

Varying variables

```
var x = 0;
function setup(){
  ...
}
function draw() {
  x = x+1;
  Eller
  x+=1;
  Eller
  x++;
}
```

- Definer variabler, så du kun behøver at ændre et sted i programmet for at ændre værdier
- God programmeringsstil:
- Erklær dem øverst i koden så de er nemme at finde
- Variabler kan ændres i løbet af programmet

Make your own functions

```
function setup() {...}
```

```
function draw(){
```

```
    myAwesomeFunction(5,5);
```

```
}
```

```
function myAwesomeFunction(xPos,yPos) {
```

```
    ...
```

```
}
```

- Opdel dit program i funktioner!
- Giver overskuelighed både for dig, og den der skal læse din kode
- Hjælper til at undgå at skrive det samme flere gange

Alpha

`fill(__,__);` → gråskala værdi + opacitet/gennemsigtighed

Eller

`fill(__, __, __, __);` → RGB farver + opacitet/gennemsigtighed

Alpha 0 → 100% transparent

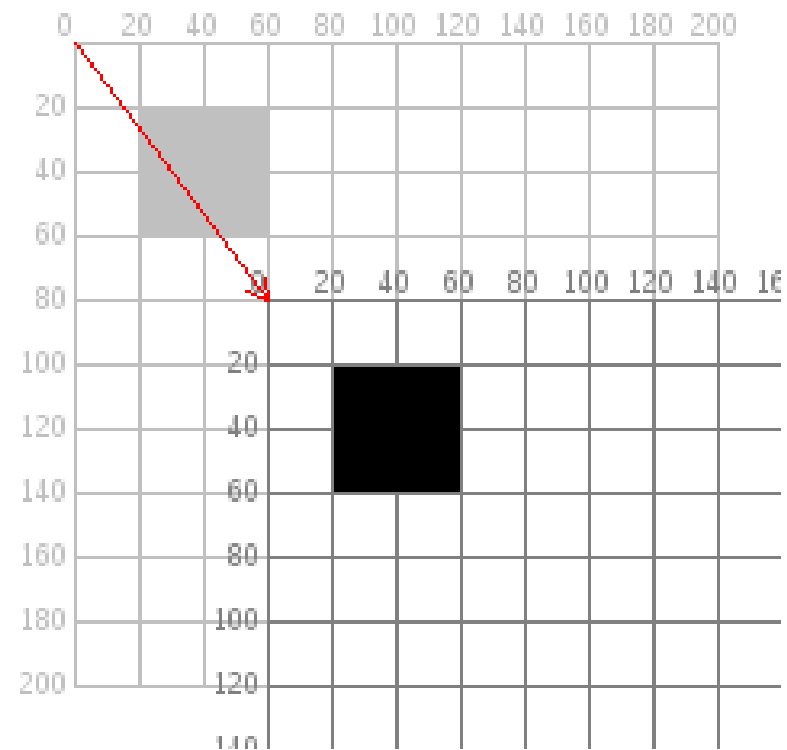
Alpha 255 → 0% transparent, dvs. udfyldt

Gælder også for `stroke();`

Translate

`translate(__, __);`

Flytter punktet (0,0)



Push og pop matrix

```
push();
```

```
    translate(__);
```

```
    rotate(__);
```

```
pop();
```

Alt herimellem der ændrer *matrixet* gælder kun mellem de to funktioner (f.eks. rotate() og translate())

Framerate

```
frameRate(__);
```

Definerer framerate

Dvs. antal frames pr. sekund

Default er 60

Dvs. hvor mange gange draw looper pr. sekund

frameCount

Den indbyggede variabel:

`frameCount`

Indeholder altid antallet af frames der er blevet vist siden programmet startede.

millis()

Funktionen

millis()

Returnerer altid antallet af millisekunder der er gået siden programmet startede.

Tilsvarende er der også: `day()`, `hour()`, `minute()`, `month()`, `second()` og `year()`

Se referencen: <https://p5js.org/reference/>

Listening events – Basic conditions

```
if (x>700) {  
    fill(0);  
  
} else if (x<300) {  
    fill(255);  
  
} else {  
    fill(150);  
  
}
```

Conditionals

> Mere end
< Mindre end
<= mindre end eller lig med
== lig med
>= mere end eller lig med
!= ikke lig med

|| ELLER

&& OG

For-loop

```
for (var i = 0; i < 10; i++) {  
    var x=20+i*50;  
    ellipse(x,height/2,20,20);  
}
```



Godt så man undgår kodeduplikering!

Arrays

En liste af ting (tal, tekst, objekter, billeder...)

```
var words= ["I", "love", "Arrays", "!"];
```

Eller

```
var words= [ ];  
words[0]="I";  
words[1]="love";  
words[2]="Arrays";  
words[3]="!";
```

Arrays and for-loop

Går godt sammen, fordi for-loop kan opstille nogle generelle regler for din Array

```
for (var i=0; i<words.length; i++) {  
    text(words[i], 100+i*100, height/2);  
}
```

I love Arrays !

MiniEx2 – discussion

Sæt jer sammen i grupper, der ikke er dem I ellers er i gruppe med.

Præsenter jeres miniEx fra sidste uge for hinanden.

- Forklar din vision.
- Vær så præcis som mulig med hvorfor du bruger lige præcis den syntax det sted igennem koden.
- Giv gerne feedback og forslag til hinanden.

Winnies code

```
3  function setup() {
4    createCanvas(windowWidth, windowHeight); //create a drawing canvas
5    background(10);
6    frameRate (10); //try to change this parameter
7  }
8
9  function draw() {
10   noStroke();
11   fill(10, 80); //check this syntax with alpha value
12   rect(0, 0, width, height);
13   drawThrobber(9);
14 }
15
16 function drawThrobber(num) {
17   push();
18   translate(width/2, height/2);
19   // 360/num >> degree of each ellipse's turn, frameCount%num >> get the remainder that indicates the movement of the ellipse
20   var cir = 360/num*(frameCount%num); //to know how many possible positions.
21   rotate(radians(cir));
22   noStroke();
23   fill(255,255,0);
24   ellipse(35,0,22,22); //the moving dot(s), the x is the distance from the center
25   pop();
26
27   stroke(255,0,0);
28   line(60,0,60,height); //a static line
29
30 }
```

Frames pr. sekund

Alpha værdi

Selvdefineret funktion med parameter

Inden for push/pop

Asterisk Painting by John P. Bell

Code:

<https://github.com/AUAP/AP2018/blob/master/class03/sketch03/sketch03.js>

Show:

<https://rawgit.com/AUAP/AP2018/master/class03/sketch03/index.html>

MiniEx 3

Winnie skriver:

- Make sure you have read the text by Winnie Soon and Jason Farman.
- Study the syntaxes that have been discussed in class, and then redesign an animated throbber.
- Upload your program/result to your own Github account under a folder called **mini_ex3**. (Make sure your program can be run on a web browser)
- Create a readme file (README.md) and upload to the same mini_ex3 directory.
- Provide peer-feedback to 2 of your classmates on their works by creating "issues" on his/her github corresponding repository. Write with the issue title "**Feedback on mini_ex(?) by (YOUR FULL NAME)**"

Throbber

Hvad er en throbber?

Hvad skal en throbber kunne?

Hvordan kan en throbber se ud?

Eksempel:

<https://rawgit.com/RaggedyAnn/OldMinis/master/2nd%20weekly%20mini%20exercise/aCalmingThrobber/index.html>

ReadMe

Winnie skriver:

- A screenshot of your program
- A URL link to your program and run on a browser.
- Describe about your throbber design, both *conceptually* and *technically*.
- What is the **time-related syntax/function** that you have used in your program? and why you use in this way?
- Think about a throbber that you have encountered in digital culture e.g streaming video on YouTube or loading latest feeds on Facebook or waiting a ticket transaction, what do you think a throbber *tells us*, or *hide*, about?

Design din egen throbber!

Vær kreative, den behøver ikke at være som Winnies
overhovedet