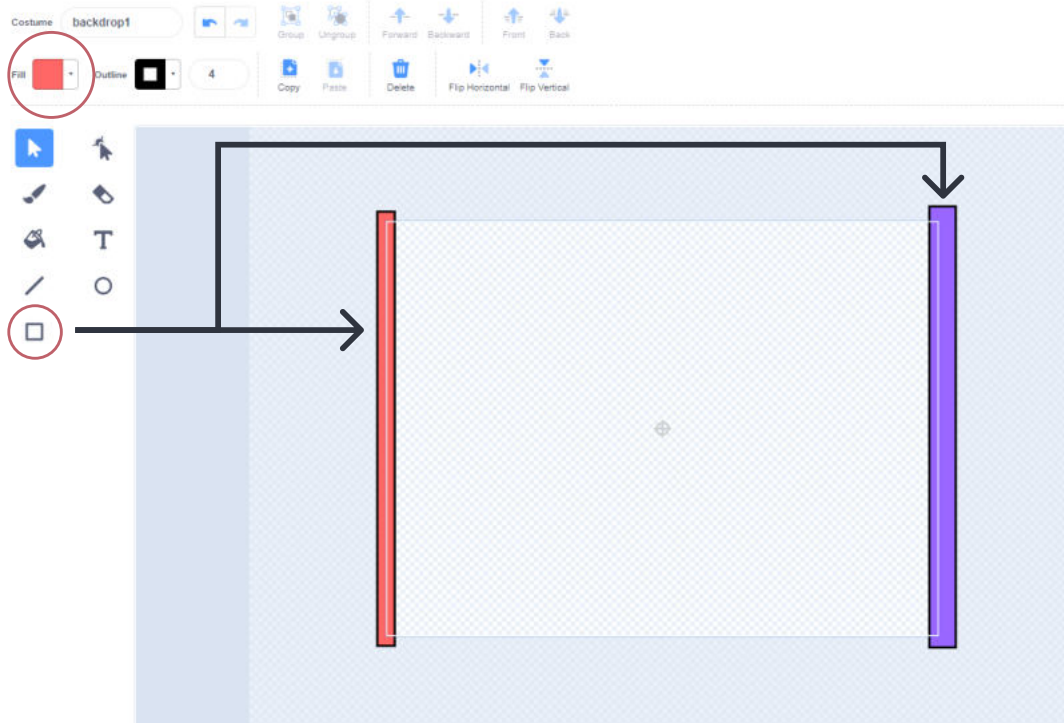
I kan selv bestemme (eller tegne!) jeres egne figurer! Så længe at i har en rund figur som bold og aflang figur som paddle



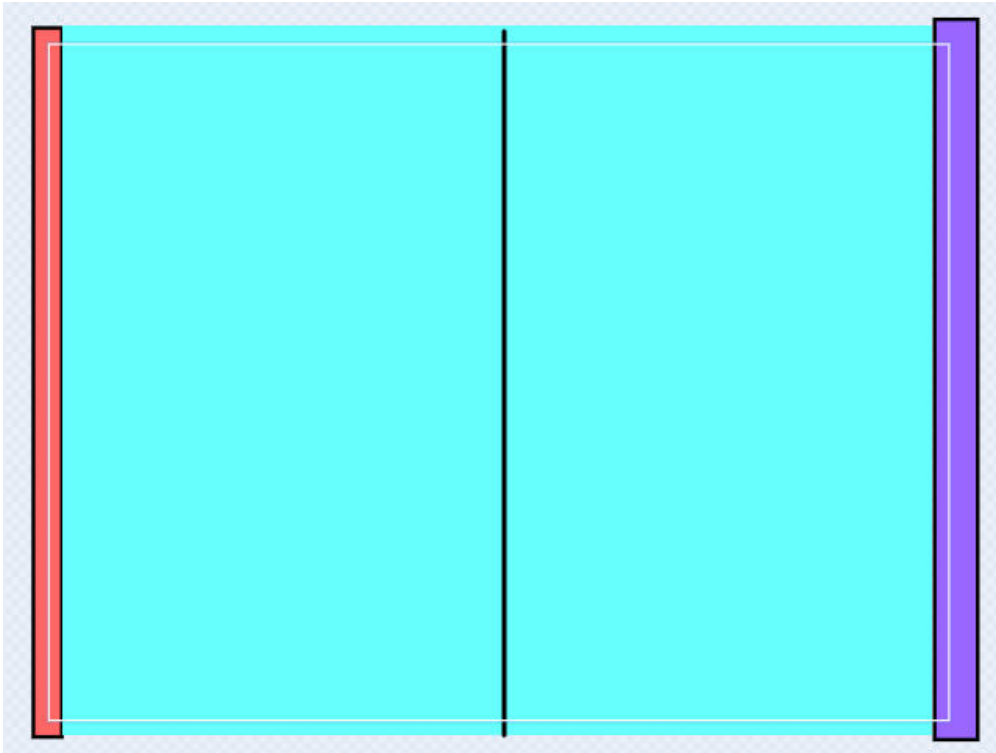
Derefter, tilpas jeres baggrund!

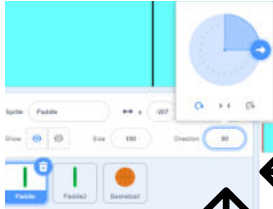


Lav 2 rektangler i forskellige farver
der kun lige går ind i firkanten



Tilføj evt. en ny baggrundsfarve og en
midterlinje





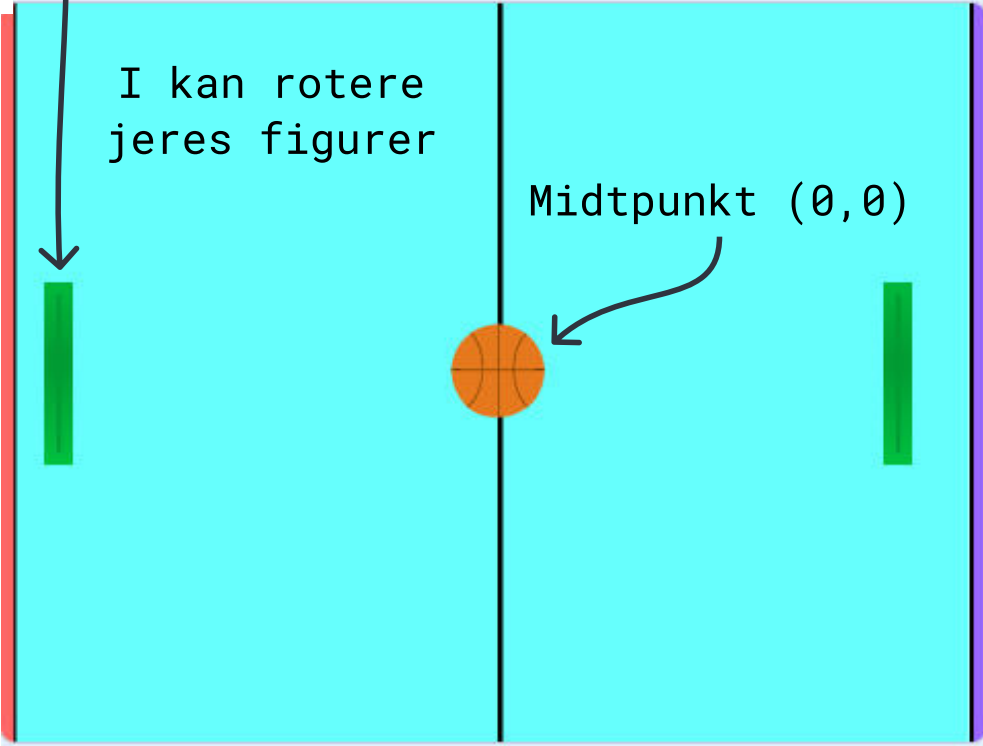
I burde nu se dette!

480 lang (-240 -> 240)

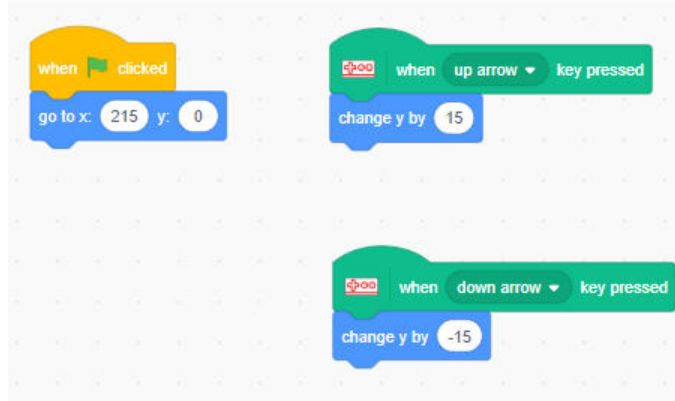
360 bred (-180 -> 180)

I kan rotere
jeres figurer

Midtpunkt (0,0)



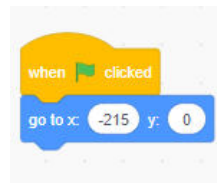
Programmering: Padlerne!



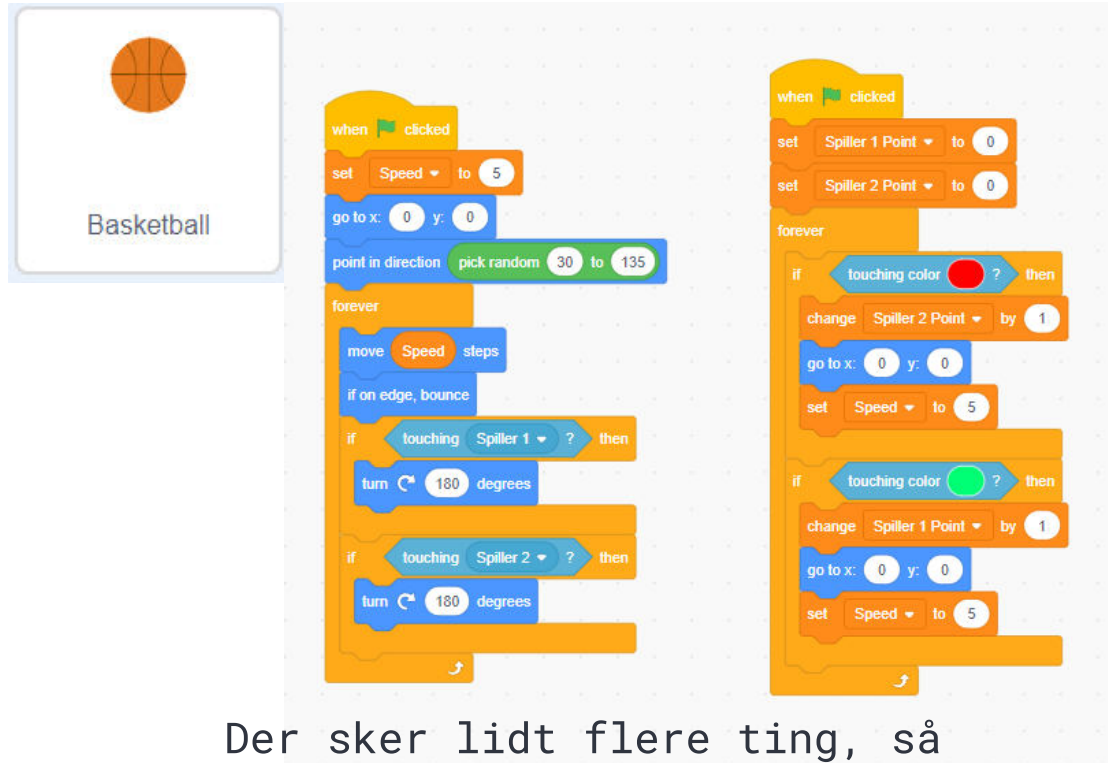
Når i bruger op og ned pilene på jeres Makey Makey's, så vil padlen også køre op og ned. Hvis i ændre værdien 15 til et højere tal vil den bevæge sig hurtigere!

Gør det samme med den anden paddle!

For venstre paddle er det bare -215 istedet for 215



Programming: Bolden!



The image shows a Scratch project titled "Basketball" with a basketball icon. It contains two scripts for a ball's movement and scoring.

Left Script (Ball Movement):

- when green flag clicked
- set Speed to 5
- go to x: 0 y: 0
- point in direction pick random 30 to 135
- forever loop:
 - move Speed steps
 - if on edge, bounce
 - if touching Spiller 1 ? then: turn 180 degrees
 - if touching Spiller 2 ? then: turn 180 degrees

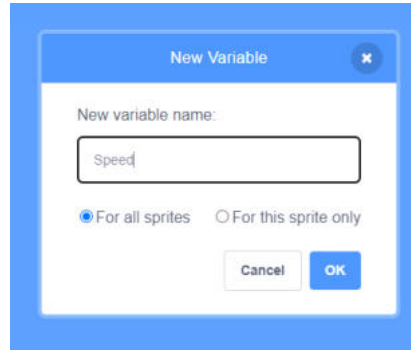
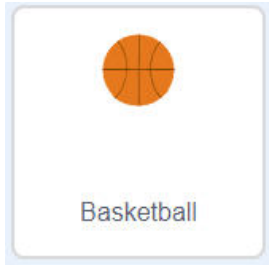
Right Script (Scoring):

- when green flag clicked
- set Spiller 1 Point to 0
- set Spiller 2 Point to 0
- forever loop:
 - if touching color red ? then:
 - change Spiller 2 Point by 1
 - go to x: 0 y: 0
 - set Speed to 5
 - if touching color green ? then:
 - change Spiller 1 Point by 1
 - go to x: 0 y: 0
 - set Speed to 5

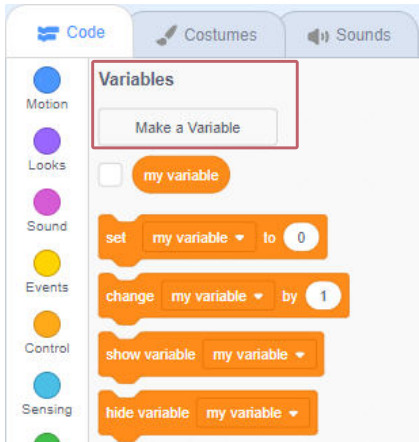
Der sker lidt flere ting, så vi tager den i flere trin!

Programmering: Bolden!

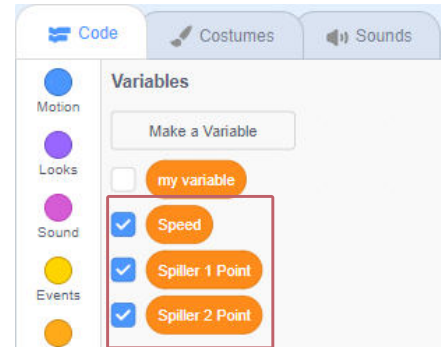
Step 1: Lav nye variable!



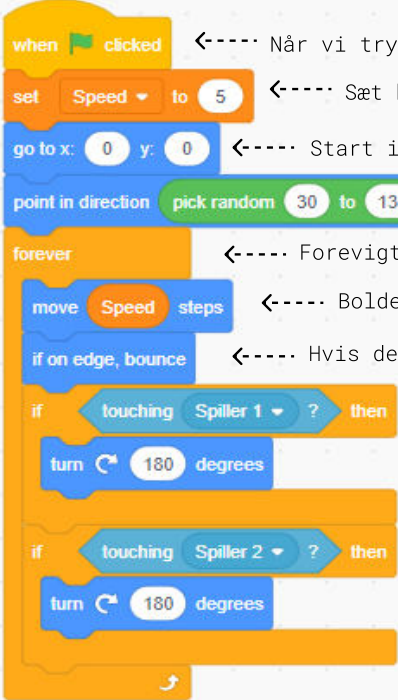
Navne på variable betyder ikke så meget, så længe i selv kan huske hvad de står for.



Lav en variabel der for hastigheden af bolden (speed) og en pointtæller for begge spillere



Programmering: Bolden! Step 2: Hvad skal der ske når bolden rører padlen eller øvre/nedre væg?



The image shows a Scratch script for a ball simulation. The script starts with a 'when clicked' event block, followed by a 'set Speed to 5' block, a 'go to x: 0 y: 0' block, and a 'point in direction pick random 30 to 135' block. A 'forever' loop contains a 'move Speed steps' block, an 'if on edge, bounce' block, an 'if touching Spiller 1 ? then' block with a 'turn 180 degrees' block, and another 'if touching Spiller 2 ? then' block with a 'turn 180 degrees' block.

when clicked ←----- Når vi trykker på det grønne flag

set Speed to 5 ←----- Sæt hastigheden til 5

go to x: 0 y: 0 ←----- Start i midten af banen

point in direction pick random 30 to 135 ←----- Flyv i en tilfældig retning

forever ←----- Forevigt skal dette fortsætte!

move Speed steps ←----- Bolden skal bevæge sig med en hastighed (speed)

if on edge, bounce ←----- Hvis den rammer en væg skal den "bounce" væk

if touching Spiller 1 ? then ←----- Hvis bolden rammer spiller 1, så skal den vender sig om 180 grader og flyve den anden vej

turn 180 degrees

if touching Spiller 2 ? then ←----- Hvis bolden rammer spiller 2, så skal den vender sig om 180 grader og flyve den anden vej

turn 180 degrees

Programmering: Bolden!

Step 3: Hvad skal der ske når bolden rører målene bag ved spillerne?



←----- Når vi trykker på det grønne flag

set Spiller 1 Point to 0 ←----- Sæt spiller 1 score til 0

set Spiller 2 Point to 0 ←----- Sæt spiller 2 score til 0

forever ←----- Forevigt skal dette fortsætte!

if touching color red? then ←----- Hvis bolden rører en rød farve (farven på den ene ende)

change Spiller 2 Point by 1 ←----- Giv spiller 2 ét point

go to x: 0 y: 0 ←----- Gå tilbage til midterposition

set Speed to 5 ←----- Sæt hastigheden tilbage til 0

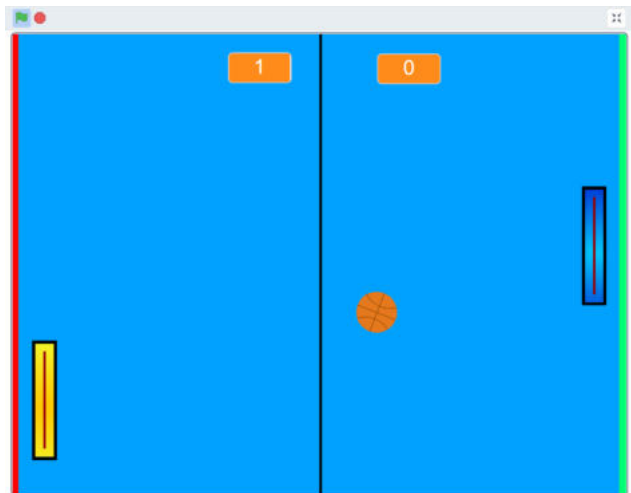
if touching color green? then ←----- Det samme for den grønne væg, men nu med spiller 1

change Spiller 1 Point by 1

go to x: 0 y: 0

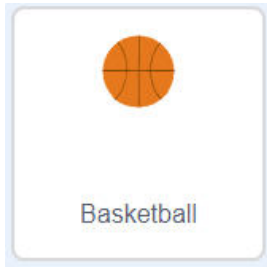
set Speed to 5

Prøv spillet i denne tilstand ved at trykke på det grønne flag!



Spillet er let simpelt og ikke særligt udfordrende, så lad os gøre koden lidt bedre!

Tilføj ekstra udregninger så bolden bouncer mere realistisk fra jeres padler!



```
when clicked
  set Speed to 5
  go to x: 0 y: 0
  point in direction: pick random 30 to 135
  forever
    move Speed steps
    if on edge bounce
    if touching Spiller 1 ? then
      set Retning_Udregning to Verstre_Y - y position / 46.5 * 75
      point in direction 90 + Retning_Udregning
      change Speed by 1
      start sound Boing
      change pitch effect by 10
    if touching Spiller 2 ? then
      set Retning_Udregning to Højre_Y - y position / 46.5 * 75
      point in direction -90 + Retning_Udregning
      change Speed by 1
      start sound recording1
      change pitch effect by 10
```

```
when clicked
  set Spiller 1 Point to 0
  set Spiller 2 Point to 0
  forever
    if touching color red ? then
      change Spiller 2 Point by 1
      go to x: 0 y: 0
      set Speed to 5
      clear sound effects
    if touching color green ? then
      change Spiller 1 Point by 1
      go to x: 0 y: 0
      set Speed to 5
      clear sound effects
```

Lav først en ny variabel kaldet: "Retning_Udregning"

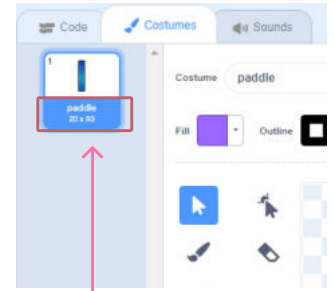
```

when clicked
  set Speed to 5
  go to x: 0 y: 0
  point in direction: pick random 30 to 135
  forever
    move Speed steps
    if on edge, bounce
    if touching Spiller 1? then
      set Retning_Udregning to Venstre_Y - y position / 46.5 * 75
      point in direction 90 + Retning_Udregning
      change Speed by 1
      start sound Boing
      change pitch effect by 10
    if touching Spiller 2? then
      set Retning_Udregning to Højre_Y - y position / 46.5 * 75
      point in direction -90 - Retning_Udregning
      change Speed by 1
      start sound recording1
      change pitch effect by 10
  
```

←---- Vi udregner vores nye retning bolden skal gå. Udregningen er: Venstre paddle Y-koordinat minus Y koordinat for bolden, divideret med halvdelen af paddlens længde ganget med 75. (Venstre_Y-y_position)/46.5*75

↑-- Sæt retningen til 90 grader + vores udregnede vinkel
 ----- Gør hastigheden lidt hurtigere!
 ←---- Afspil en sjov lyd og gør den hurtigere!

Gør det samme for Spiller 2, men nu med -90 grader og Højre_Y variabel



Opret også to nye variable, Venstre_Y og Højre_Y for hver paddle

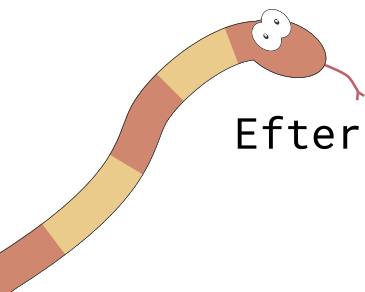
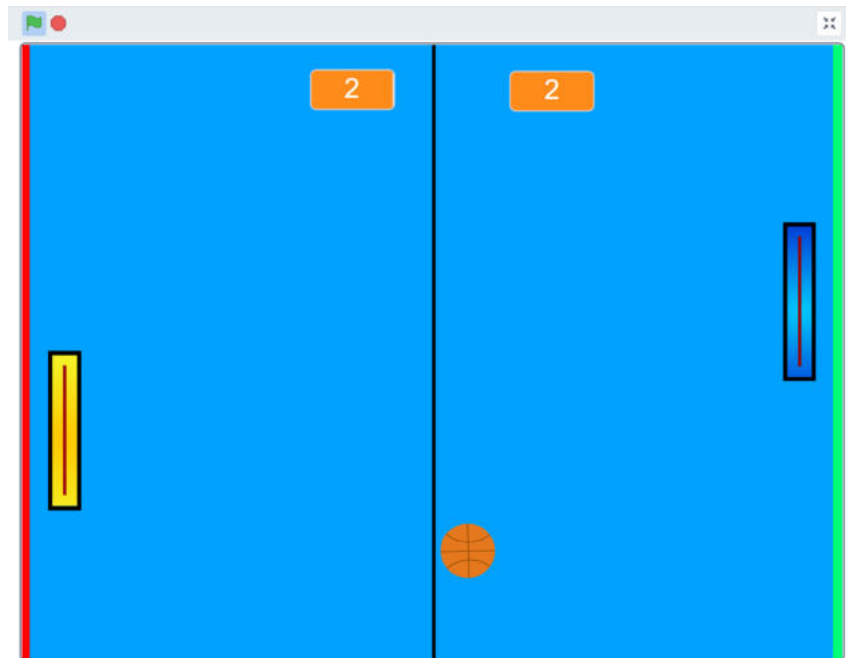
```

when clicked
  go to x: 215 y: 0
  set Højre_Y to y position

when up arrow key pressed
  change y by 15
  set Højre_Y to y position

when down arrow key pressed
  change y by -15
  set Højre_Y to y position
  
```

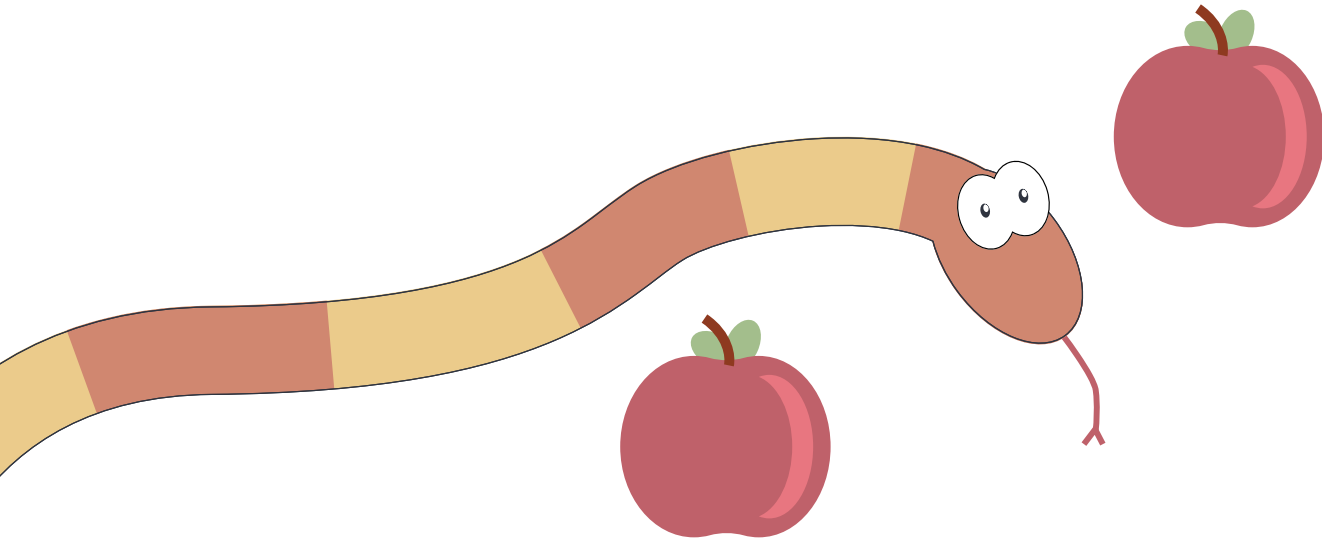
Prøv igen spillet og se hvordan bolden nu bevæger sig!



Efter dette skal vi igang med Snake spillet.

Snake!

Det andet spil vi skal lave er spillet Snake. I snake er man en slange som skal spise æbler på en skærm. Spillet er også meget gammelt, men har overlevet i forskellige variationer som fx "Achtung die Kurve"



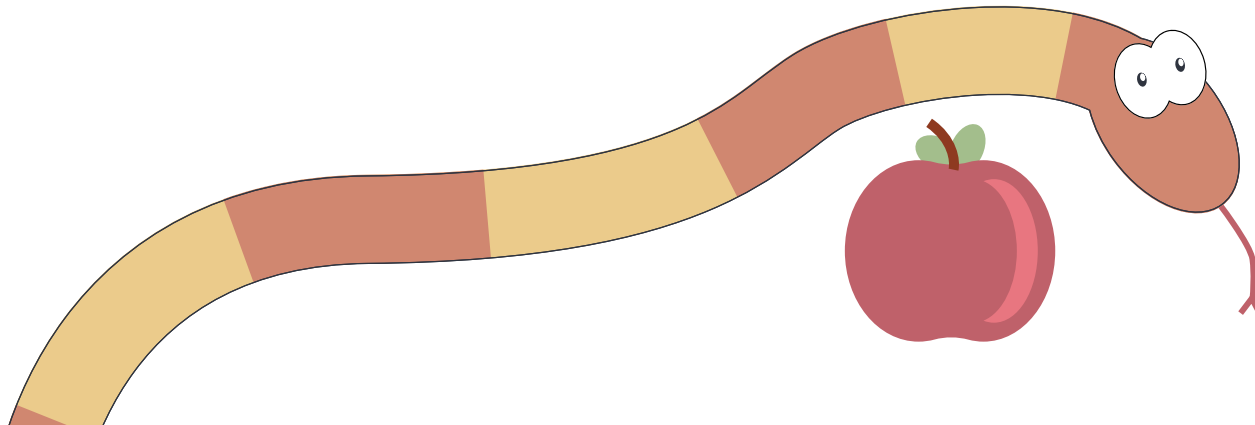
Snake!

Slangen skal kunne dreje i de fire forskellige retninger: Op, ned, højre, venstre

Slangen skal blive længere hvis man spiser et æble

Spillet skal genstarte, hvis man bider sig selv

Spillet skal genstarte, hvis man rammer væggen

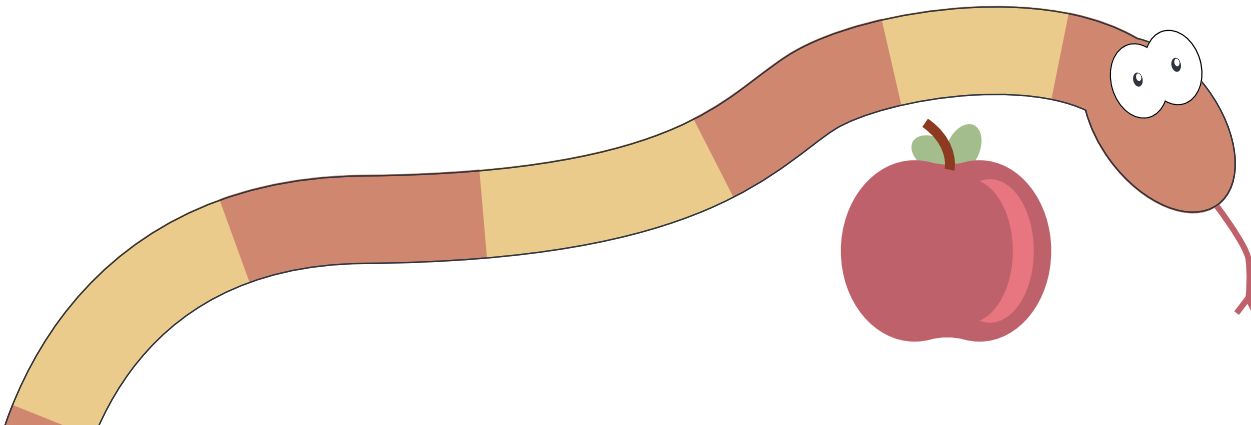


Snake: Trin 0

Opret et nyt Scratch-program med navnet Snake

Vi skal have 3 forskellige "Sprites" i vores Scratch kode. Vores hoved, vores krop og vores æble

I må gerne bruge lidt tid på at tegne nogle forskellige, ellers kan i bare importere nogle ting fra Scratch's bibliotek.



Snake: Trin 1

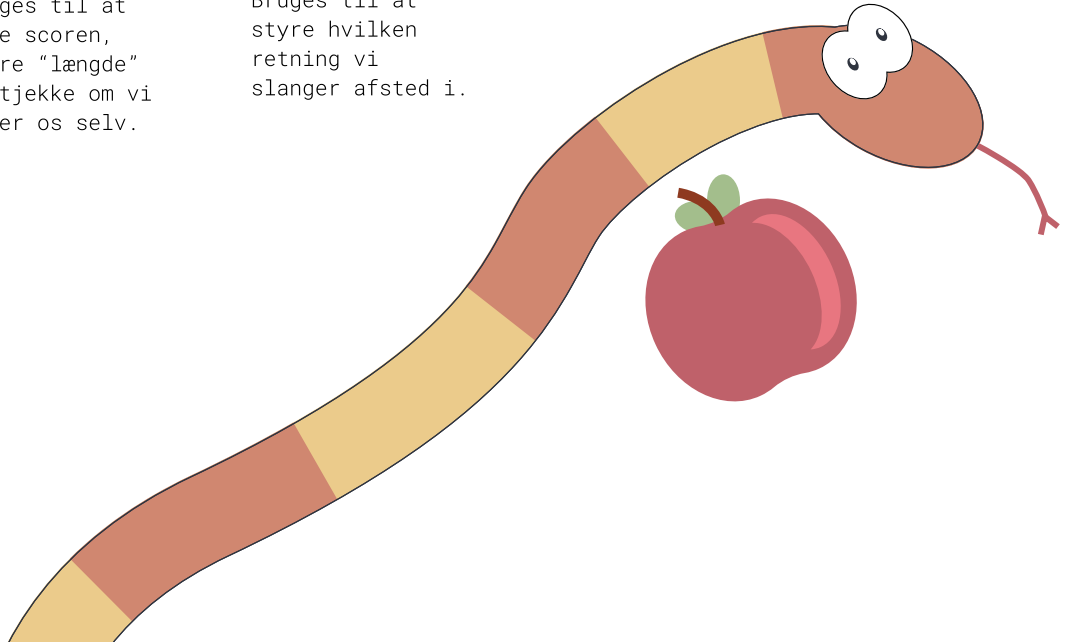
Nu skal vi oprette de nødvendige variable. Når man laver et snake-spil skal man bruge de følgende variable:

“Længde”, “Æbler Spist” og “Retning”.

Bruges til at gøre slangen længere

Bruges til at vise scoren, ændre “længde” og tjekke om vi bider os selv.

Bruges til at styre hvilken retning vi slanger afsted i.



Snake: Trin 2

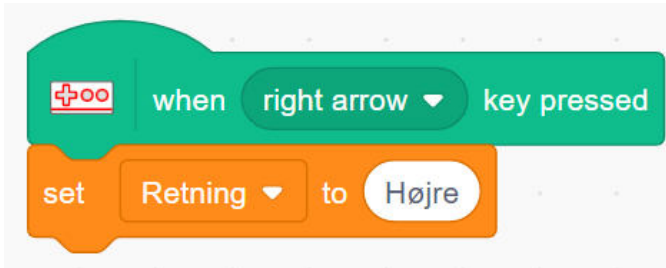
Vi skal skrive de funktioner som styrer vores retninger.

Dette skal ske inden på jeres hoved-kode. Kunne fx være et pindsvin

...



Vi skal oprette 4 styks af den her funktion,
med de fire retninger



Snake: Trin 3

Nu skal vi starte slangen ved et nulpunkt, og så sørge for at den bevæger sig!



The image shows a sequence of four Scratch code blocks. The first block is a yellow 'when clicked' block. The second is a blue 'go to x: 0 y: 0' block. The third is an orange 'set Længde to 0' block. The fourth is an orange 'set Retning to Højre' block. Each block is accompanied by a text explanation on the right, connected by a dashed arrow.

when clicked ←----- Når vi trykker på det grønne flag

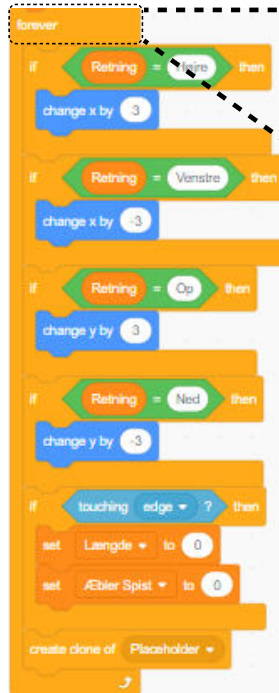
go to x: 0 y: 0 ←----- Sæt 'slangen i midten

set Længde to 0 ←----- Sæt værdien af længden til 0

set Retning to Højre ←----- Den første retning bliver højre

Snake: Trin 4

Nu skal vi integrere alle vores if-statements. Det er lidt vildt noget af det, men vi skal nok komme igennem det hele.



The image shows a vertical stack of Scratch code blocks. At the top is an orange 'forever' loop block. Inside the loop, there are five 'if' blocks. The first four 'if' blocks check for 'Retning = Høje', 'Retning = Venstre', 'Retning = Op', and 'Retning = Ned' respectively, each followed by a 'change x by' or 'change y by' block with a value of -3. The fifth 'if' block checks 'touching edge = ?' and is followed by two 'set' blocks for 'Længde' and 'Æbler Spist', both set to 0. The loop ends with a 'create clone of Placeholder' block. A dashed black line connects the 'forever' block to a large orange box on the right containing the word 'forever' in white text.

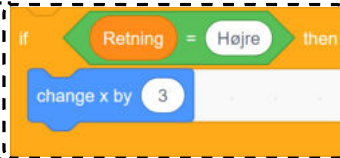
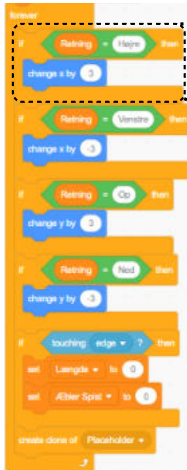
forever

forever

Alle vores if-statements skal være indeholdt i et forever-loop! Forever-loopet kører indtil vi trykker på den røde knap.

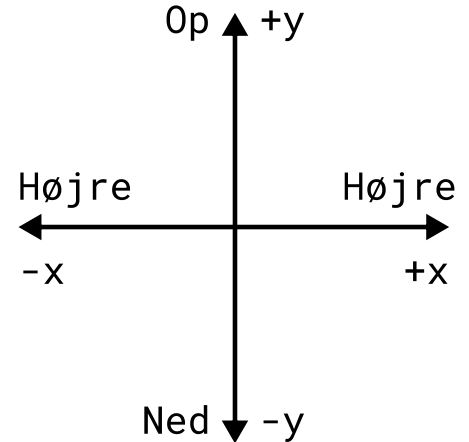
Snake: Trin 5

Det næste trin hænger sammen med de fire bokse med kode, som vi lavede i trin 2.

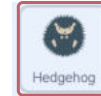


Vores retninger fungerer på følgende måde:

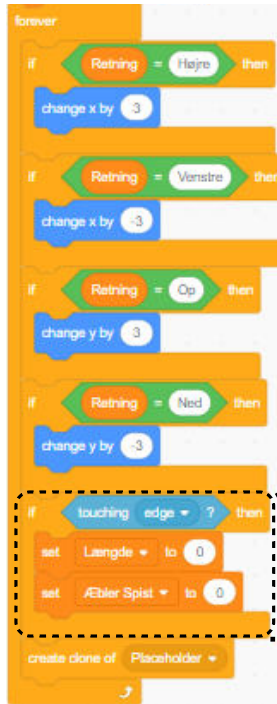
Der skal laves 4 if-statements af den her type med ændringer som passer til retningen



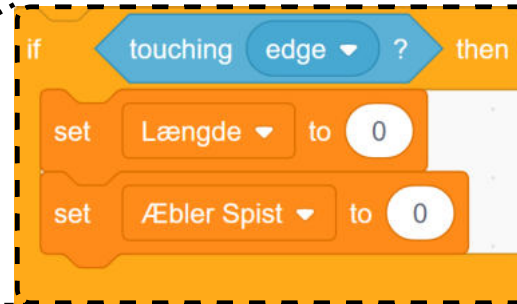
Snake: Trin 6



Nu skal vi lave det if-statement, som gør at vi ikke må ramme kanten.



Det ser ud på følgende måde



Snake: Trin 7



Til sidst sætter vi et stykke kode ind, som giver slangen sin hale. Hvordan det virker giver meget bedre mening, når vi kommer lidt længere ind.

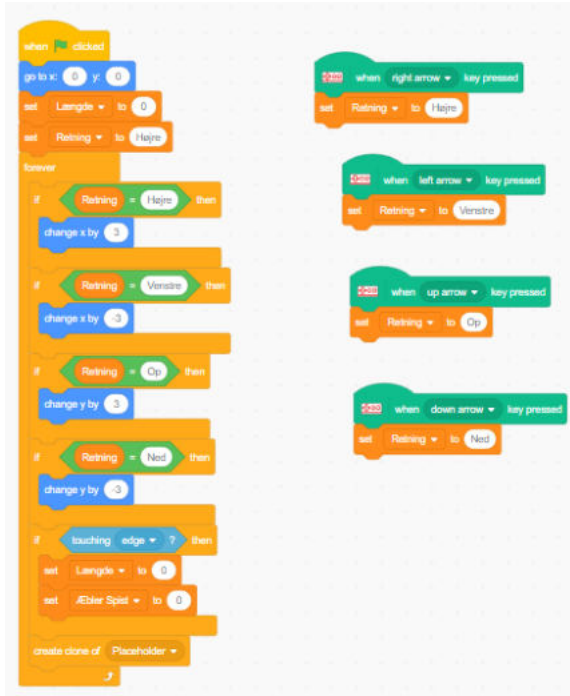
```
forever
  if Retning = Højre then
    change x by -3
  if Retning = Venstre then
    change x by -3
  if Retning = Op then
    change y by -3
  if Retning = Ned then
    change y by -3
  if touching edge = ? then
    set Længde to 0
    set /Efter Spis to 0
  create clone of Placeholder
```

Det er halen vi laver en klon af

```
create clone of Placeholder
```

Nu er hovedet klaret

Hvis jeres kode ser nogenlunde sådan her ud, så er i klar til at gå videre!



HOVEDET ER DONE!

Snake: Trin 8 - Videre til æblet

Koden hos æblet er kortere, og indeholder mange af de samme ting som hovedet på vores slange.

skift til æblet



```
when green flag clicked <----- Når vi trykker på det grønne flag
set Æbler Spist to 0 <----- Antallet af æbler nulstilles når vi starter spillet
go to random position <----- Æblet dukker op et tilfældigt sted på skærmen
forever <----- Igen introducerer vi et forever-loop
  if touching Hedgehog ? then <----- Her sørger vi for at hovedet kan spise æblet
    change Længde by 0.2 <----- Længden bliver større når vi spiser et æble
    go to random position <----- Æblet dukker igen op et tilfældigt sted på skærmen
    change Æbler Spist by 1 <----- Vores score stiger med 1
```

Snake: Trin 9 - Til Sidst, halen

Koden hos æblet er kortere, og indeholder mange af de samme ting som hovedet på vores slange.

skift til halen



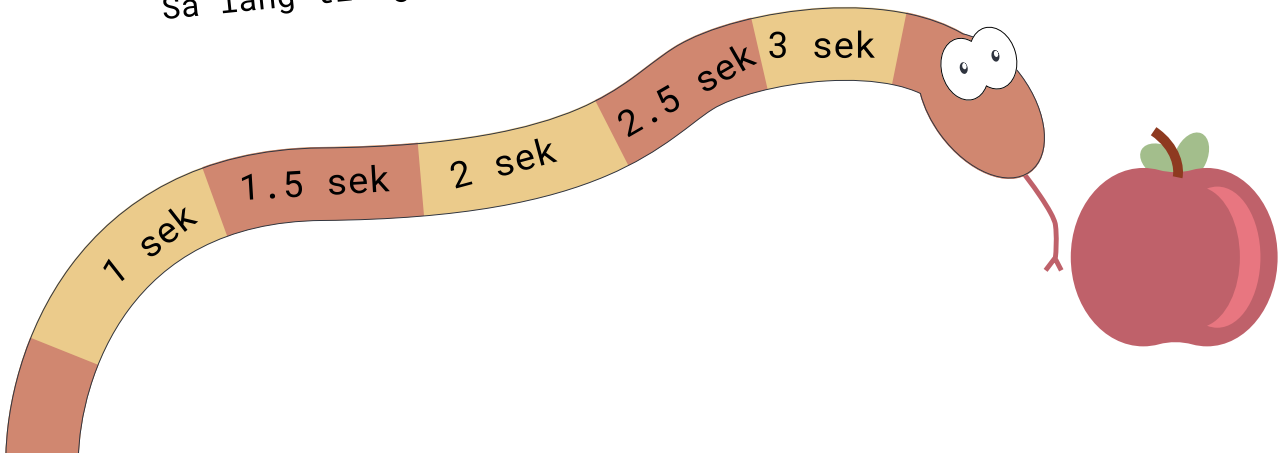
```
when clicked ←----- Når vi trykker på det grønne flag
hide ←----- Halen skal ikke være der før slangen begynder at bevæge sig

when I start as a clone ←----- I trin 7 lavede vi noget kode der skabte en 'klon'. Det
show ←----- er her vi følger op på den kode
go to Hedgehog ←----- Halen bliver oprettet der hvor hovedet er
wait 0.5 seconds ←----- Den her linje koder sørger for at vi ikke får problemer med at
repeat /Ebler Spist ←----- hovedet 'bider' halen med det samme
if not touching Hedgehog then ←----- Vi forstår godt hvis det her virker lidt 'kryptisk' (se næste slide)
wait 0.2 seconds ←----- Hvis hovedet ikke er i kontakt med halen
else ←----- Så venter vi lidt
set Længde to 0 ←----- 'Ellers', altså hvis hovedet faktisk bider halen
set /Ebler Spist to 0 ←----- Sæt længden til 0
delete this clone ←----- Sæt scoren til 0
delete this clone ←----- Slet klonen
```

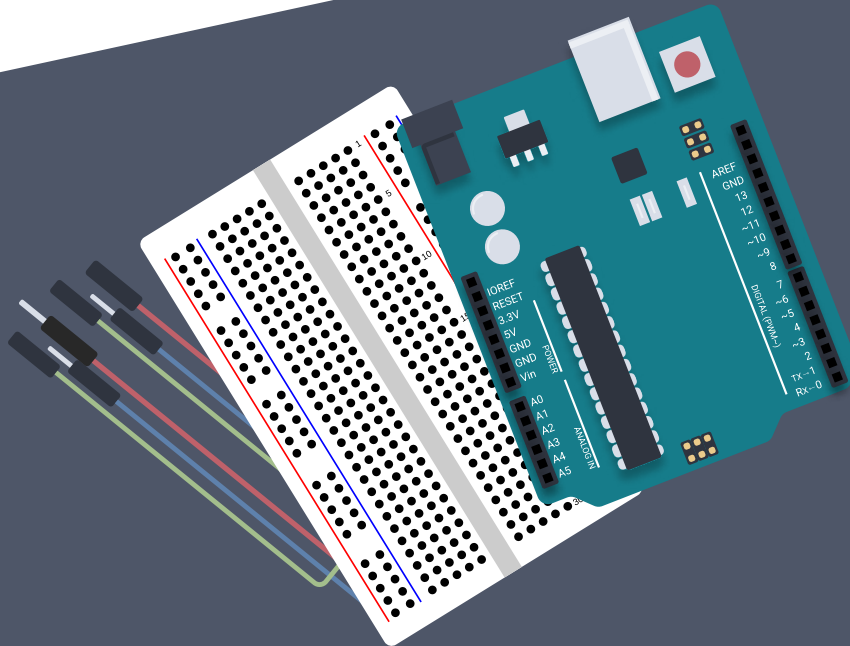
Hvordan virker halen

Halen på vores slange er skabt ved hjælp af en lidt interessant metode (algoritme). Hale-firkanterne bevæger sig faktisk overhovedet ikke. De forsvinder bare igen efter et stykke tid, hvilket giver en illusion om at halen følger med slangen. Hvor lang tid der går inden halen forsvinder er afgjort af antallet af spiste æbler.

Så lang tid går der før firkanten forsvinder



Arduino og digital elektronik

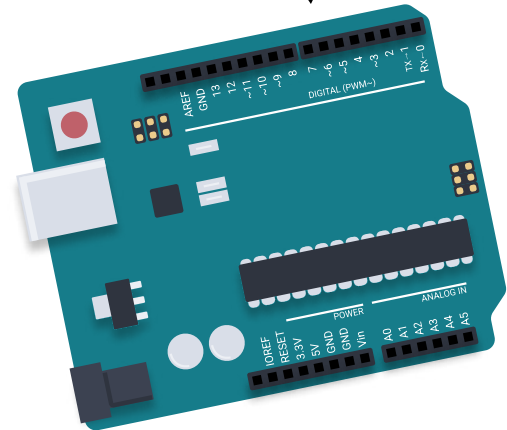
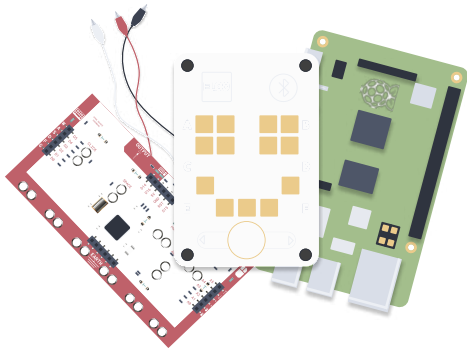


Platforme for digital elektronik

Nu er vi her! Arduino!

Arduino Platformen er den nok mest brugte platform til at udvikle elektroniske (mekatroniske) prototyper.

Vi går videre fra MakeyMakey og kører nu Arduino



Platforme for digital elektronik

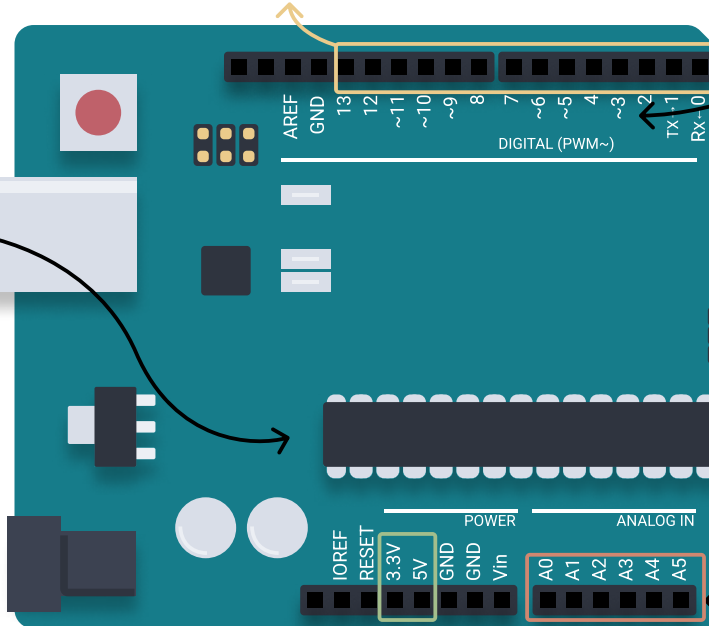
Det her er vores digitale pins. De kan læse strøm eller sende strøm ud

Det her (~) betyder at analoge værdier kan sendes ud

Det her er vores processor

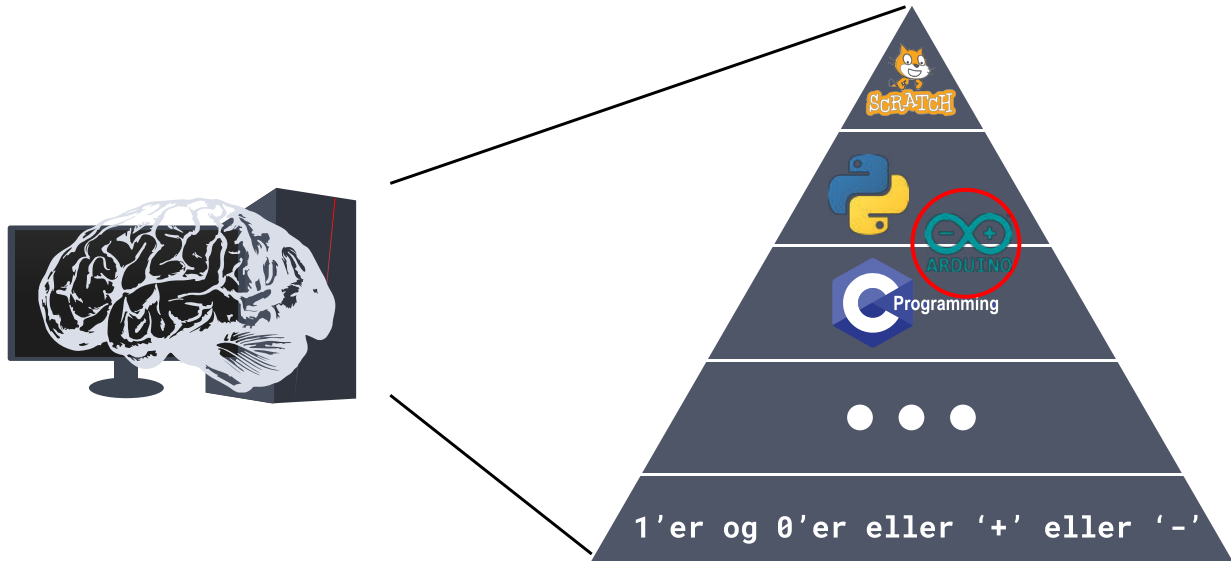
A0 til A5 kan sende og læse "Analog" værdier, altså tal der kan gå fra 0-1024

Fra disse to pins kan man trække 3.3V eller 5V fra



Hvilket sprog er det vi bruger

Det bliver lidt sværere nu, men vi har arbejdet hårdt for at gøre det lidt lettere for jer! Vi kommer til at skrive Arduino sproget, som minder rigtig meget om C.



Arduino Kode Eksempel

kortlink.dk/2g63c

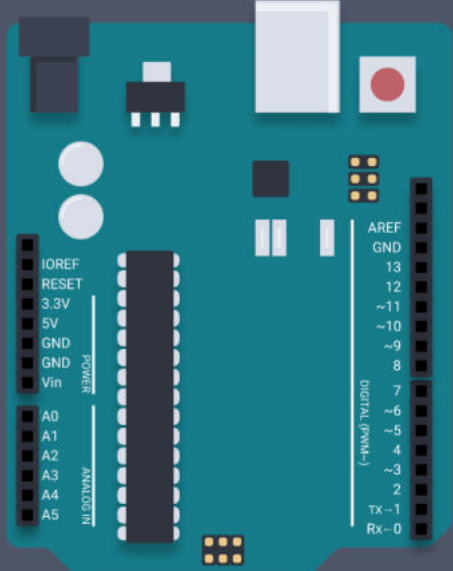
I kommer til at gå forbi den her (tilgå link)

**ARDUINO PROTOTYPE CODING 1:
GENERAL PRINCIPLES**

Coding an Arduino there are two main categories of code:

- Component Code** ⓘ
- Functionality Code** ⓘ

Press one of the categories to continue

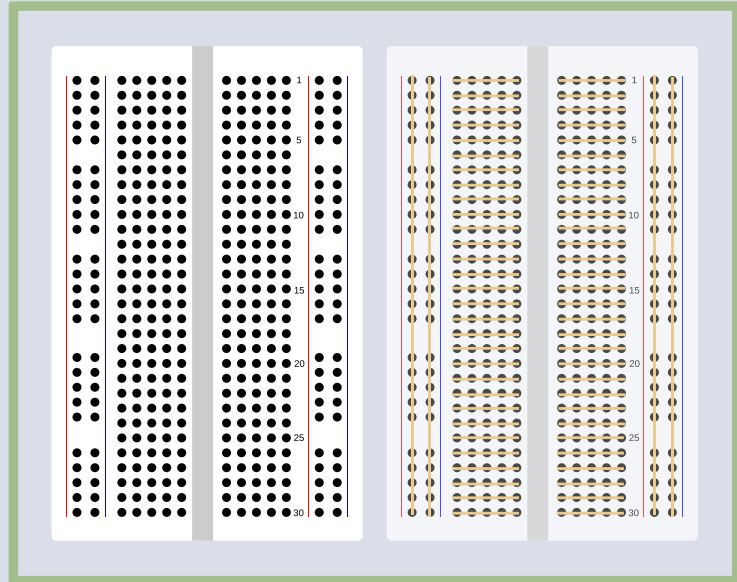
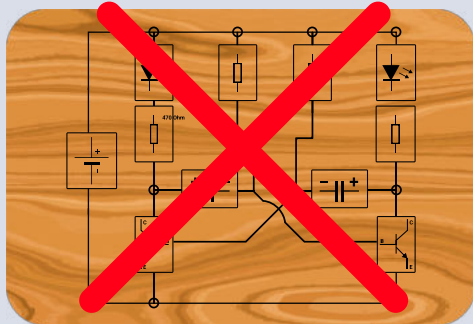
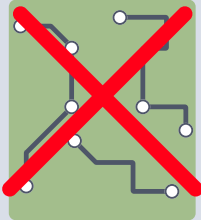


The illustration shows an Arduino Uno R3 board with the following labels:

- POWER:** IOREF, RESET, 3.3V, 5V, GND, GND, Vin
- ANALOG IN:** A0, A1, A2, A3, A4, A5
- DIGITAL (PWM+):** AREF, GND, 13, 12, ~11, ~10, ~9, 8, 7, ~6, ~5, 4, ~3, 2, TX-1, Rx-0

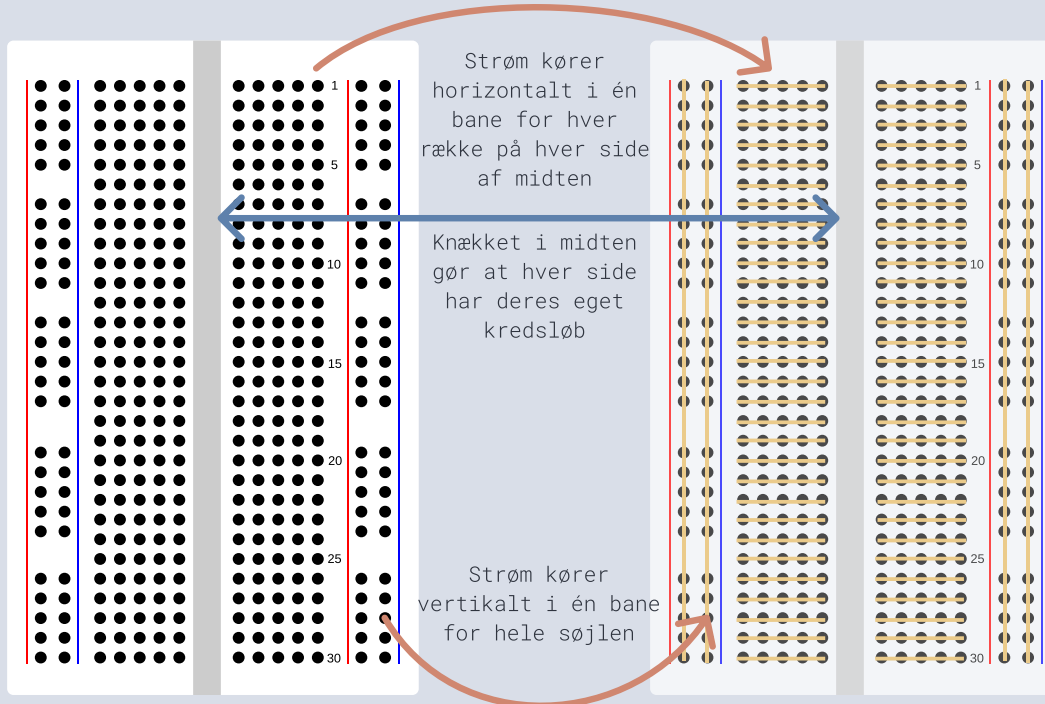
Introducerer: Fumlebrættet!

(På Engelsk "Breadboard", deraf "Brødbræt")



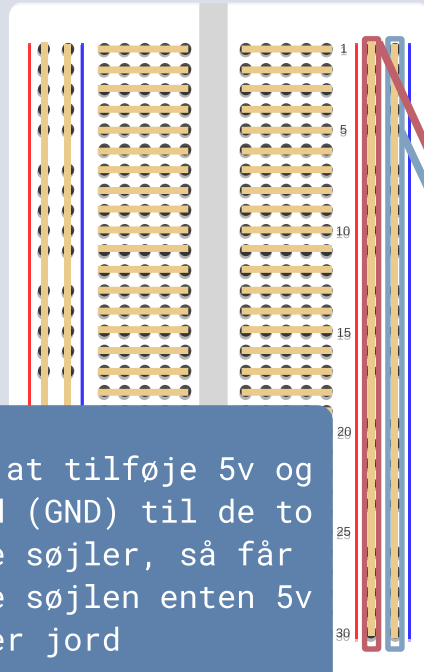
Fumlebrættet

Hvordan fungerer det?

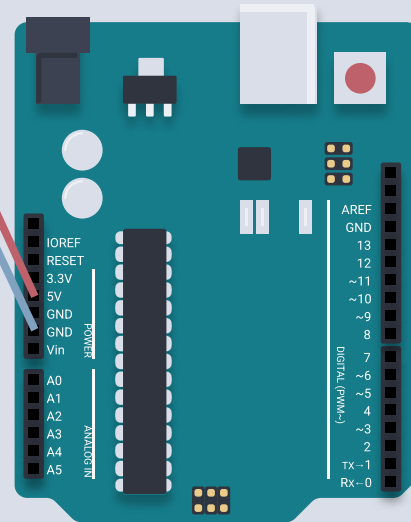


Fumlebrættet

Hvordan fungerer det?



Ved at tilføje 5v og jord (GND) til de to ydre søjler, så får hele søjlen enten 5v eller jord



Introduktion til Arduino

Step 0: Gå ind på hjemmesiden "simpltronics.netlify.app"

Step 1: Gå i gang med Get Started muligheden og lav koden til jeres første Arduino projekt.



Step 2: Tryk på knappen circuit, og begyndt at bygge kredsløbet

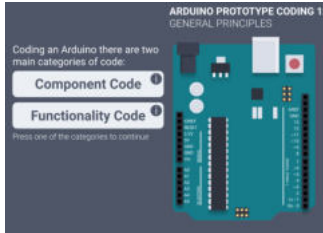
< Gå tilbage

Dit første kredsløb

I denne guide viser vi dig, hvordan du bygger kredsløbet til den kode, du oprettede i kodningsværktøjet.

Hvis du har en Arduino og de komponenter, vi bruger i dette

Step 3: Genopfrisk det vi introducerede til at starte med. Hvad er analog og digital?



< Gå tilbage

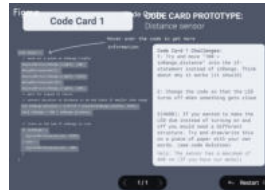
Dit første kredsløb

I denne guide viser vi dig, hvordan du bygger kredsløbet til den kode, du oprettede i kodningsværktøjet.

Hvis du har en Arduino og de komponenter, vi bruger i dette eksempel, er du velkommen til at bygge den [kredsløb](#) og afprøve prototypen. Hvis du ikke har en, er det også helt i orden, du kan bare bruge en online emulator som for eksempel TinkerCAD.

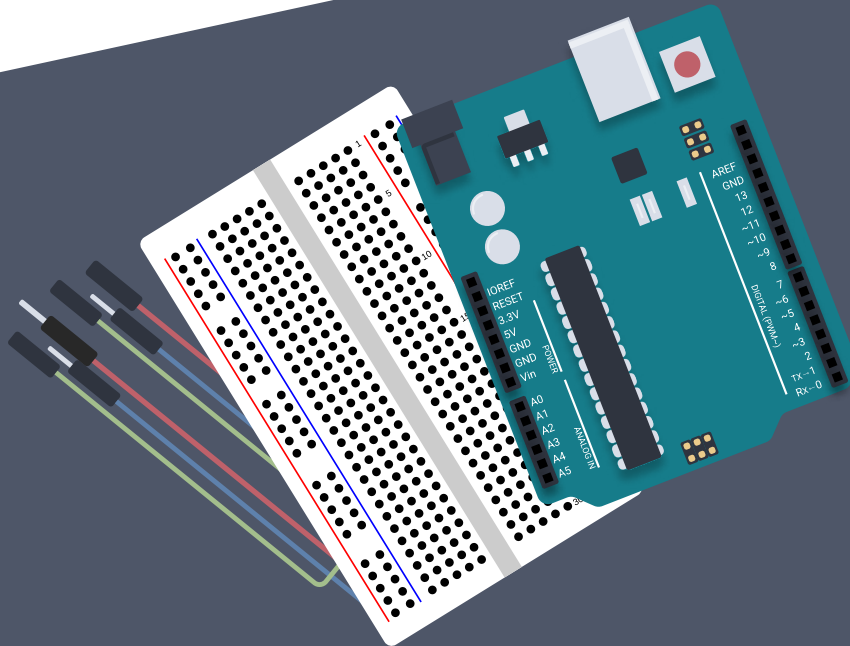
Før du går videre, hvis du ikke er klar over, hvad en [brødbæret](#), anbefaler vi at klikke på [brødbæret](#) hyperlink og tjekke vores beskrivelse af det.

Step 4: Forsøg jer med at løse udfordringerne på det første "kode kort". Det er nyttigt til at begynde at tænke algoritmisk



Arduino og digital elektronik

2

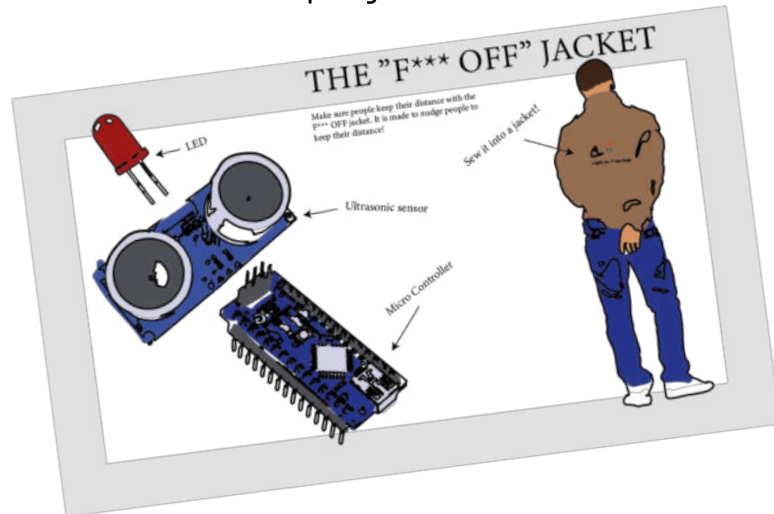


Platforme for digital elektronik

Sidste gang lavede vi den her jakke, som kunne måle om folk var for tæt på, og tænde en LED, hvis de var.

I dag, begynder vi at forberede os på enden (N00000000).

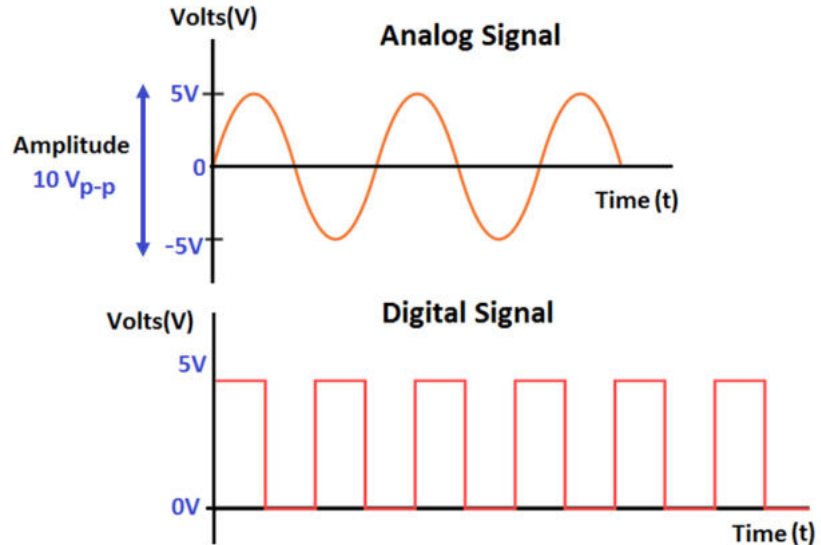
Vi skal arbejde med hvordan man finder fejl, og så skal vi introducere et lidt mere avanceret kode-projekt.



Lidt Repetition

Vi snakkede om digitale og analoge signaler sidst.

Det er sådan her de ser ud, hvis man måler på dem med et volt-meter.

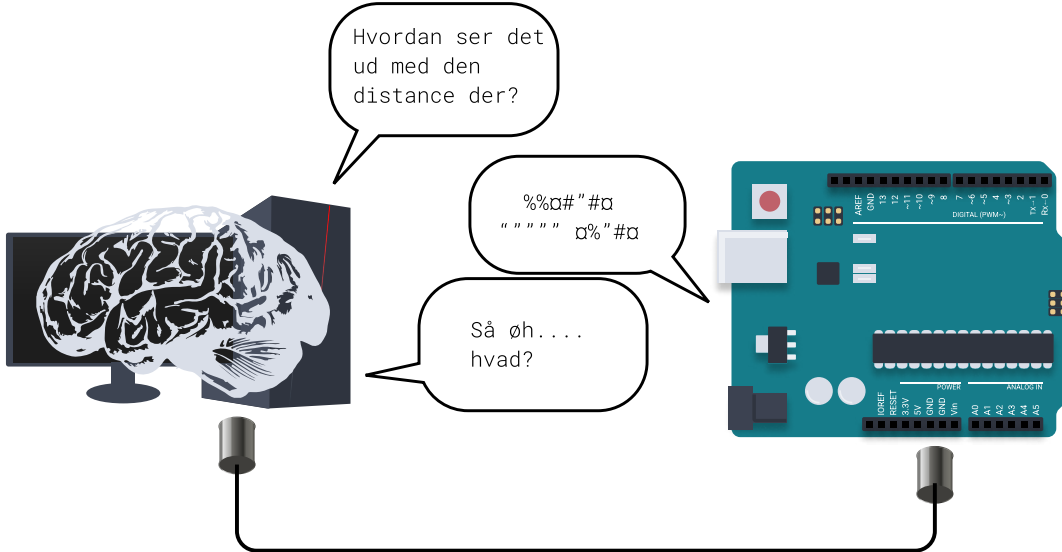


<https://instrumentationtools.com/what-are-analog-and-digital-signals-differences-examples/>

I dag skal vi snakke om 'debugging'

Det er vigtigt at man lærer hvordan man får Arduinoen til at snakke tilbage til computeren

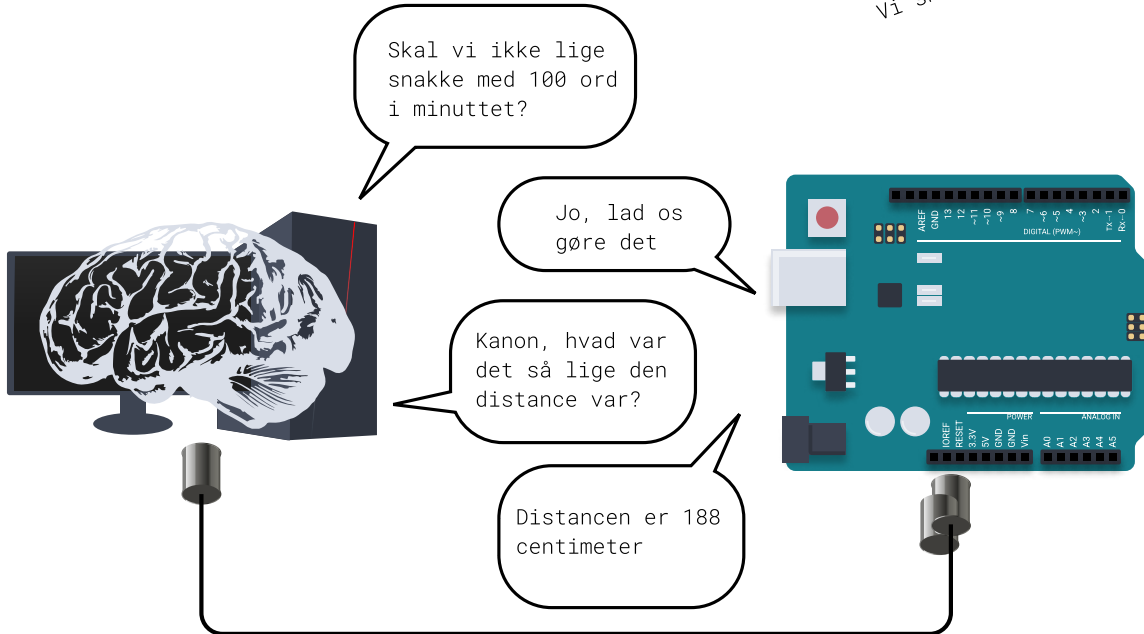
Det betyder "at finde fejl"



Seriekommunikation

Det er vigtigt at man lærer
hvordan man får Arduinoen til at
snakke tilbage til computeren

Vi snakker igennem ledningen



Arduino Kodeeksempel

```
void setup() {  
  // Start samtale mellem Arduino og Computer  
  Serial.begin(9600);  
}
```

Arduino kode skal have en void Setup() function. Setup-delen køres igennem en gang, og bliver derfor brugt til kode, som ikke behøver blive gentaget

← I arduino-kode bruger vi rigtig mange semi-kolon

Vi snakker med en hastighed på 9600 bits pr. second

Vi skal "begynde" at snakke sammen igennem ledningen

Der er nogen som har skrevet et bibliotek med navnet "Serial", som har den kode vi skal bruge når man laver seriekommunikation

Arduino Kodeeksempel

void loop() skal også være en del af en arduino-kode, ligesom void setup(). Her bliver koden dog gentaget igen og igen

Citationstegnene indikerer at 'vi' skriver noget

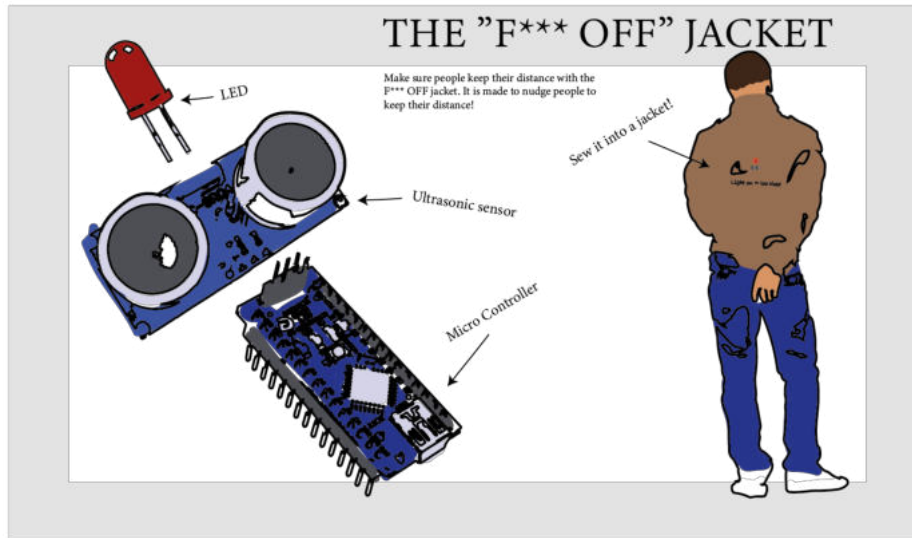
```
void loop() {  
  Serial.print("Distancen er:");  
  Serial.println(distancen);  
}
```

Distancen er navnet på en variabel, som bliver udregnet i koden

ln gør at vi skifter linje efter vi har printet.

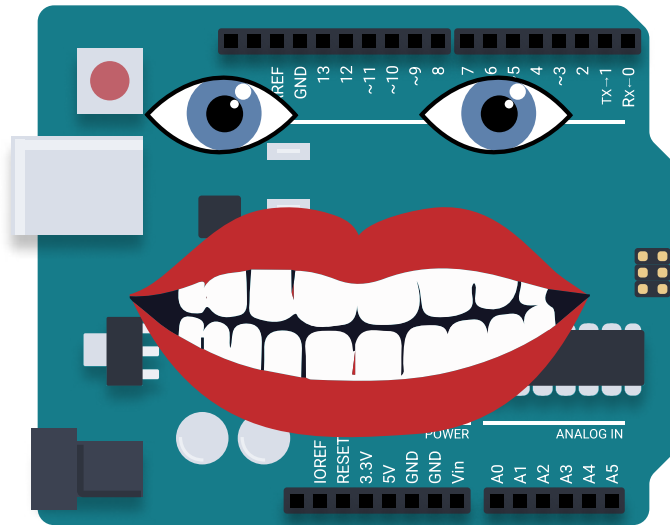
Forsøg at skrive det ind

Forsøg at lave jakke-eksemplet igen, og den her gang skal i forsøge at introducere seriekommunikation med Arduinoen.



Hvorfor???

Seriekommunikation er jeres bedste værktøj til at finde ud af, hvorfor jeres kode ikke fungerer som i gerne vil have det. Det giver nemlig Arduinoen mulighed for at fortælle jer hvad den ser



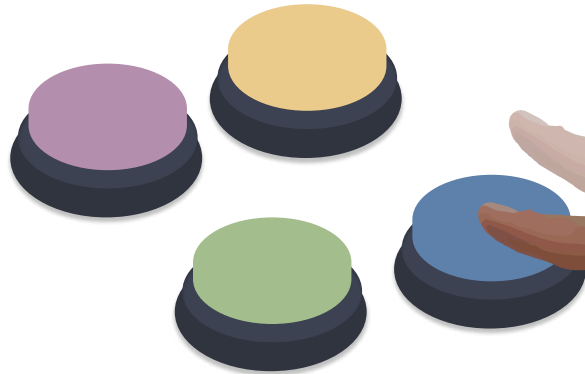
Så var der sgu lige 5V på pin 7. Er det ikke vildt?! Det skete jo også for 3 sekunder siden

Forhåbningen for hvad vi når at lave i dag:

Idéen er at vi skal kunne lave et "Buzz" quiz spil med tre knapper, tre LED'er og to højttalere (piezo buzzer).

Start med at få de enkelte koder til at fungere og så prøv at få dem samlet én efter én!

Prøv at tænke: Hvordan ville jeg gøre det i scratch?



Vejen mod Quiz Spillet!

Step 0: Gå ind på hjemmesiden "simpltronics.netlify.app"

Step 1: Gå til "Vejledning" siden

Step 2: Løs projektet "Trykknop med LED" (husk at gem denne kode!)

Step 3: Efter i er færdige med det, prøv at lav projektet "Enkel højttaleroopsætning", også inde på "Vejledning" siden. Gem også koden herfra!

Step 4: Prøv nu at kombinér LED og knap koden sammen med jeres højttaler således at når i trykker på knappen tænder LED'en OG højttaleren spiller en lyd!

Step 5: Quiz Spil?
Spiller 1: 0 Spiller 2: 1

Det er alt for i dag!

Lille gennemgang af koden:

11:24:46

Trykknop med LED

Introduktion

I denne guide lærer du at kontrollere en digital udgang (LED) og en digital indgang (knap), der tænder LED'en, når knappen trykkes ned.

Inden du fortsætter denne vejledning, anbefales det, at du har læst - eller har grundlæggende viden om [trykknop](#) og [LED'er](#).

```
const int buttonPin = 2; // vores pin der styrer knappen
const int LEDPin = 4; // vores pin der styrer LED'en
```

Hey jeg er en knap!
og jeg er en LED!

```
int buttonState = 0; //Denne variabel vil fortælle om knappen er tændt eller slukket
// a variable is assigned to set the status of the LED
int LEDState = 0;
```

```
void setup() {
  // Vi sætter først vores knap pin til at være noget vi gerne vil modtage
  //i.e at vi gerne vil læse knappens signal
  pinMode(buttonPin, INPUT);
  // Modsat, vi vil gerne have at vores LED pin skal sende noget ud, dvs. enten
  // at slukke eller tænde for LED'en
  pinMode(LEDPin, OUTPUT);
}
```

Jeg er en knap der
gerne vil læses!

Jeg er en LED der
gerne vil tændes!

```
void loop() {
  // Vi vil gerne "læse" hele tiden om knappen er tændt eller slukket
  buttonState = digitalRead(buttonPin);
```

Jeg er en knap og
her er min status!

```
  // Hvis knappen er trykket ned, så skal det følgende ske
  if (buttonState == HIGH) {
    // Hvis knappen er trykket ned, så skal lyset tændes
    digitalWrite(LEDPin, HIGH);
  } else { //ellers:
    // Sluk LED'en hvis knappen ikke er trykket ned.
    digitalWrite(LEDPin, LOW);
  }
}
```

Hvis knap er tændt,
så tænder vi også
LED'en, ellers skal
LED'en være slukket

```
  // Til sidst tilføjer vi en lille pause for at give vores Arduino en chance
  //for at læse om knappen er trykket ned igen
  delay(50);
}
```


Lille gennemgang af koden:

Hey jeg er en buzzer!

1 af 11 sider

Enkel højttalersætning

Introduktion

Højttalere kan bruges til mange ting, for eksempel at spille melodier, kommunikere til en bruger af et produkt eller endda give en advarsel om, at der er noget galt.

I denne guide lærer du, hvordan du tilslutter en højttaler til din mikrocontroller og får den til at afgive en støj ved hjælp af en højttaler eller aktiv buzzer.

Inden du fortsætter denne vejledning, anbefales det, at du har læst - eller har grundlæggende viden om [Arduino](#).

```
const int PiezoPin = 2; //Vi sætter vores PiezoPin til at være nr. 2
```

```
void setup() {  
  // Der er ikke behov for et setup i denne guide :)  
}
```

```
void loop() {  
  // Laver en lyd der er 2000-, 3000- og 4000 hertz høj i et halvt (500ms) sekund  
  // make a pitch sound at 2000-, 3000- and 4000 hertz for half a second each  
  tone(PiezoPin, 2000, 500); //Laver tonen  
  delay(500); // Sætter en pause for tonen  
  tone(PiezoPin, 3000, 500);  
  delay(500);  
  tone(PiezoPin, 4000, 500);  
  delay(500);  
}
```

Jeg vil gerne spille
en tone på min
buzzer

Jeg venter i et halv
sekund

Så spiller jeg en ny
tone.

Osv.

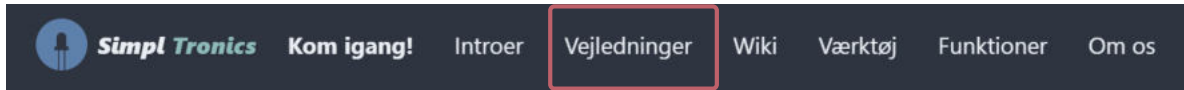
Læg mærke til tonen her!
Den ændrer sig hver gang

Høj frekvens (hertz) = høj
piv tone, mens lav frekvens
er en dyb/mørk tone

Vejen mod Quiz Spillet!

Step 0: Gå ind på hjemmesiden "simpltrronics.netlify.app"

Step 1: Gå til "Vejledninger" siden



Step 3: Efter i er færdige med dét, prøv at lav projektet "Enkel højtaleropsætning", også inde på "Vejledninger" siden. Gem også koden herfra!

Step 2: Løs projektet "Trykknop med LED" (husk at gem denne kode!)



Step 3: Efter i er færdige med dét, prøv at lav projektet “Enkel højttaleropsætning”, også inde på “Vejledninger” siden. Gem også koden herfra!

[← Gå tilbage](#)

Enkel højttaleropsætning

Introduktion

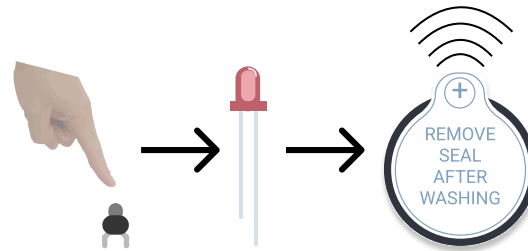
Højttalere kan bruges til mange ting, for eksempel at spille melodier, kommunikere til en bruger af et produkt eller endda give en advarsel om, at der er noget galt.

I denne guide lærer du, hvordan du tilslutter en højttaler til din mikrocontroller og får den til at afspille en støj ved hjælp af en højttaler eller aktiv buzzer.

Inden du fortsætter denne vejledning, anbefales det, at du har læst - eller har grundlæggende viden om [højttalere](#).



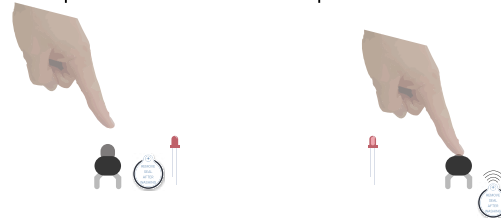
Step 4: Prøv nu at kombinér LED og knap koden sammen med jeres højttaler således at når i trykker på knappen tænder LED'en OG højttaleren spiller en lyd!



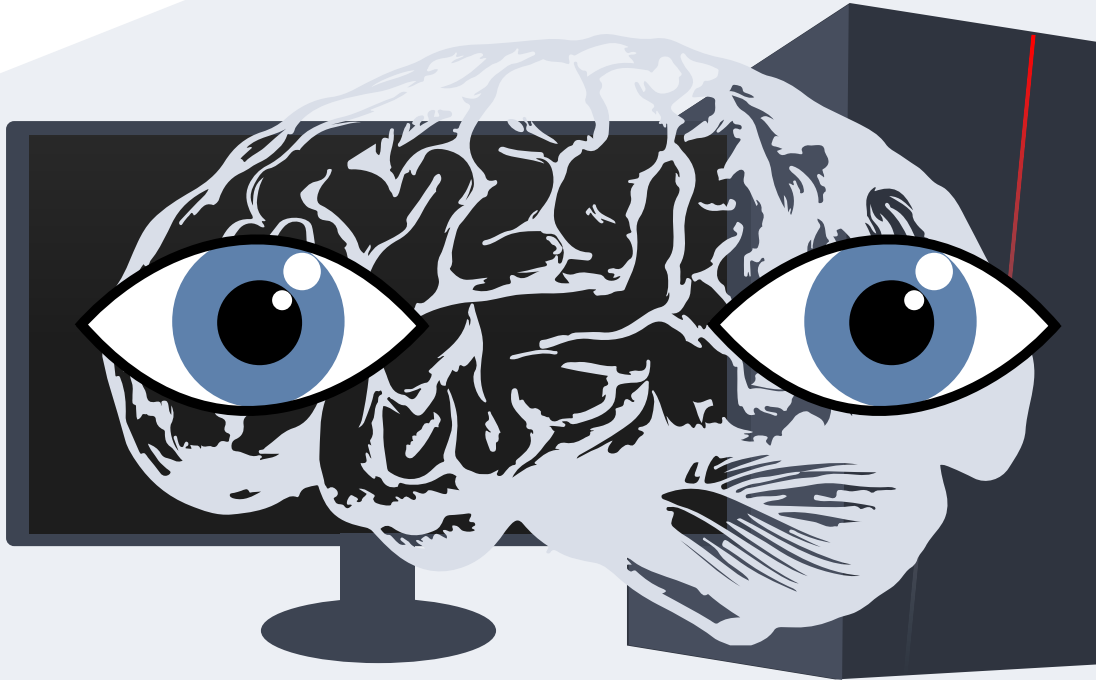
Step 5: Quiz Spil?

Spiller 1: 0

Spiller 2: 1

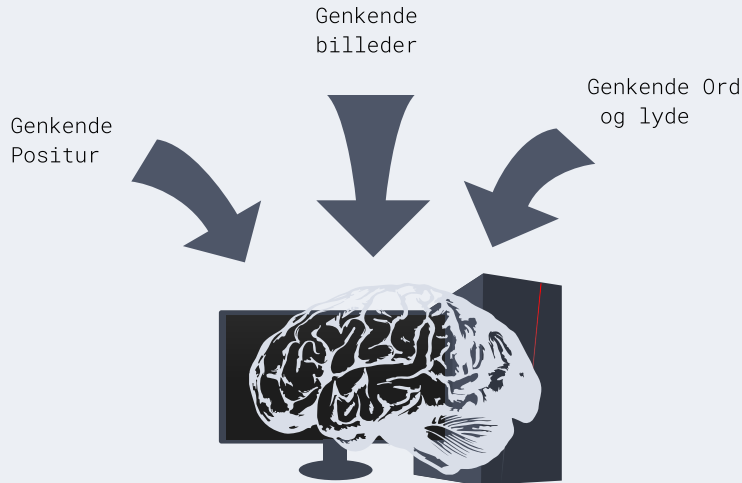


Machine Learning



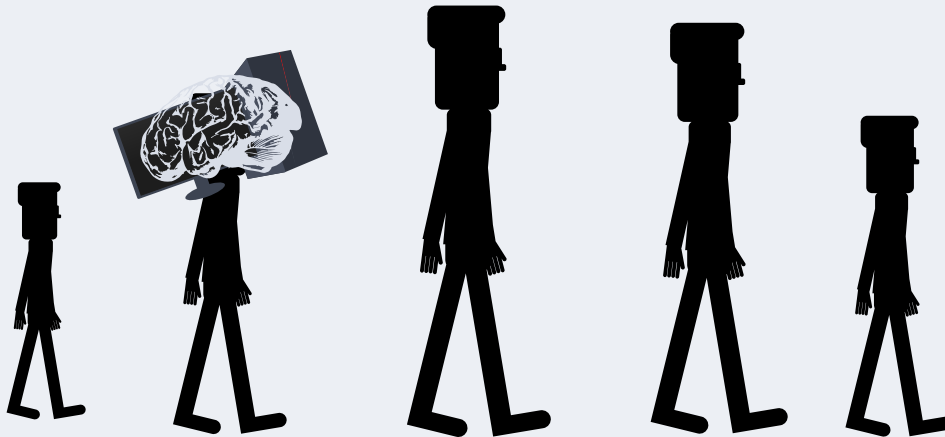
Machine Learning

Machine learning kan være meget kompliceret og blive brugt til utrolig mange ting. Vi kommer til at bruge det til at give vores computer evnen til at genkende positur, genkende billeder og genkende ord og lyde. Når vores computer kan de her ting, kan vi lave sjove koncepter!



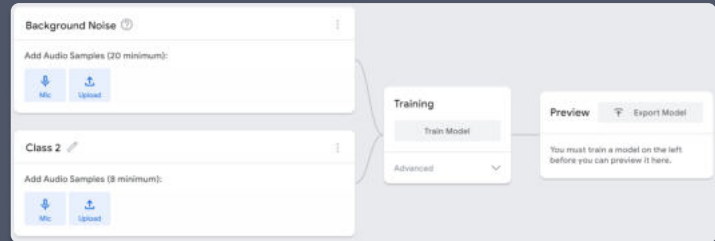
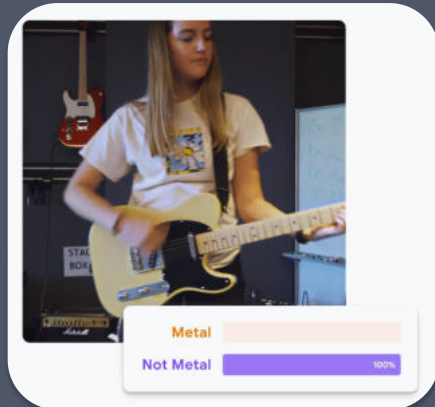
Machine Learning

Computere bliver 'klogere' når vi lærer dem ting. Det er det i skal tænke machine learning som. I fremtiden kan vi måske lærer computere meget komplicerede ting!



Teachable Machine

Til at hjælpe os bruger vi et program google har lavet der hedder teachable Machine. Programmet hjælper os med at lave modeller som giver computere mulighed for at forstå billeder, lyde og positur.




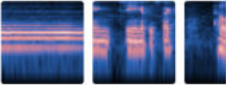



Image Project

Teach based on images, from files or your webcam.



Audio Project

Teach based on one-second-long sounds, from files or your microphone.

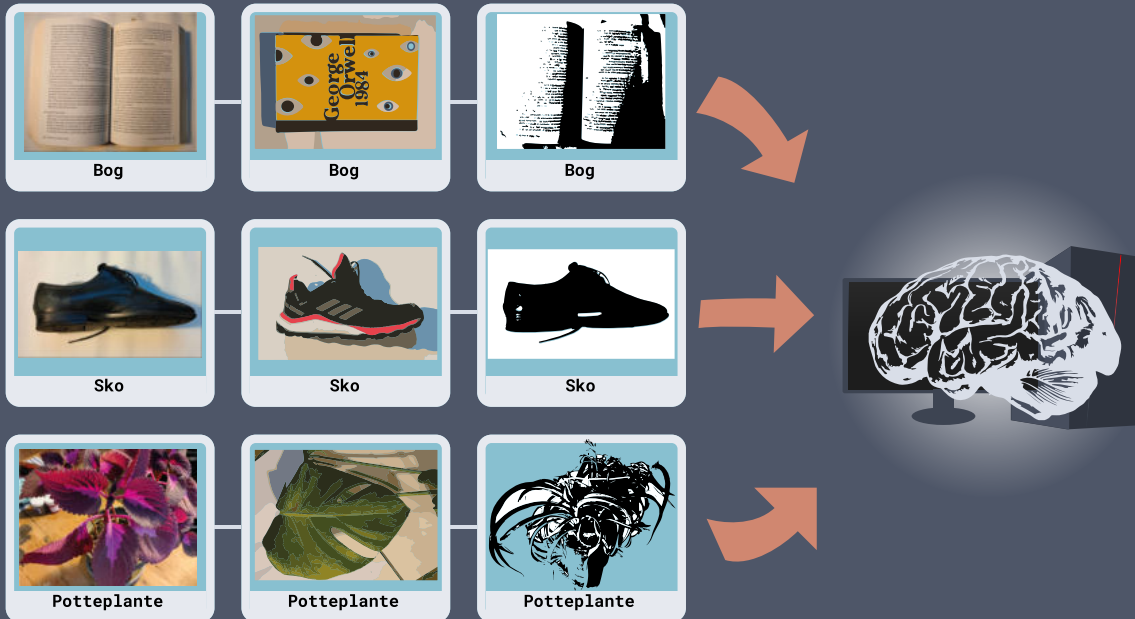


Pose Project

Teach based on images, from files or your webcam.

Teachable Machine

De modeller vi skaber fungerer alle på samme måde - så lad os tage et eksempel. Nemlig at lære billeder at kende. I det her eksempel lærer vi computeren hvad en bog, en sko, og en potteplante er. Vi giver computeren 3 billeder af hvert ting, så den kan forstå hvordan den skal genkende dem.



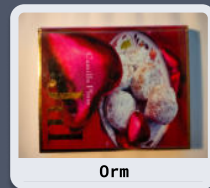
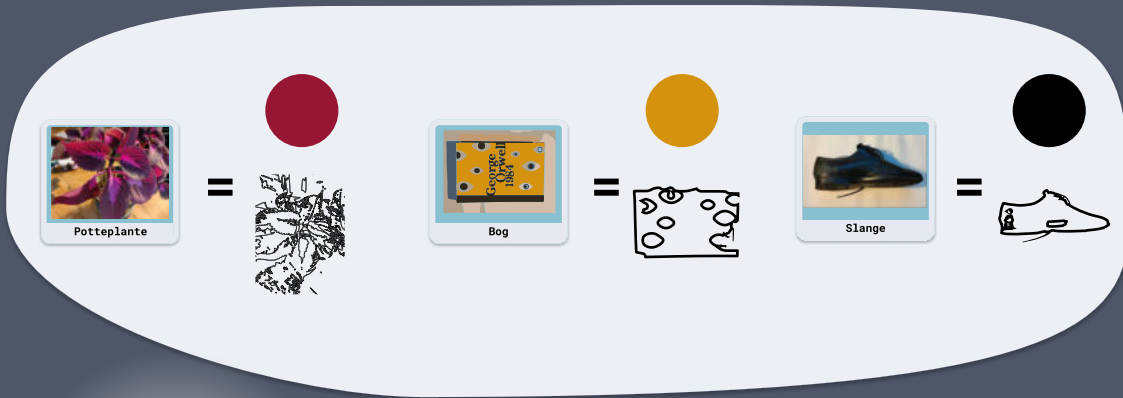
Teachable Machine

Der kan dog godt opstå fejl, hvis nogle billeder falder imellem to kategorier. Et billede af en rød bog, ville fx forvirre den her computer, da den ville tænke "Planten er rød eller grøn, og derfor er det nødt til at være en potteplante". Hvis vi havde vist den at bøger også kan være røde, havde den haft en bedre chance.



Teachable Machine

Opsummeret, fungerer det altså på den her måde. Computeren kigger efter former, farver, osv. Så tænk over det i jeres eksempler.

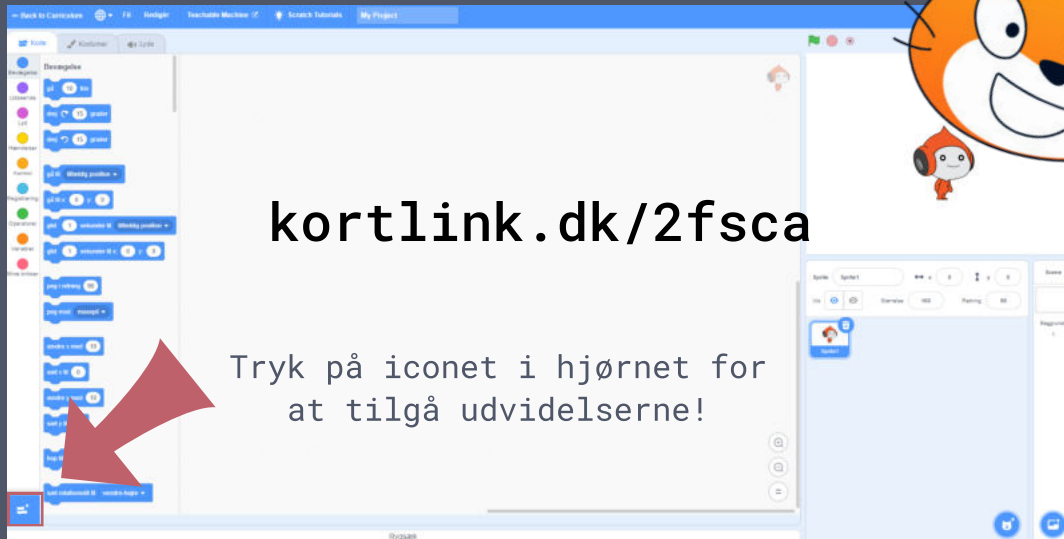


?

?


Teachable Machine

På det her link kan i finde en udgave af scratch, hvor man kan bruge teachable machine!



kortlink.dk/2fsca

Tryk på iconet i hjørnet for at tilgå udvidelserne!



Teachable Machine

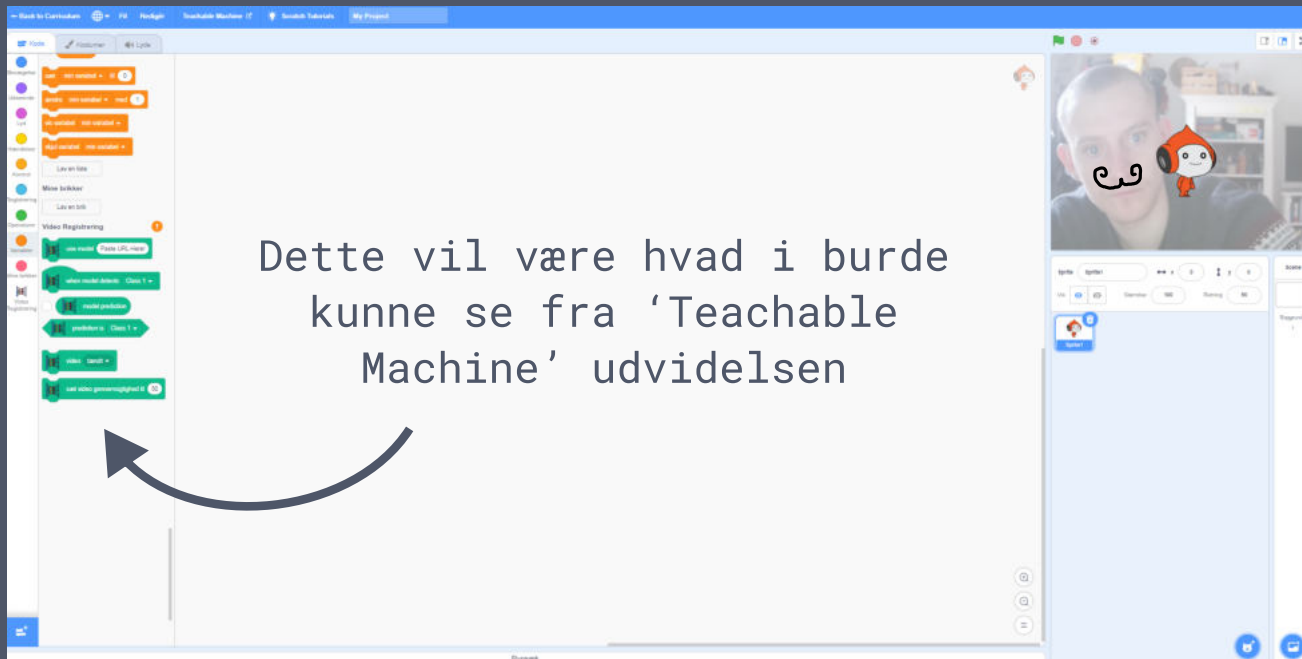


The screenshot shows the 'Vejlø udvidelse' (Browse Extensions) window in Scratch. The interface includes a 'Tilbage' (Back) button and a search bar. A grid of extension cards is displayed, each with an icon, title, description, and compatibility information. A large red arrow points to the 'Teachable Machine' extension card in the top-left corner.

Extension Name	Description	Compatible With
Teachable Machine	Use Google Teachable Machine models in your Scratch project.	Scratch 2.0
Hand Sensing	Sense hand movement with the camera.	Scratch 2.0
Face Sensing	Sense face movement with the camera.	Scratch 2.0
Body Sensing	Sense body position with the camera.	Scratch 2.0
Musik	Spil på instrumenter og trommer	Scratch 2.0
Pen	Tegn med dine sprites.	Scratch 2.0
Video Registrering	Registrerer bevægelse med kameraet	Scratch 2.0
Tekst til tale	Få dine projekter til at tale.	Scratch 2.0, Amazon Web Services
Oversæt tekst til talte sprog	Oversæt tekst til talte sprog.	Scratch 2.0, Google
microbit	Forbind dine projekter med den fysiske verden.	Scratch 2.0, micro:bit
LEGO MINDSTORMS EV3	Byg interaktive robotter og mere	Scratch 2.0, LEGO
LEGO BOOST	Gør dine robotter levende.	Scratch 2.0, LEGO
LEGO Education WeDo 2.0	Byg med motorer og sensorer	Scratch 2.0, LEGO
Go Direct Force & Acceleration	Registrér skub, træk, bevægelse og drejning rundt.	Scratch 2.0, Vernier

Find og vælg 'Teachable Machine' under udvidelserne!

Teachable Machine



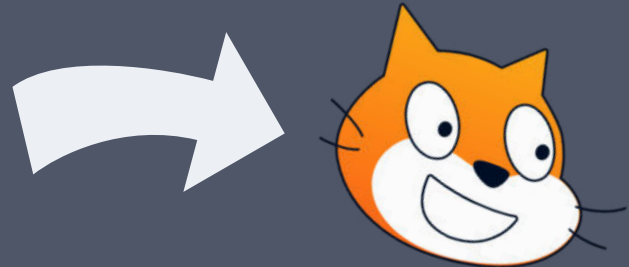
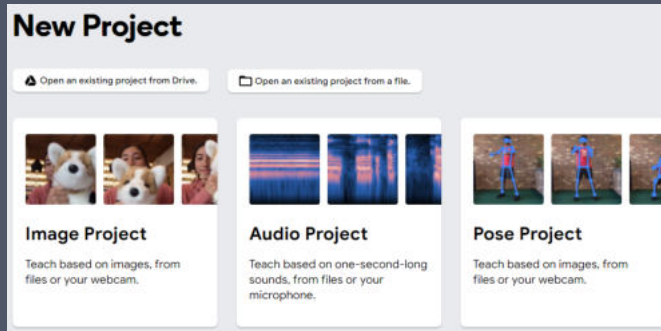
The image shows a screenshot of the Teachable Machine web interface. On the left, there is a flowchart representing a neural network with various nodes and connections. The central area is a large white space. On the right, there is a video chat window showing a person's face with a small red robot icon overlaid. The text 'Dette vil være hvad i burde kunne se fra 'Teachable Machine' udvidelsen' is overlaid on the central area, with a curved arrow pointing from the text to the flowchart.

Dette vil være hvad i burde kunne se fra 'Teachable Machine' udvidelsen

Teachable Machine

Det næste der skal ske er at i importerer en model fra Teachable Machine! Som det første, tilgå Teachable Machine's hjemmeside.

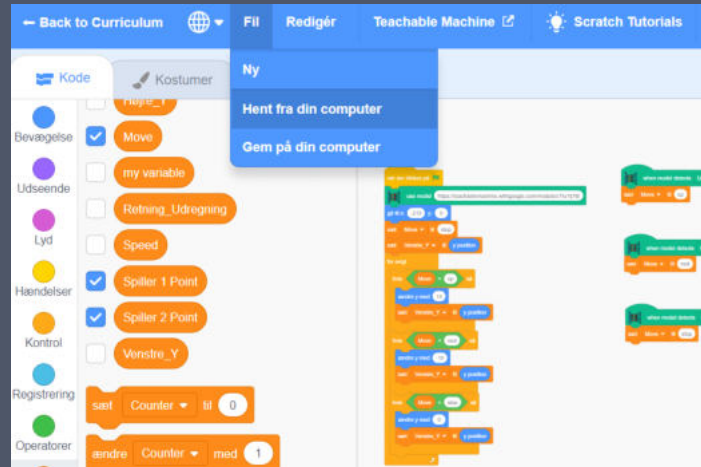
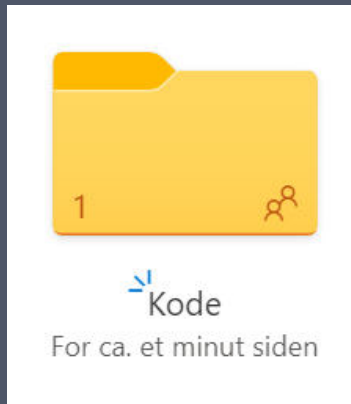
<https://teachablemachine.withgoogle.com/>



Teachable Machine

kortlink.dk/2g63h


I kan downloade færdiglavet Scratch kode gennem kortlinket ovenpå.




Teachable Machine

Nu kan i lave en model! I kan selv vælge hvilken type model i vil lave, så længe i får lavet tre forskellige måder at programmet kan registrerer jer, sådan at vi kan få vores Scratch spil til at gå 'Op', 'Ned' og 'Stoppe'.

New Project

 Open an existing project from Drive.

 Open an existing project from a file.

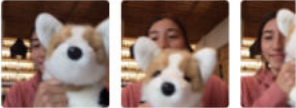
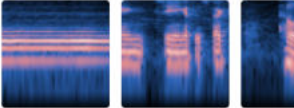



Image Project

Teach based on images, from files or your webcam.



Audio Project

Teach based on one-second-long sounds, from files or your microphone.

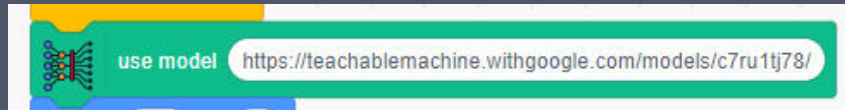
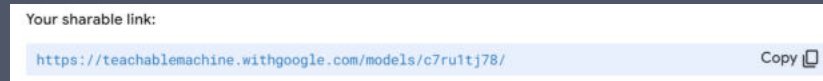


Pose Project

Teach based on images, from files or your webcam.

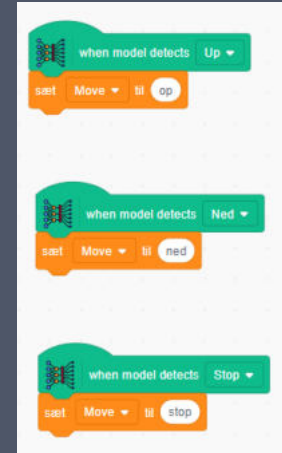
Teachable Machine

Når i har lavet jeres model kan i 'Uploade' og eksporterer den, hvor i får et delbart link.



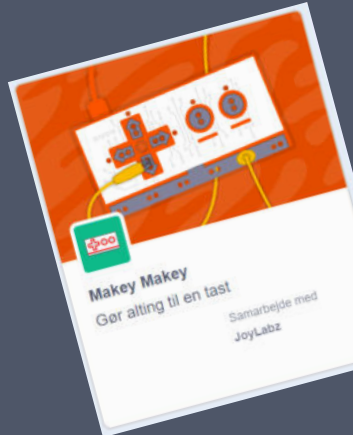
Dette link kan således kopieres og indsættes i Scratch's 'use model' blokken.

Herfra kan styringen af Scratch spillet nu styres gennem jeres Teachable Machine model!



Teachable Machine

Udover Teachable Machine er der mange flere udvidelser til Scratch!

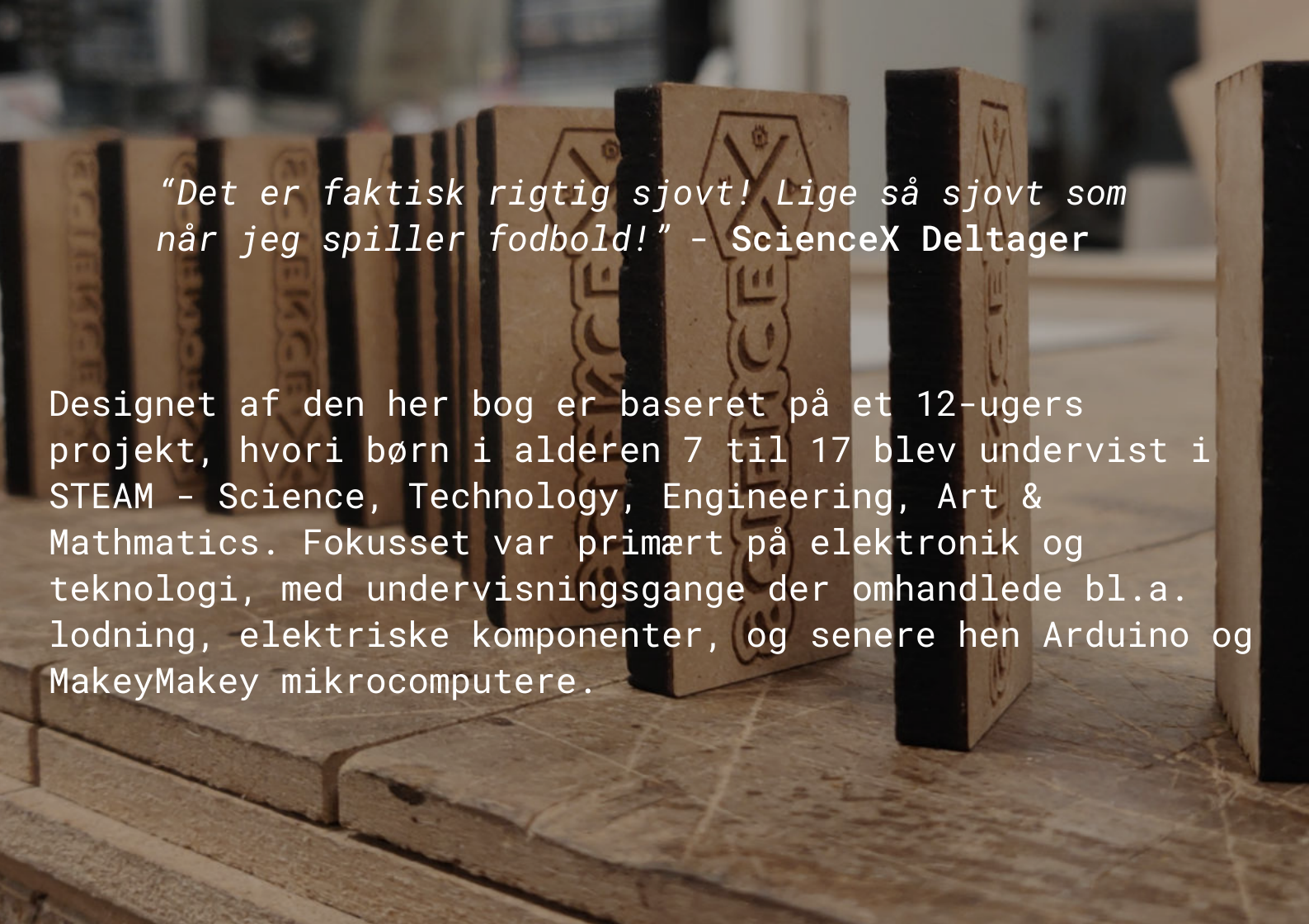


Det var alt, folkens!



Forfattere: Anton Hermann & Frederik Nielsen

Tak for et spændende forløb!



“Det er faktisk rigtig sjovt! Lige så sjovt som når jeg spiller fodbold!” - ScienceX Deltager

Designet af den her bog er baseret på et 12-ugers projekt, hvori børn i alderen 7 til 17 blev undervist i STEAM - Science, Technology, Engineering, Art & Mathematics. Fokuset var primært på elektronik og teknologi, med undervisningsgange der omhandlede bl.a. lodning, elektriske komponenter, og senere hen Arduino og MakeyMakey mikrocomputere.