



Programmering modul 03

CSIK E2025

I dag

Evaluering af udtryk igen

Pause 09:15-09:25

Syntaks og semantik af kommandoer fra 09:25

Hold selv pause, når det passer teamet

Fælles opsamling fra 11:45

Frokost 12:00

codeabby.com



I teams

Canvas:
samarbejdsstrukturer03.pdf

codeabby.com



I teams

Canvas:

samarbejdsstrukturer03.pdf

Få alle i teamet frem
til samme side i
"Stack machine"

Fortsæt arbejdet herfra

Opsamling: evaluering af udtryk

Hvor er du mest forvirret?



Evaluering af logiske udtryk

```
code(e1 and e2) = code(e1)  
                  code(e2)  
                  and
```

```
code(e1 or e2)  = code(e1)  
                  code(e2)  
                  or
```

Evaluering af logiske udtryk

```
code(e1 and e2) = code(e1)
                  code(e2)
                  and
code(e1 or e2)  = code(e1)
                  code(e2)
                  or
```

b1	b2	b1 and b2	b1 or b2
true	true	true	true
true	false	false	true
false	true	false	true
false	false	false	false

Evaluering af logiske udtryk

```
code(e1 and e2) = code(e1)
                  code(e2)
                  and
code(e1 or e2)  = code(e1)
                  code(e2)
                  or
```

b1	b2	b1 and b2	b1 or b2
true	true	true	true
true	false	false	true
false	true	false	true
false	false	false	false

e1 and e2

Evaluer **e1**. Hvis værdien er **false**, så er det værdien af hele udtrykket.
Ellers evaluér **e2** og lad det være værdien af hele udtrykket.

e1 or e2

Evaluer **e1**. Hvis værdien er **true**, så er det værdien af hele udtrykket.
Ellers evaluér **e2** og lad det være værdien af hele udtrykket.

Evaluering af logiske udtryk

```
code(e1 and e2) = code(e1)
                  code(e2)
                  and
code(e1 or e2)  = code(e1)
                  code(e2)
                  or
```

b1	b2	b1 and b2	b1 or b2
true	true	true	true
true	false	false	true
false	true	false	true
false	false	false	false

Forkert!
Med den oversættelse
får vi altid evalueret e2.

e1 and e2

Evaluer **e1**. Hvis værdien er **false**, så er det værdien af hele udtrykket.
Ellers evaluér **e2** og lad det være værdien af hele udtrykket.

e1 or e2

Evaluer **e1**. Hvis værdien er **true**, så er det værdien af hele udtrykket.
Ellers evaluér **e2** og lad det være værdien af hele udtrykket.

Evaluering af logiske udtryk

```
code(e1 and e2) = code(e1)
                  code(e2)
                  and
code(e1 or e2)  = code(e1)
                  code(e2)
                  or
```

b1	b2	b1 and b2	b1 or b2
true	true	true	true
true	false	false	true
false	true	false	true
false	false	false	false

Forkert!
Med den oversættelse
får vi altid evalueret e2.

Vi er nødt til at kunne
springe over code(e2)
afhængig af resultatet
af at afvikle code(e1) udtrykket.

e1 and e2

Evaluer **e1**. Hvis værdien er **false**, så er det værdien af hele udtrykket.
Ellers evaluér **e2** og lad det være værdien af hele udtrykket.

e1 or e2

Evaluer **e1**. Hvis værdien er **true**, så er det værdien af hele udtrykket.
Ellers evaluér **e2** og lad det være værdien af hele udtrykket.

Betingede udtryk / conditionals

e1 and e2

Evaluér **e1**. Hvis værdien er **false**, så er det værdien af hele udtrykket.
Ellers evaluér **e2** og lad det være værdien af hele udtrykket.

e1 or e2

Evaluér **e1**. Hvis værdien er **true**, så er det værdien af hele udtrykket.
Ellers evaluér **e2** og lad det være værdien af hele udtrykket.

Betingede udtryk / conditionals

e1 and e2

Evaluér **e1**. Hvis værdien er **false**, så er det værdien af hele udtrykket.
Ellers evaluér **e2** og lad det være værdien af hele udtrykket.

e1 or e2

Evaluér **e1**. Hvis værdien er **true**, så er det værdien af hele udtrykket.
Ellers evaluér **e2** og lad det være værdien af hele udtrykket.

e1 ? e2 : e3 (C, C++, Java, JavaScript, C#)

e2 if e1 else e3 (Python)

Evaluér **e1**.
Hvis værdien er **true**, så evaluér **e2** og lad det være værdien af hele udtrykket.
Ellers evaluér **e3** og lad det være værdien af hele udtrykket.

Betingede udtryk / conditionals

e1 and e2

Evaluér **e1**. Hvis værdien er **false**, så er det værdien af hele udtrykket.
Ellers evaluér **e2** og lad det være værdien af hele udtrykket.

e1 or e2

Evaluér **e1**. Hvis værdien er **true**, så er det værdien af hele udtrykket.
Ellers evaluér **e2** og lad det være værdien af hele udtrykket.

e1 ? e2 : e3 (C, C++, Java, JavaScript, C#)

e2 if e1 else e3 (Python)

Evaluér **e1**.
Hvis værdien er **true**, så evaluér **e2** og lad det være værdien af hele udtrykket.
Ellers evaluér **e3** og lad det være værdien af hele udtrykket.

e1 and e2 betyder det samme som

e1 ? e2 : false

e1 or e2 betyder det samme som

e1 ? true : e2

Betingede udtryk / conditionals

```
code(e1 ? e2 : e3) =
```

Betingede udtryk / conditionals

```
code(e1 ? e2 : e3) = code(e1)
                    jmpf else
                    code(e2)
                    jump done
                    else:
                    code(e3)
                    done:
```

Betingede udtryk / conditionals

```
code(e1 ? e2 : e3) = code(e1)
                    jmpf else
                    code(e2)
                    jump done
                    else:
                    code(e3)
                    done:
```

```
code(e1 and e2)    =
```


Betingede udtryk / conditionals

```
code(e1 ? e2 : e3) = code(e1)
                    jmpf else
                    code(e2)
                    jump done
                    else:
                    code(e3)
                    done:

code(e1 and e2)    = code(e1 ? e2 : false)
```


Betingede udtryk / conditionals

```
code(e1 ? e2 : e3) = code(e1)
                    jmpf else
                    code(e2)
                    jump done
                    else:
                    code(e3)
                    done:

code(e1 or e2)      = code(e1 ? true : e2)
                    = code(e1)
                    jmpf else
                    push true
                    jump done
                    else:
                    code(e2)
                    done:
```

codeabby.com



I teams

Canvas:
samarbejdsstrukturer03.pdf

Kommandoer

Syntaks:

```
write e
```

write-kommando

Semantik:

Evaluér e . Udskriv værdien.

Eksempel:

```
write 11 * 4
```

```
let v = e
  b
```

let-kommando

Evaluér e . Gem værdien i variabelen v .
Udfør kommando-blokken b .
Nedlæg variabelen v .

```
let x = read
  write x + 7
  write x * 7
```

Kommandoer

Syntaks:

```
if e
  b1
else
  b2
```

if/else-kommando

```
while e
  b
```

while kommando

Semantik:

Evaluér e . Hvis værdien er sand, så udfør kommando-blokken b_1 , ellers udfør kommando-blokken b_2 .

Evaluér e . Hvis værdien er sand, så udfør kommando-blokken b og gentag.

Eksempel:

```
let n = read
if n < 0
  write -n
else
  write n
```

```
let n = read
let i = 0
while i < n
  write i * i
  set i = i + 1
```

codeabby.com



I teams

Canvas:
samarbejdsstrukturer03.pdf

Hjemmearbejde

Se på Canvas under Plan for modul 03

Reaktioner på i dag

Spørgsmål?

Bekymringer?

Protester?

Kritik?