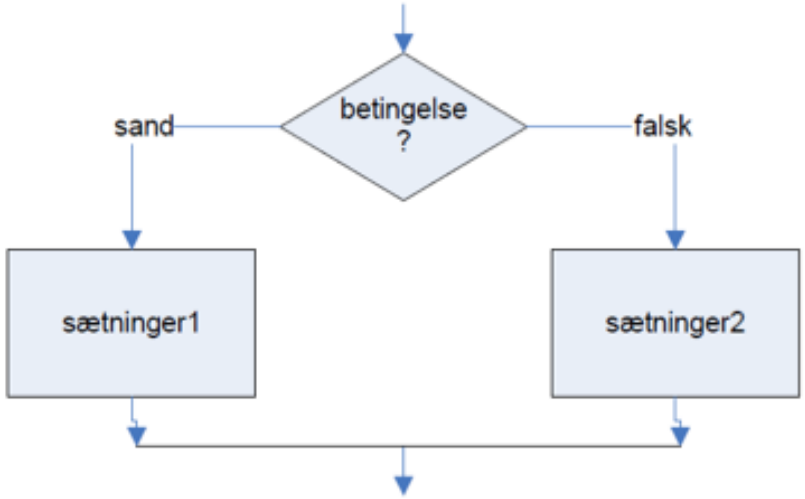
Koden afvikles sekventielt – linje for linje
Men kodens kørsel kan kontrolleres

via KONTROLSTRUKTURER

if, else if, else
switch
Løkker – for(….), while

Valg - if

Valg afhængig af en betingelse (binært valg)



Sætninger2
kan være tom

Syntaks

```
if ( betingelse )
{     //sætninger
}
```

```
if ( betingelse )
{     //sætninger1
}
else
{     //sætninger2
}
```

Hvis dette er sandt, så…

Ellers kør dette!

Relationsoperatorer

Relationsoperatorer benyttes i betingelser (f.eks. **if**-sætninger)

2 værdier sammenlignes, og resultatet returneres som en boolsk (logisk) værdi – true eller false

Operator:	*Beskrivelse:*
==	Lig med
!=	Forskellig fra
<	Mindre end
<=	Mindre end eller lig med
>	Større end
>=	Større end eller lig med

F.eks.:

```
if (tal1 == tal2)
…
if (tal1 <= 17)
…
```

```
int x = 100;

if (x > 100) {
 ellipse(50, 50, 36, 36);
}

if (x < 100) {
 rect(33, 33, 34, 34);
}

if (x == 100) {
 line(20, 20, 80, 80);
}
```

- Hvad udskriver programmet, og hvad er forskellen imellem = og == ?

```
int x = 100;

if (x > 100) {
 ellipse(50, 50, 36, 36);
}

if (x < 100) {
 rect(33, 33, 34, 34);
}

if (x == 100) {
 line(20, 20, 80, 80);
}
```

- Logiske udtryk i forbindelse med kontrolstrukturer, som styrer programmets kørsel:
  - kan angives med relationerne
    `>, <, >=, <=, ==, !=`
  - og med operatorerne
    `&&, ||, !`
    `("og", "eller", "ikke")`

Kontrolstrukturer – if, else if, else

Skriv et program hvor baggrundsfarven er sort, hvis musen er på venstre side, og hvid hvis musen er på højre side

- *background(0) sætter farven til sort og baggrund(255) til hvid – skal placeres inden i draw*
- *mouseX giver adgang til musens x-position*
- *width giver adgang til vinduets bredde*

```
void setup(){
  size(1200,600); }

void draw(){
   // betingelser sættes ind i draw-loopet
  if(mouseX….)
}
```

- og med operatorerne

```
&&, ||, !
("og", "eller",
 "ikke")
```

Kontrolstrukturer – if, else if, else

Skriv et program hvor baggrundsfarven ændrer sig
alt efter hvor på skærmen musen befinder sig på x-aksen

- *tre forskellige farver ønskes – brug fx if i kombination med else if samt else*
- *background(0) sætter farven til sort*
- *mouseX giver adgang til musens x-position*
- *width giver adgang til vinduets bredde*
- *Varier eventuelt med y-positionen hvor && eller || bruges*

```
void setup(){
  size(1200,600); }

void draw(){
  if(mouseX….)
}
```

- og med operatorerne

```
&&, ||, !
("og", "eller",
 "ikke")
```

Logiske operatorer

```
if ((i > 35) && (i < 60))
{
  stroke(0);
}
```

Logiske operatorer anvendes, når flere relationer (logiske udtryk) skal sammensættes

Operator:	*Beskrivelse:*
&&	And / og
||	Or / eller
!	Not / ikke

~~**if** (10<tal<100)~~

F.eks.:

if (tal1==tal2) **&&** (tal1<=17)

…

if (tal1==tal2) **||** (tal1<=17)

…

if ! (tal1==tal2)

…

if (10<tal) **&&** (tal<100)

…

Kontrolstrukturer – if, else if, else

Skriv program hvor en knap ændrer fx baggrundsfarven eller et andet elements position:

- *mouseX giver adgang til musens x-position*
- *mousePressed giver adgang til om musen er klikket eller ej*
- *brug int-variable til at tegne rect-knappen og til at kontrollere om musen er inden for knappens område eller ej*
- *brug && til at binde betingelsen sammen.*

```
void setup(){
 size(1200,600);
}
void draw(){
   // tegn rect her og skriv betingelsen ind her…
   // er musen inden for knappens felt og er den klikket?
}
```

- Logiske udtryk i forbindelse med kontrolstrukturer, som styrer programmets kørsel:
  - kan angives med relationerne
    `>, <, >=, <=, ==, !=`
  - og med operatorerne
    `&&, ||, !`
    `("og", "eller", "ikke")`