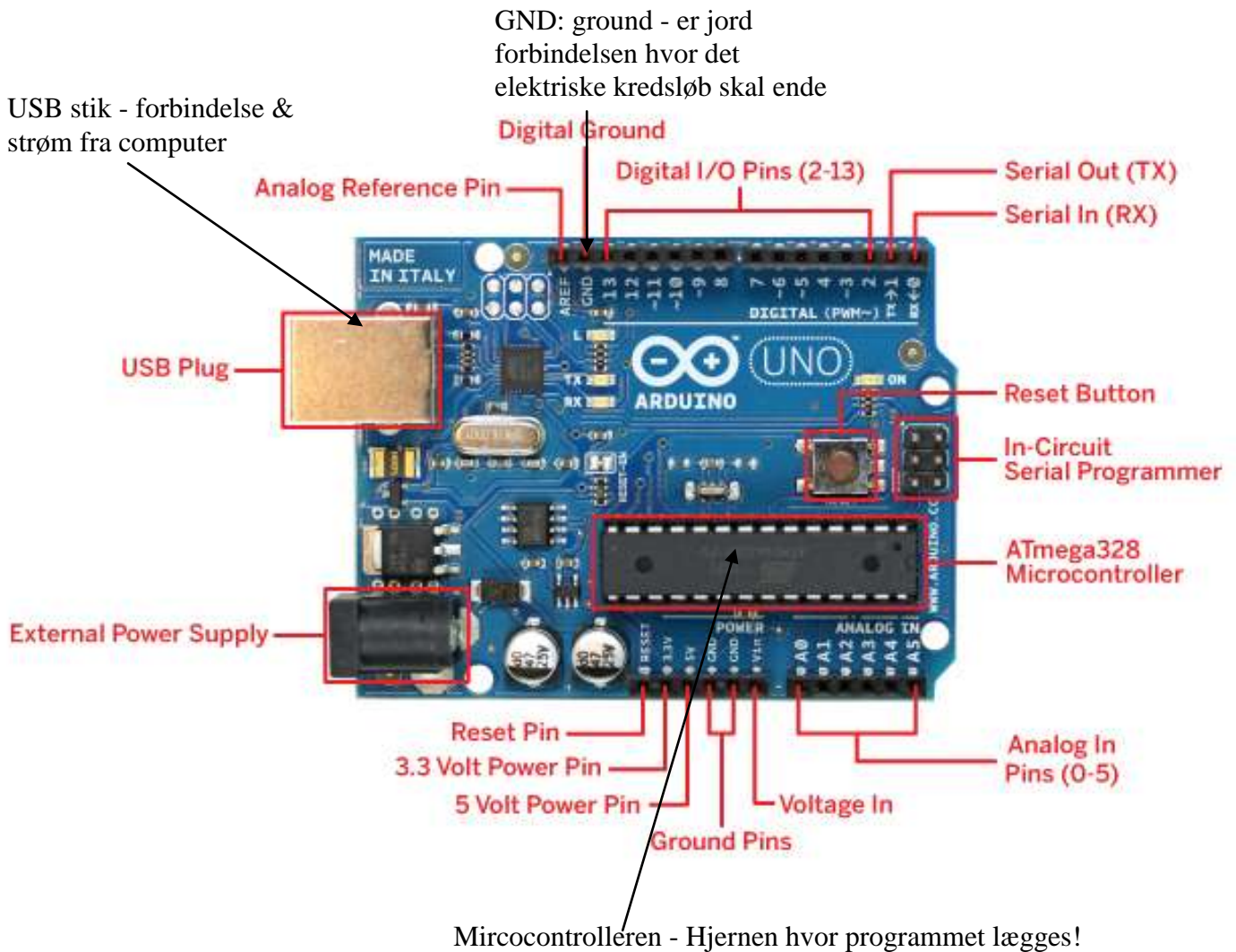


# Arduino kursus lektion 1:



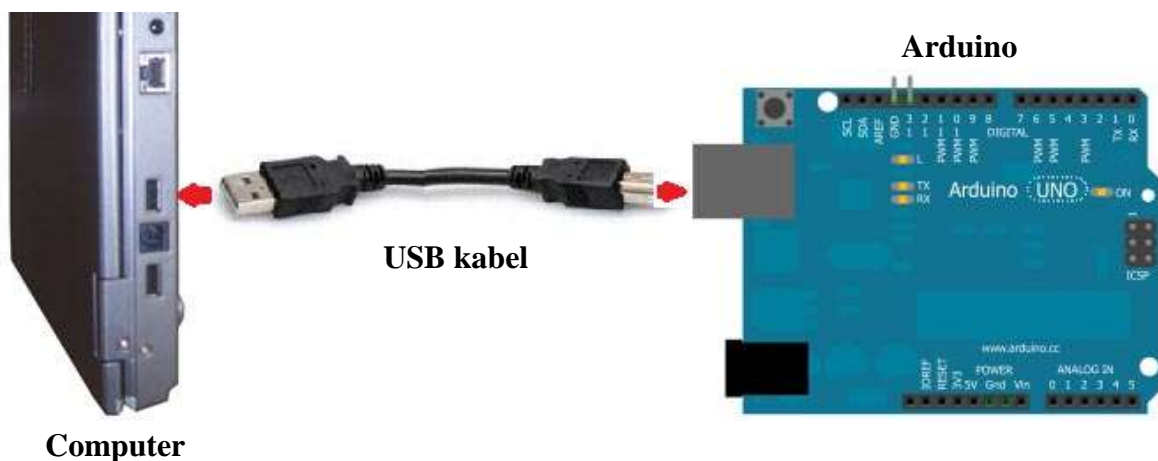
## Hvad er Arduino?

Arduino er et lille Printboard som kan kontrollere forskellige elektroniske komponenter som f.eks. dioder (pærer), motorer, højttalere, sensorer og meget mere! Det er microcontrolleren på boardet (se billede forrige side - den store chip) der styrer strømmen til de forskellige udgange øverst (digital) og nederst (analog)!

Microcontrolleren kan programmeres i et specielt programmeringssprog som du lærer på de følgende sider! På ens computer laver og designer man et program som overføres til Arduino's microcontrollerer! Dette sker igennem et helt almindeligt USB kabel!

## Forbind Arduino til computeren:

Arduino'en kan forbindes til en computer gennem et almindeligt USB kabel - som de fleste vil kende fra f.eks. et printer kabel! Det almindelige USB stik sættes ind i computerens USB port mens den anden del (større stik) af kablet sættes in i arduino'en (se tegning)



## Installation af program:

For at kunne programmere og bruge Arduino'en skal man først installere programmet som findes på denne hjemmeside:

<http://arduino.cc/en/Main/Software>

Her skal man installere den version der passer til det styresystem man har (Windows, Mac eller måske Linux - hvis man er rigtig fremme i skoene).

**NB - Java:** Man skal være opmærksom på at programmet kræver Java installeret på computeren! Hvis man derfor ikke allerede har installeret Java vil man derfor blive bedt om det!



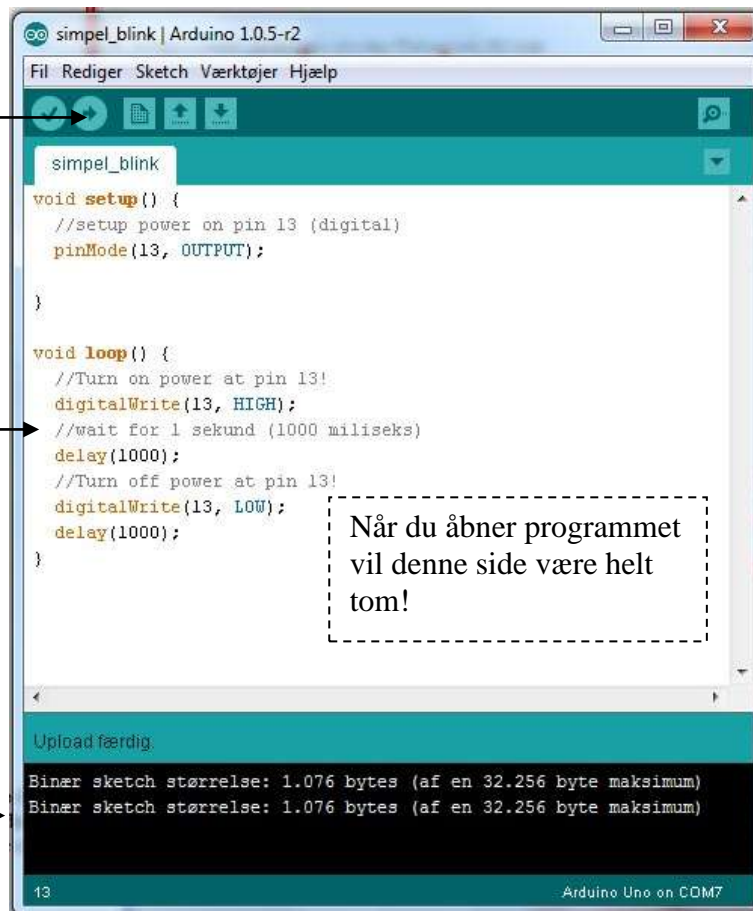
## Start af programmet:

Man burde nu kunne starte arduino programmet ved at klikke på følgende ikon:

Denne knap kompilerer og sender programmet til Arduino'en

Her skrives koden/programmet

Her skriver programmet hvis der er fejl i koden!



## Programmets opbygning:

Når Arduino programmet starter vil der oftest skulle laves noget startup programmeirng/kode! Det kan f.eks. være at åbne forskellige porte og udgange! Dette sker i setup funktionen!

```
void setup() {
    //Her er programkoden til startup!
}
```

Når startup/setup er færdig udfører arduino'en den del af programmet som er i følgende funktion:

```
void loop() {
    //Her er programkoden til resten af tiden Arduino'en kører!
}
```

Loop betyder løkke på dansk og det betyder at Arduino'en bliver ved med, at kalde loop funktionen indtil der slukkes for Arduino'en!

**NB:** Et program behøver ikke at indeholde begge funktioner men kan f.eks. blot have et loop!

**Opgave 1:** Det første program

Programmet nedenfor (skrevet med anderledes skrifttype) vil få en LED pære til at blinke på selve Arduino'en! Pæren hedder L og er placeret lige bag USB kabel stikket på Arduinoen!

**NB:** du behøver ikke at skrive alle linjer markeret med // da det er en kommentar i programmet!

```
void setup() {
  //setup pin nr 13 (finds til i same side som USB stikket!)
  //dette er i øvrigt en kommentar du ikke behøver at skrive!
  pinMode(13, OUTPUT);
}

void loop() {
  //Åben for signal/strøm på pin nr 13!
  digitalWrite(13, HIGH);
  //vent/sov 1 sekund (1000 milisekunder)
  delay(1000);
  //Sluk for signal/strøm på pin nr 13!
  digitalWrite(13, LOW);
  //vent/sov 1 sekund (1000 milisekunder)
  delay(1000);
}
```

**Er du i tvivl om hvor du skal skrive koden?**

Se på billedet på forrige side - her er den samme kode skrevet!

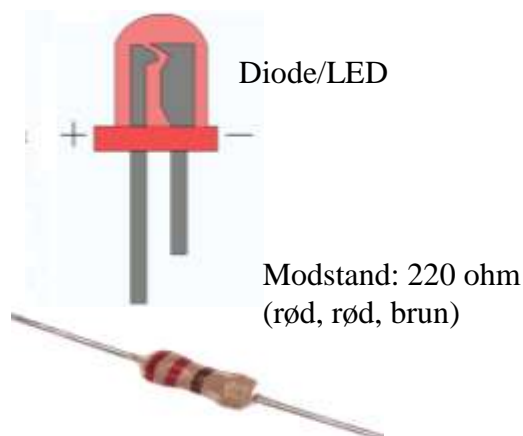
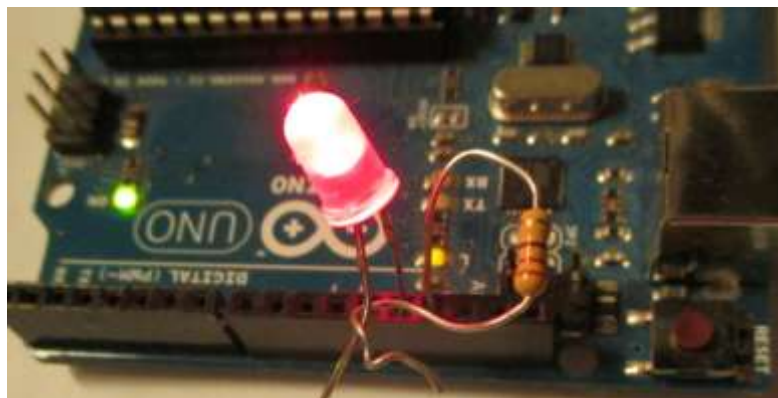
*Hvad gør jeg når jeg har skrevet min program kode?*

Så skal du trykke på ikon nr 2 i menuen

*Hvad gør jeg hvis den siger at der ikke er forbindelse til Arduinoen men det er der?*

Nogen gange sker dette! Træk da usb stikket ud og sæt det ind igen - sørg ligeledes for at den rigtige Serie port er brugt (se billede) samt at det rigtige kort er valgt!





### Opgave 2: Få en diode til at lyse

Vi udvider nu programmet ved at tage en diode som sættes ned i pin 13 og gennem en modstand 220 ohm til GND (jord) på Arduino'en! Dette kan man gøre mens den kører!

**Dioden:** En diode har ligesom så meget andet elektronisk en + (plus) og en - (minus) pol! Dioden lyser kun når strømmen løber den ene vej - det betyder at hvis den vendes forkert så lyser den slet ikke! Plus pol (+) skal sættes ned i 13 og det er det længste ben der er plus (+)!

Prøv kort at sætte bene fra dioden ned i nr 13 og GND - hvis den vender rigtig blinker den!

**Modstanden:** For at beskytte dioden men også selve Arduino'en mod at brænde sammen er det nødvendig at sætte en modstand i fra diodens minus (-) ben og til GND (jord) - se billede!

Modstanden skal sænke strømstyrken så der ikke er noget som brændes af! En modstand måles i ohm og den der skal bruges har 220 ohm! Hvis man er i tvivl kan man altid måle modstanden med et multimeter eller se på farvekoden og 220 ohm har Rød, Rød, Brun! Der er også en fjerde farve ring som er guld farvet! Denne angiver afvigelsen som i dette tilfælde er +/-5 %!

**NB:** Det er ligegyldigt hvilken vej modstanden vender!

### Opgave 3: Ændre perioden for blinkene!

I denne opgave skal du forsøge at ændre blinke perioden - begge delay's kan ændres!

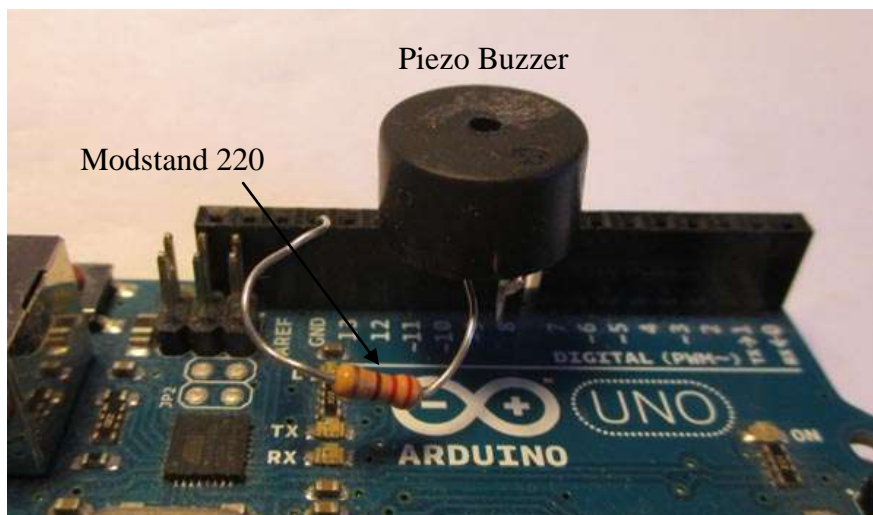
Man kan f.eks. prøve at bruge funktionen `random(talstart, talslut)`

```
random(0, 2000);
```

Her laver Arduinoen et tilfældigt tal imellem 0 og 2000 som den returnerer! Herefter kan man give dette tilfældige tal til delay funktionen som følger:

```
delay(random(0, 2000));
```

Det som sker her er at `random(0, 2000)` kaldes først og giver et tal! Dette tal gives til `delay()` som så får arduinoen til at sove det antal tallet angiver!



#### Opgave 4: At lave en hyletone

Med i pakken med Arduino findes ofte også en lille rund højttaler som kaldes for en Piezo buzzer! Den kan lave en forfærdelig hyletone! Piezo buzzeren har to ben hvor det ene kan f.eks. sættes i pin 10 på den digitale udgang (dem som er bag USB stikket).! Det andet ben forbindes herefter til en modstand (1000 ohm brun, sort, rød eller alternativ 220 ohm!) som går videre til GND. Det er ligegyldigt hvordan buzzeren eller modstanden vendes!

Dem der har lavet Arduino programmet er så glade for den lille højttaler at de har lavet en lille funktion som nemt kan få den til at sige noget!

```
tone(pin-nr, frekvens, længde);
```

Så vores program kommer til at se ud som følger:

```
void loop() {
  tone(10, 1000, 0);
}
```

Her laves en hyletone som antager at buzzeren er sat til i pin nr 10 og sender en tone ud med frekvensen 1000! Længden af tonen er 0!

#### Opgave 5: Lav en lille sang f.eks. C, E G!

Brug delay() funktionen du lærte at bruge fra forrige side!

##### Frekvenser for Noder:

C = 523	C# = 554
D = 587	D# = 622
E = 659	
F = 698	F# = 740
G = 784	G# = 831
A = 880	A# = 932
B = 988	
C = 1047	

**Opgave 5:** Lav en hyletone der bliver højere i tonen

Vi skal nu udvide hyletonen, hvor vi gradvist vil ændre tonen så den bliver pitch'et højere hele tiden!

Dette gøres nemmest ved at bruge en løkke også kaldt et while-loop:

```
void loop() {  
    //Vi laver en variable/tal som skal holde styr på  
    //hvor langt vi er kommet i loopet! int = integer  
    int counter = 0;  
  
    //while = sålænge counter < 10000 så kør loopet  
    while(counter < 10000) {  
        //spil en tone med frekvensen counter  
        tone(10, counter, 100);  
  
        //læg en til counter så frekvensen stiger med 1!  
        counter = counter + 1;  
    }  
}
```