METODER/FUNKTIONER

MODULARITET & METODER

Metoder er små moduler!
Kode som kan pakkes og genbruges i programmet!

Metoden kan kaldes igen og igen efter behov –
I stedet for at skulle skrive koden linje for linje for linje...

En metode har et *afgrænset ansvar*, og man vælger et sigende eller selvforklarende navn…
Fx random(10,100) – kald af metode/funktion som returnerer værdi…

OPGAVE: simpel metode
Skriv først simpel metode uden parametre, men som bare udskriver sætningen:
”Metoder er små moduler, som kan kaldes igen og igen”
Kald metoden to gange!

```
void minMetode(){
println(”...”);
}
```

Kald din metode indefra draw – sæt noLoop(); - så draw ikke kaldes igen og igen…

Metode med parametre - én returværdi

Der kan returneres netop én værdi

Returværdiens datatype angives før metodenavnet i definitionen

void betyder ”ingen returværdi”

Værdien returneres med **return**

Definition

```
public returdatatype Metode ( datatype1 var1 , datatype2 var2 , ... )
{       //sætninger
        return lokalvariabel ;
}
```

Kald

```
variabel = Metode ( variabel1 , variabel2 , ... ) ;
```

At skrive egne funktioner/metoder

```
void setup() {
  size(100, 100);
  noStroke();
  noLoop();
  smooth();
}

void draw() {
  // Højre øje
  fill(255);
  ellipse(65, 44, 60, 60);
  fill(0);
  ellipse(75, 44, 30, 30);
  fill(255);
  ellipse(81, 39, 6, 6);
  // Venstre øje
  fill(255);
  ellipse(20, 50, 60, 60);
  fill(0);
  ellipse(30, 50, 30, 30);
  fill(255);
  ellipse(36, 45, 6, 6);
}
```

```
void setup() {
  size(100, 100);
  noStroke();
  noLoop();
  smooth();
}

void draw() {
  eye(65, 44);
  eye(20, 50);
}

void eye(int x, int y) {
  fill(255);
  ellipse(x, y, 60, 60);
  fill(0);
  ellipse(x+10, y, 30, 30);
  fill(255);
  ellipse(x+16, y-5, 6, 6);
}
```

At skrive egne funktioner/metoder

- Min egen funktion hedder ”eye”
- Den har to parametre x og y som skal være heltal (int)
- Den tegner en figur i punktet (x,y) som ligner et øje
- Man skriver altså bare fx **eye(17,57)** når den fx skal tegnes i punktet (17,57)

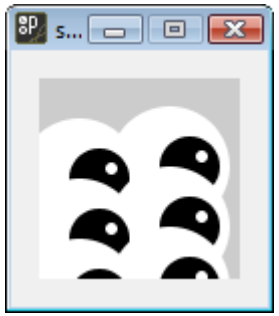
```
void setup() {
  size(100, 100);
  noStroke();
  noLoop();
  smooth();
}

void draw() {
  eye(65, 44);
  eye(20, 50);
}

void eye(int x, int y) {
  fill(255);
  ellipse(x, y, 60, 60);
  fill(0);
  ellipse(x+10, y, 30, 30);
  fill(255);
  ellipse(x+16, y-5, 6, 6);
}
```

- I Processing kan funktioner defineres og bruges fra bl.a. setup() og draw()

- Når en funktion først er defineret kan den genbruges ubegrænset



```
void setup() {
  size(100, 100);
  noStroke();
  smooth();
  noLoop();
}

void draw() {
  eye(65, 44);
  eye(20, 50);
  eye(65, 74);
  eye(20, 80);
  eye(65, 104);
  eye(20, 110);
}

void eye(int x, int y) {
  fill(255);
  ellipse(x, y, 60, 60);
  fill(0);
  ellipse(x+10, y, 30, 30);
  fill(255);
  ellipse(x+16, y-5, 6, 6);
}
```

Returnering af værdi fra funktion/metode

parametre

```
void setup() {
  size(100, 100);
  float f = udregnGennemsnit(12.0, 6.0);
  println(f);
}

float udregnGennemsnit(float num1, float num2)
{
  float gennemSnit = (num1 + num2) / 2.0;
  return gennemSnit;
}
```

Kald af metode
Variabel til at modtage returværdi

Metodens argumenter defineret

Type

Retur af værdi