

# FORGRENINGER

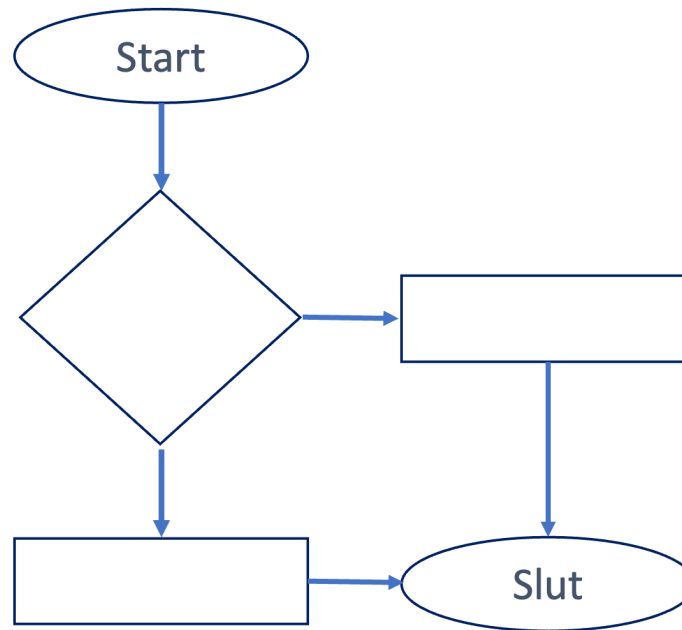
---

På tværs...

# FORGRENINGER (= BRANCHINGS)

---

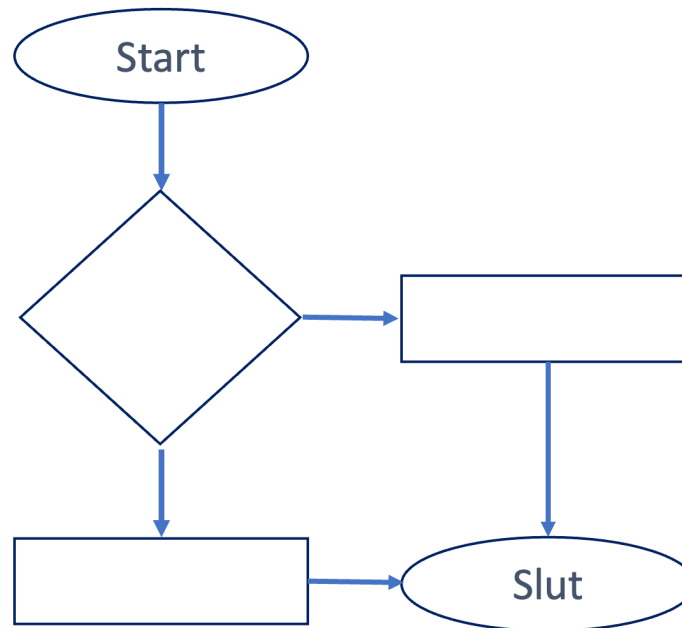
Forgreninger bruges, når det videre forløb afhænger af, om en betingelse er opfyldt (dvs sand) eller ikke er opfyldt (dvs falsk)



# FORGRENINGER (= BRANCHINGS)

---

Forgreninger bruges, når det videre forløb afhænger af, om en betingelse er opfyldt (dvs sand) eller ikke er opfyldt (dvs falsk)



et rutediagram, der viser  
forgreningen –  
rutediagrammer bruges  
ikke så meget mere...

# PSEUDOKODEFORGRENINGER

---

## Eksempler

hvis din score er større end min  
udskriv, at du vandt

---

hvis mønten giver plat  
udskriv, at du bestemmer

ellers  
udskriv, at jeg bestemmer

# PSEUDOKODEFORGRENINGER

---

## Eksempler

hvis din score er større end min

→ udskriv, at du vandt

---

hvis mønten giver plat

→ udskriv, at du bestemmer

ellers

→ udskriv, at jeg bestemmer

bemærk indrykninger  
(= indentations)

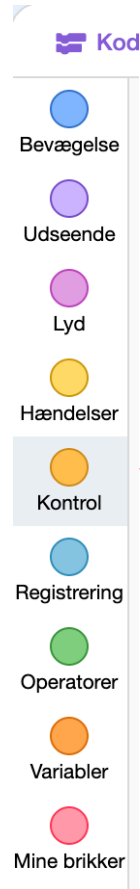
# SCRATCH-FORGRENINGER

---

# SCRATCH-FORGRENINGER

---

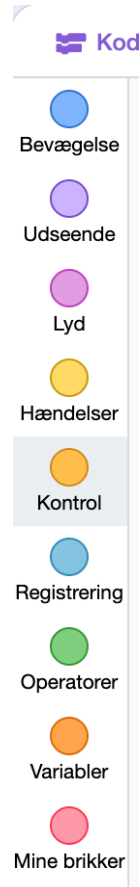
Der er to slags forgreninger i Scratch, som findes under Kontrol



# SCRATCH-FORGRENINGER

---

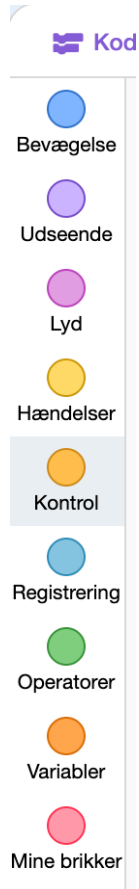
Der er to slags forgreninger i Scratch, som findes under Kontrol





# SCRATCH-FORGRENINGER

Der er to slags forgreninger i Scratch, som findes under Kontrol

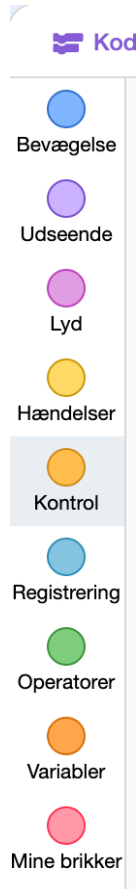


en betingelse, der er sand eller falsk



# SCRATCH-FORGRENINGER

Der er to slags forgreninger i Scratch, som findes under Kontrol



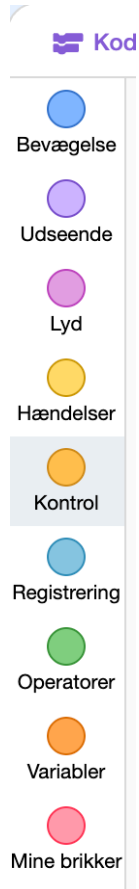
en betingelse, der er sand eller falsk, fx



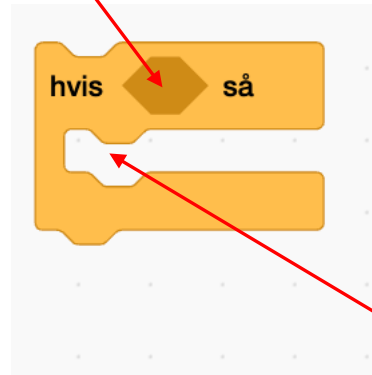
# SCRATCH-FORGRENINGER

---

Der er to slags forgreninger i Scratch, som findes under Kontrol



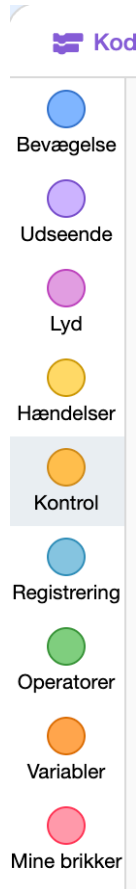
en betingelse, der er sand eller falsk



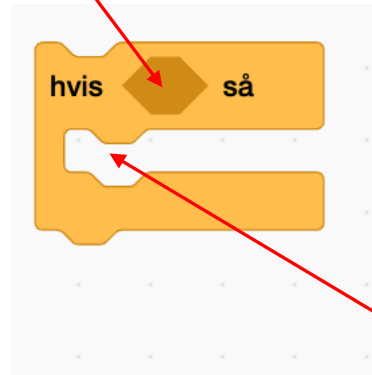
hvis betingelsen er sand, udføres de blokke, der skrives her

# SCRATCH-FORGRENINGER

Der er to slags forgreninger i Scratch, som findes under Kontrol



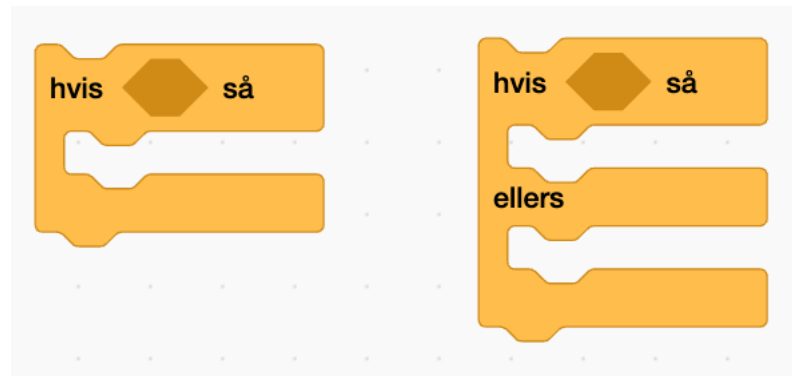
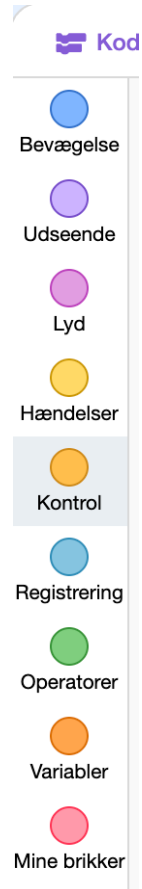
en betingelse, der er sand eller falsk



hvis betingelsen er sand, udføres de blokke, der skrives her  
og hvis betingelsen er falsk, sker der ikke noget

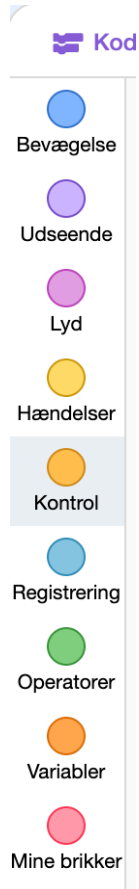
# SCRATCH-FORGRENINGER

Der er to slags forgreninger i Scratch, som findes under Kontrol

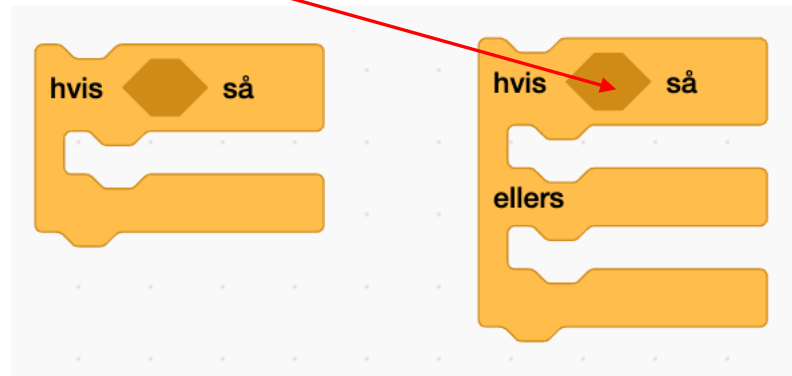


# SCRATCH-FORGRENINGER

Der er to slags forgreninger i Scratch, som findes under Kontrol

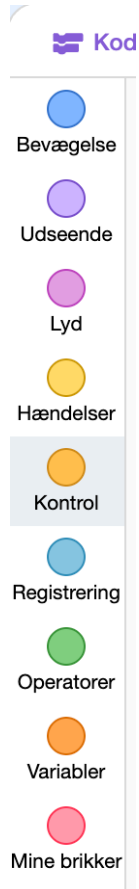


en betingelse, der er sand eller falsk

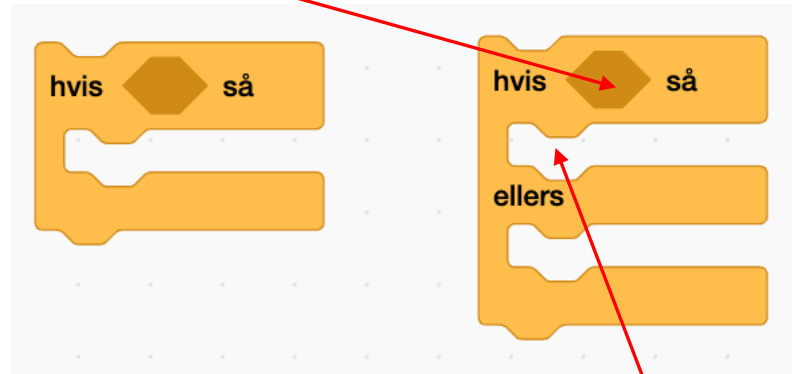


# SCRATCH-FORGRENINGER

Der er to slags forgreninger i Scratch, som findes under Kontrol



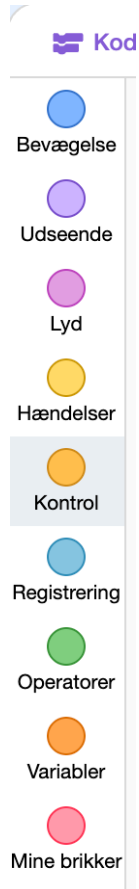
en betingelse, der er sand eller falsk



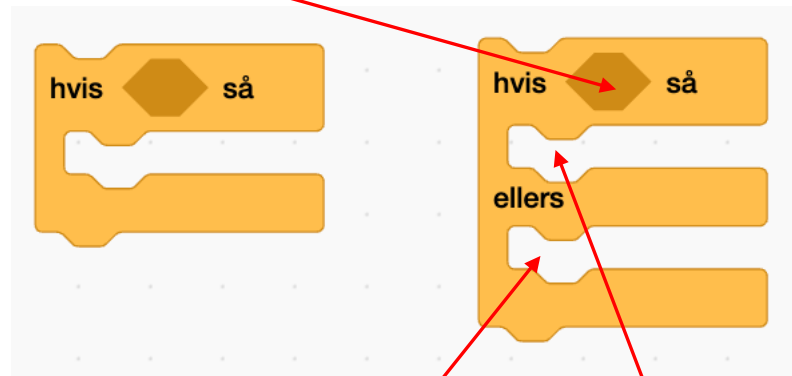
hvis betingelsen er sand, udføres de blokke, der placeres her

# SCRATCH-FORGRENINGER

Der er to slags forgreninger i Scratch, som findes under Kontrol



en betingelse, der er sand eller falsk



hvis betingelsen er sand, udføres de blokke, der placeres her  
og hvis falsk de blokke, der placeres her

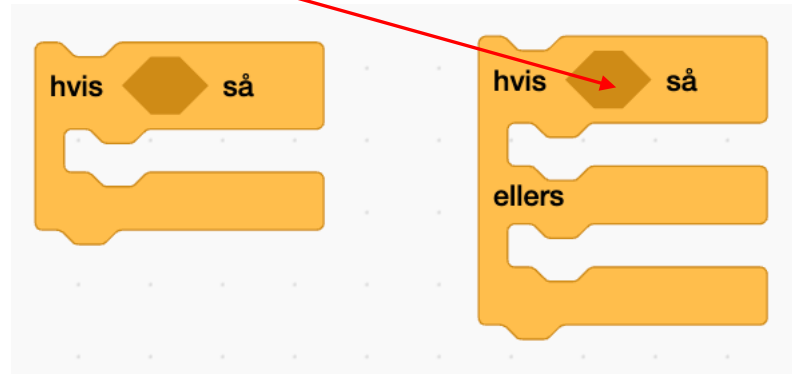


# SCRATCH-FORGRENINGER

Der er to slags forgreninger i Scratch, som findes under Kontrol



en betingelse, der er sand eller falsk



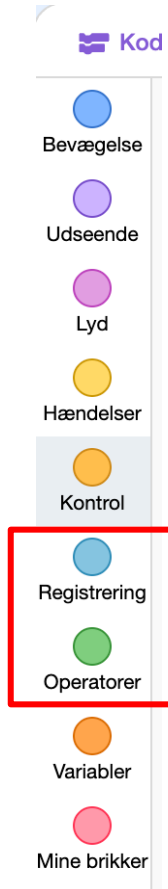
hvis betingelsen er sand, udføres de blokke, der placeres her

og hvis falsk de blokke, der placeres her

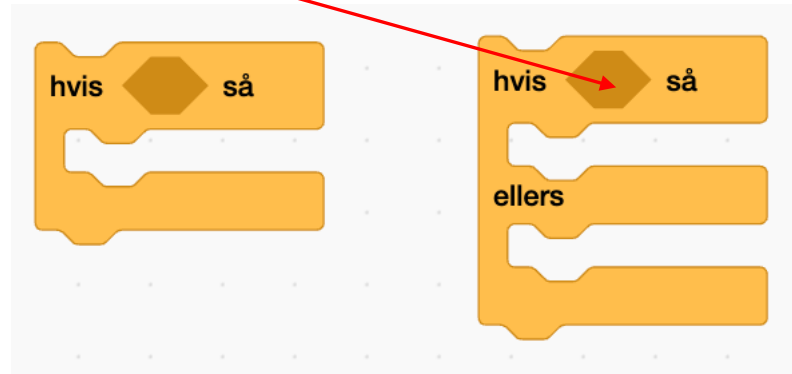
betingelser sammensættes af blokke fra Operatorer og Registreringer

# SCRATCH-FORGRENINGER

Der er to slags forgreninger i Scratch, som findes under Kontrol



en betingelse, der er sand eller falsk



hvis betingelsen er sand, udføres de blokke, der placeres her

og hvis falsk de blokke, der placeres her

betingelser sammensættes af blokke fra Operatorer og Registreringer



# PYTHON- FORGRENINGER

---

# PYTHON- FORGRENINGER

---

Backus Naur Form (BNF)

$\langle \textit{forgrening} \rangle ::= \langle \textit{if} - \textit{type1} \rangle | \langle \textit{if} - \textit{type2} \rangle | \langle \textit{if} - \textit{type3} \rangle$

# PYTHON- FORGRENINGER

---

Backus Naur Form (BNF)

$\langle \textit{forgrening} \rangle ::= \langle \textit{if} - \textit{type1} \rangle | \langle \textit{if} - \textit{type2} \rangle | \langle \textit{if} - \textit{type3} \rangle$

$\langle \textit{if} - \textit{type1} \rangle ::= \textit{if} \langle \textit{betingelse} \rangle : \langle \textit{blok} \rangle$

# PYTHON- FORGRENINGER

---

Backus Naur Form (BNF)

$\langle \text{forgrening} \rangle ::= \langle \text{if} - \text{type1} \rangle | \langle \text{if} - \text{type2} \rangle | \langle \text{if} - \text{type3} \rangle$

$\langle \text{if} - \text{type1} \rangle ::= \text{if} \langle \text{betingelse} \rangle : \langle \text{blok} \rangle$

$\langle \text{if} - \text{type2} \rangle ::= \text{if} \langle \text{betingelse} \rangle : \langle \text{blok} \rangle \text{ else: } \langle \text{blok} \rangle$

# PYTHON- FORGRENINGER

---

Backus Naur Form (BNF)

$\langle \text{forgrening} \rangle ::= \langle \text{if} - \text{type1} \rangle | \langle \text{if} - \text{type2} \rangle | \langle \text{if} - \text{type3} \rangle$

$\langle \text{if} - \text{type1} \rangle ::= \text{if} \langle \text{betingelse} \rangle : \langle \text{blok} \rangle$

$\langle \text{if} - \text{type2} \rangle ::= \text{if} \langle \text{betingelse} \rangle : \langle \text{blok} \rangle \text{ else: } \langle \text{blok} \rangle$

$\langle \text{if} - \text{type3} \rangle ::= \text{if} \langle \text{betingelse} \rangle : \langle \text{blok} \rangle (\text{elif} \langle \text{betingelse} \rangle : \langle \text{blok} \rangle)^+ \text{ else: } \langle \text{blok} \rangle$

# PYTHON- FORGRENINGER

---

Backus Naur Form (BNF)

$\langle \text{forgrening} \rangle ::= \langle \text{if - type1} \rangle | \langle \text{if - type2} \rangle | \langle \text{if - type3} \rangle$

$\langle \text{if - type1} \rangle ::= \text{if } \langle \text{betingelse} \rangle : \langle \text{blok} \rangle$

$\langle \text{if - type2} \rangle ::= \text{if } \langle \text{betingelse} \rangle : \langle \text{blok} \rangle \text{ else: } \langle \text{blok} \rangle$

$\langle \text{if - type3} \rangle ::= \text{if } \langle \text{betingelse} \rangle : \langle \text{blok} \rangle (\text{elif } \langle \text{betingelse} \rangle : \langle \text{blok} \rangle)^+ \text{ else: } \langle \text{blok} \rangle$

$\langle \text{blok} \rangle ::= \langle \text{ny linje} \rangle \langle \text{indent} \rangle \langle \text{sætninger} \rangle \langle \text{ny linje} \rangle \langle \text{dedent} \rangle$



# PYTHON- FORGRENINGER

---

## Type 1

```
if < betingelse >:  
    ...  
    ...  
    ...
```

## Type 2

```
i f <betingelse >:  
    ...  
    ...  
    ...  
e l s e:  
    ...  
    ...  
    ...
```

## Type 3

```
i f <betingelse >:  
    ...  
    ...  
    ...  
e l i f <betingelse >:  
    ...  
    ...  
    ...  
e l s e:  
    ...  
    ...  
    ...
```

# PYTHON- FORGRENINGER

---

## Type 1

```
if < betingelse >:  
    ...  
    ...  
    ...
```

## Type 2

```
i f <betingelse >:  
    ...  
    ...  
    ...  
e l s e :  
    ...  
    ...  
    ...
```

## Type 3

```
i f <betingelse >:  
    ...  
    ...  
    ...  
e l i f <betingelse >:  
    ...  
    ...  
    ...  
e l s e :  
    ...  
    ...  
    ...
```

Bemærk indrykningerne, som  
opstår automatisk i Jupiter Notebook  
efter et kolon og Enter

# PYTHON- FORGRENINGER

## Type 1

```
if < betingelse >:  
    ...  
    ...  
    ...
```

## Type 2

```
i f <betingelse >:  
    ...  
    ...  
    ...  
e l s e :  
    ...  
    ...  
    ...
```

## Type 3

```
i f <betingelse >:  
    ...  
    ...  
    ...
```

```
e l i f <betingelse >:  
    ...  
    ...  
    ...
```

```
e l s e :  
    ...  
    ...  
    ...
```

kan gentages  
flere gange

Bemærk indrykningerne, som  
opstår automatisk i Jupiter Notebook  
efter et kolon og Enter

# PÅ TVÆRS...

---

## Pseudokode

---

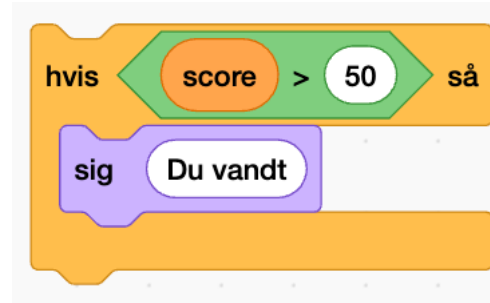
hvis score er større end 50  
    skriv, at du vandt

# PÅ TVÆRS...

## Pseudokode

hvis score er større end 50  
  skriv, at du vandt

## Scratch

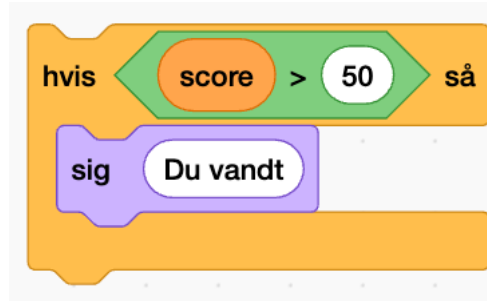


# PÅ TVÆRS...

## Pseudokode

hvis score er større end 50  
  skriv, at du vandt

## Scratch



## Python

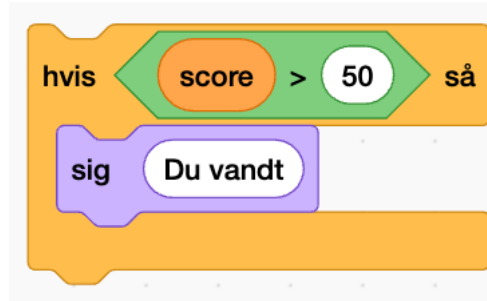
```
if score > 50:  
    print('Du vandt')
```

# PÅ TVÆRS...

## Pseudokode

hvis score er større end 50  
    skriv, at du vandt

## Scratch



## Python

```
if score > 50:  
    print('Du vandt')
```

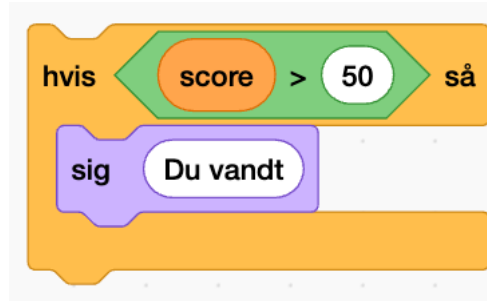
hvis kondital er mindre end 30  
    skriv Dårlig form  
ellers  
    skriv Ikke dårlig

# PÅ TVÆRS...

## Pseudokode

hvis score er større end 50  
    skriv, at du vandt

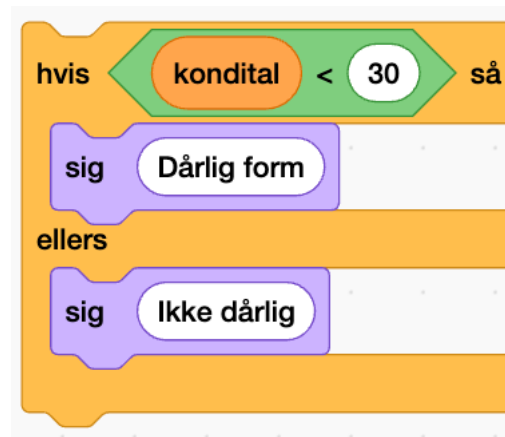
## Scratch



## Python

```
if score > 50:  
    print('Du vandt')
```

hvis kondital er mindre end 30  
    skriv Dårlig form  
ellers  
    skriv Ikke dårlig



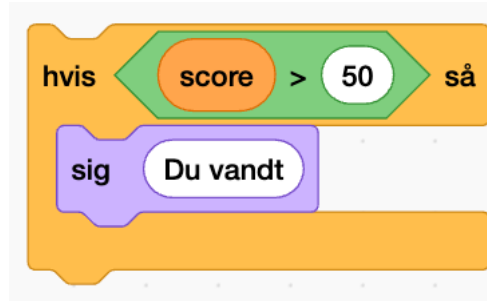


# PÅ TVÆRS...

## Pseudokode

hvis score er større end 50  
    skriv, at du vandt

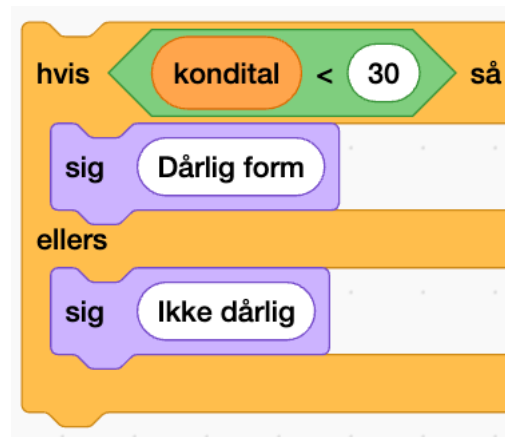
## Scratch



## Python

```
if score > 50:  
    print('Du vandt')
```

hvis kondital er mindre end 30  
    skriv Dårlig form  
ellers  
    skriv Ikke dårlig



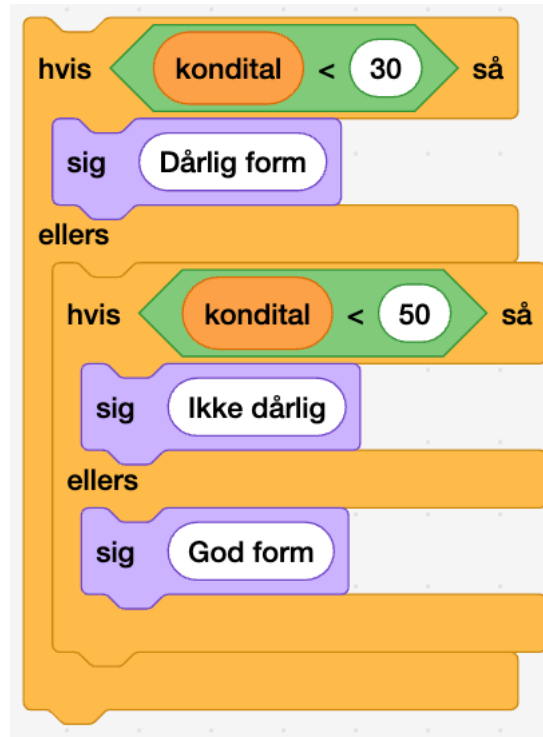
```
if kondital < 30:  
    print('Dårlig form')  
else:  
    print('Ikke dårlig')
```

# PÅ TVÆRS...

## Pseudokode

hvis kondital er mindre end 30  
    skriv Dårlig form  
ellers hvis kondital er...  
    skriv Ikke dårlig  
ellers  
    skriv God form

## Scratch



## Python

```
if kondital < 30:  
    print('Dårlig form')  
elif kondital < 50:  
    print('Ikke dårlig')  
else:  
    print('God form')
```