ERHVERVSAKADEMI
AARHUS

Programmering modul 02

CSIK E2025

I dag

Syntaks for udtryk igen

Semantik af udtryk: evaluering

10+20 minutters pause undervejs

Frokost 12:00

Syntaks for udtryk igen

```
(7 + x) * (8 - y / 3)
```

Syntaks for udtryk igen

```
(7 + x) * (8 – y / 3)
```

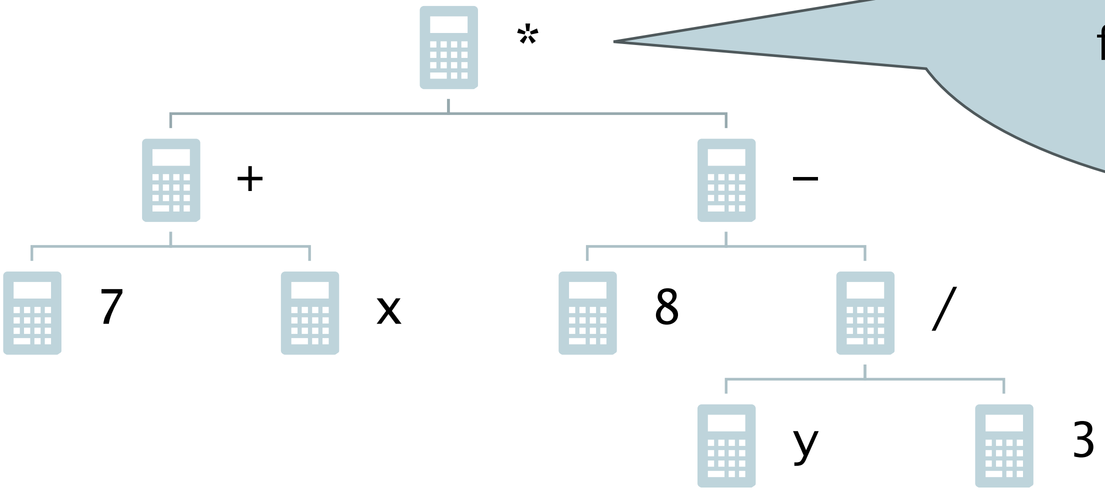
Kode er som udgangspunkt **tekst**

Syntaks for udtryk igen

```
(7 + x) * (8 – y / 3)
```

Kode er som udgangspunkt **tekst**

Programmeringssprogets syntaks-regler fortæller, hvordan teksten skal opfattes som et **træ**

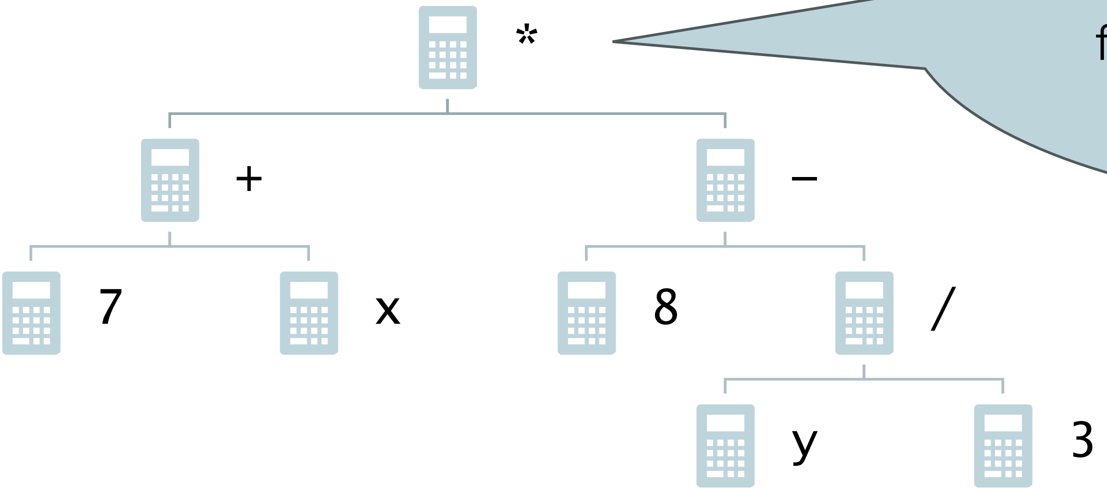


ERHVERVSAKADEMI
AARHUS

Syntaks for udtryk igen

```
(7 + x) * (8 – y / 3)
```

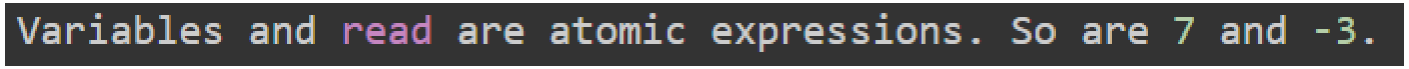
Kode er som udgangspunkt **tekst**



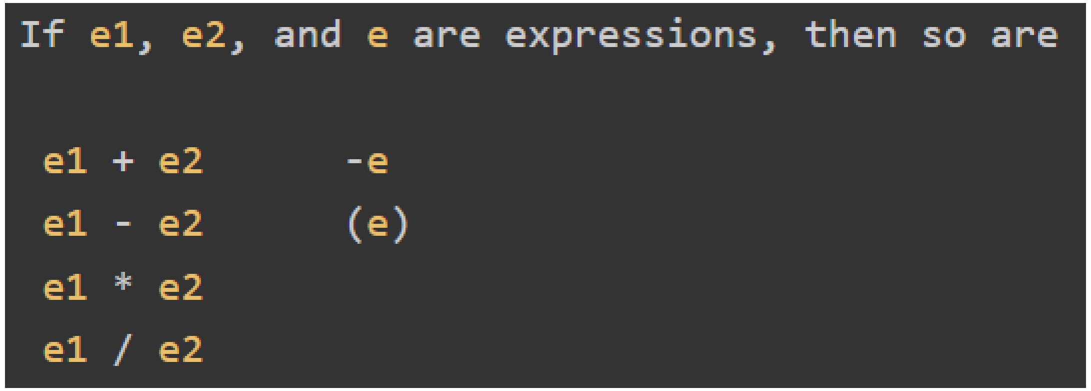
Programmeringssprogets syntaks-regler fortæller, hvordan teksten skal opfattes som et **træ**

Processen, der tager os fra tekst til træ, kaldes **syntaksanalyse** eller **parsing**

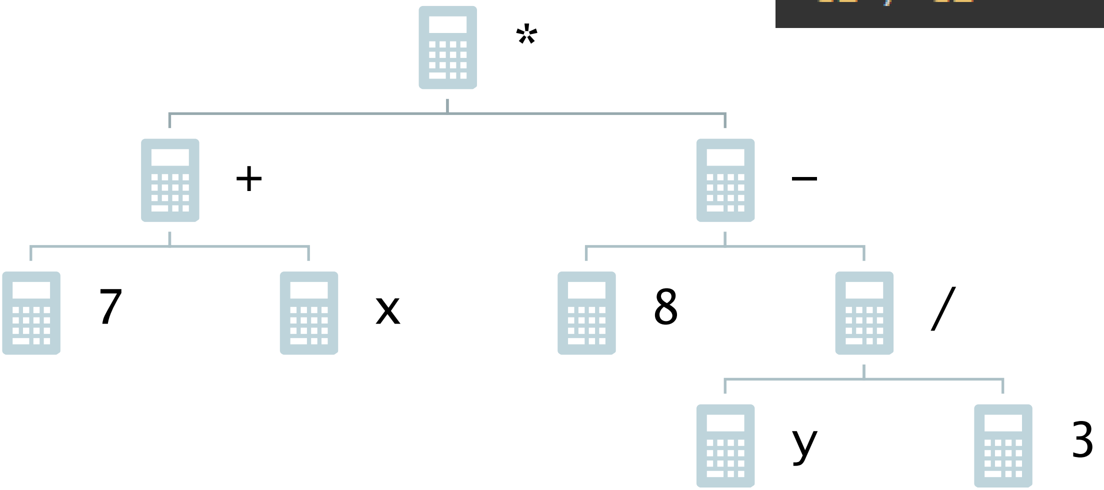
Syntaks for udtryk igen



```
Variables and read are atomic expressions. So are 7 and -3.
```

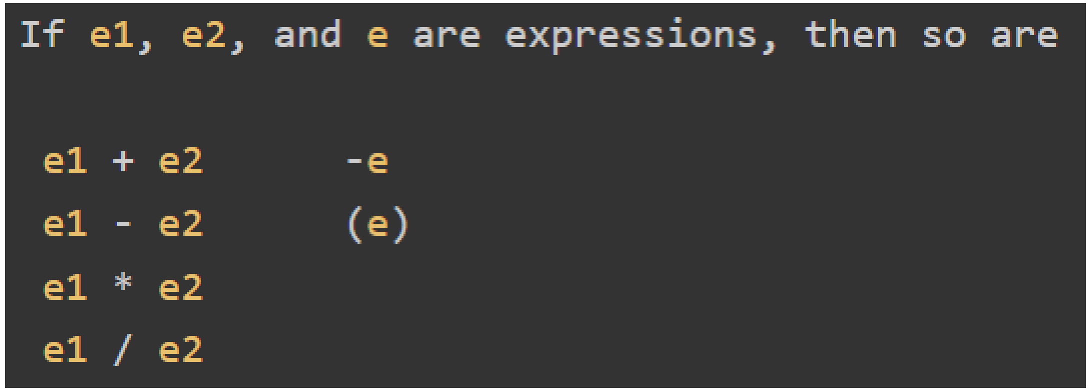


```
If e1, e2, and e are expressions, then so are


 e1 + e2      -e
 e1 - e2      (e)
 e1 * e2
 e1 / e2
```

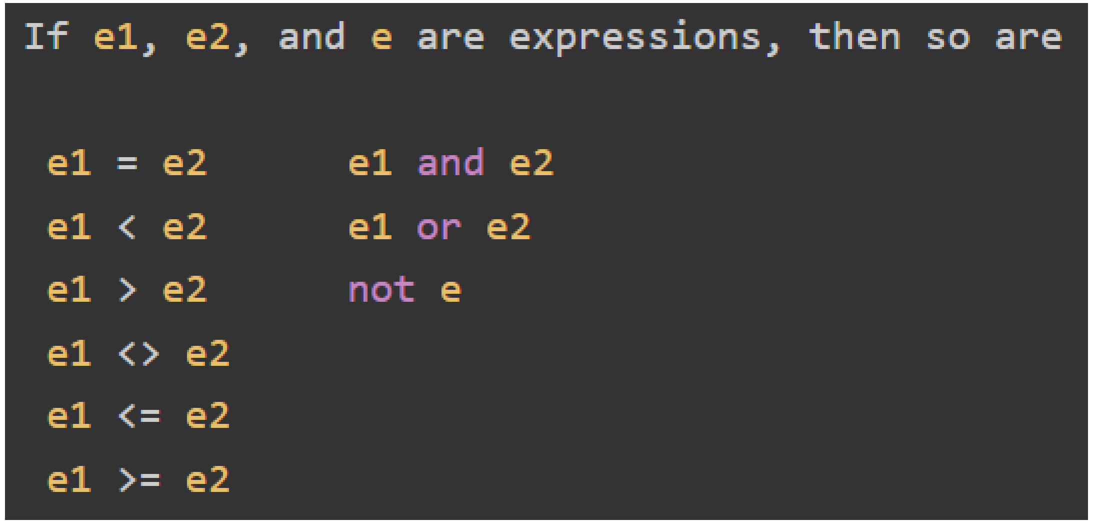


Syntaks for udtryk igen

```
(7 + x) * (8 - y / 3) < b or x % b = 0
```



```
If e1, e2, and e are expressions, then so are


 e1 + e2       -e
 e1 - e2       (e)
 e1 * e2
 e1 / e2
```



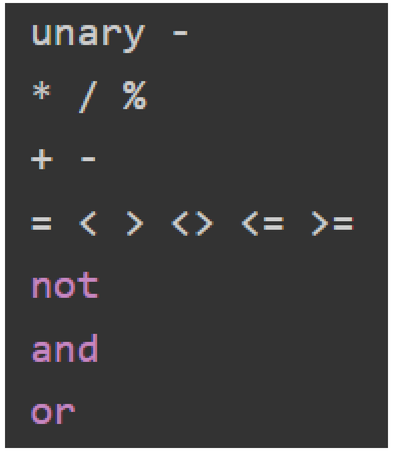
```
If e1, e2, and e are expressions, then so are


 e1 = e2       e1 and e2
 e1 < e2       e1 or e2
 e1 > e2       not e
 e1 <> e2
 e1 <= e2
 e1 >= e2
```

Syntaks for udtryk igen

```
(7 + x) * (8 - y / 3) < b or x % b = 0
```

```
(((7 + x) * (8 - (y / 3))) < b) or ((x % b) = 0)
```

Programmeringssprogets præcedens-regler giver os mulighed for at udelade mange parenteser

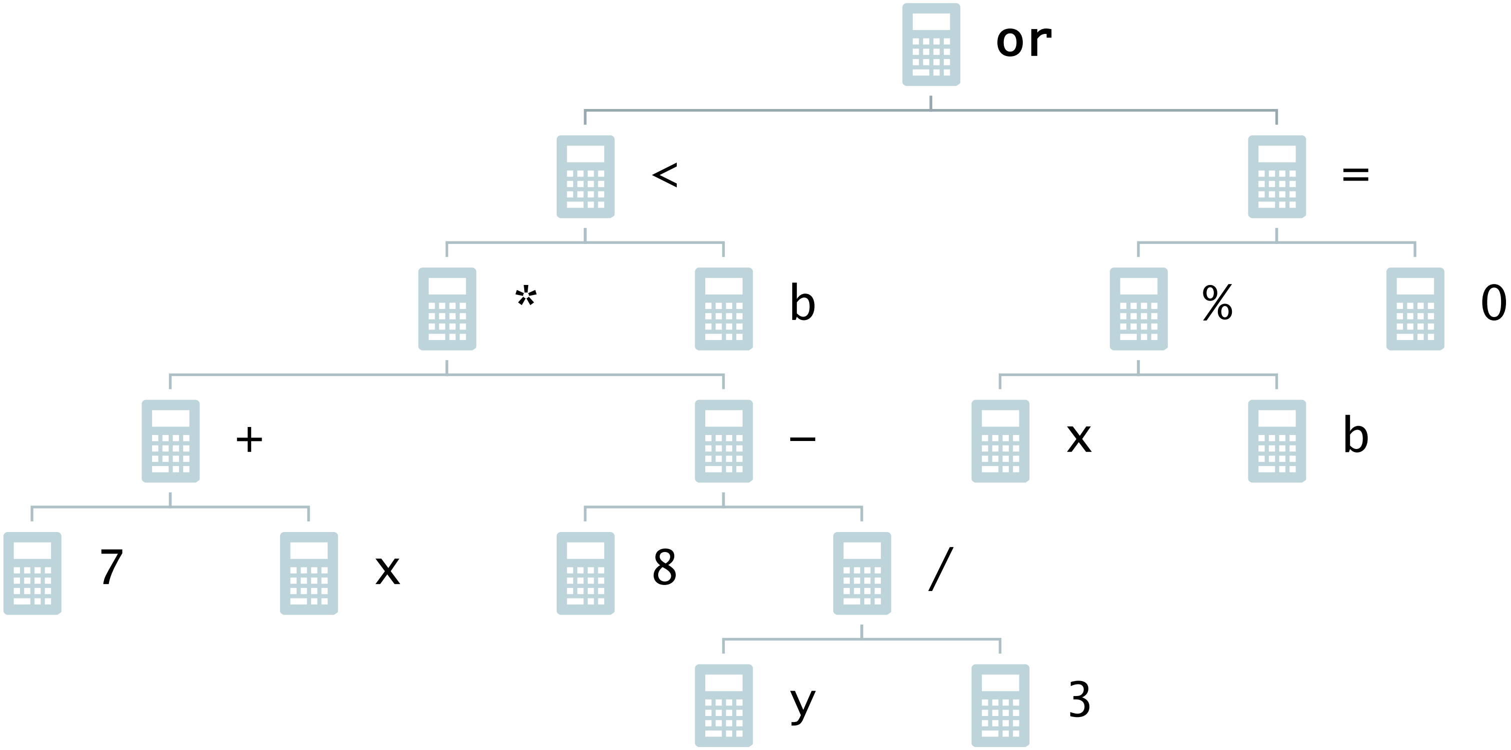


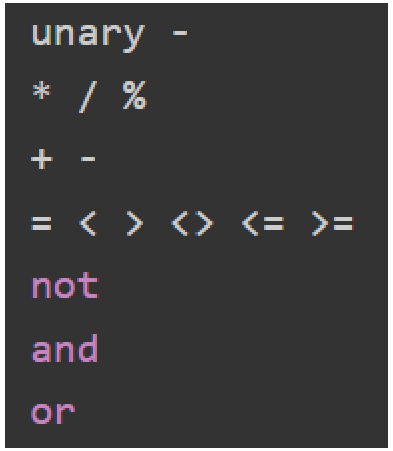
```
unary -
* / %
+ -
= < > <> <= >=
not
and
or
```

Syntaks for udtryk igen

```
(7 + x) * (8 - y / 3) < b or x % b = 0
(((7 + x) * (8 - (y / 3))) < b) or ((x % b) = 0)
```

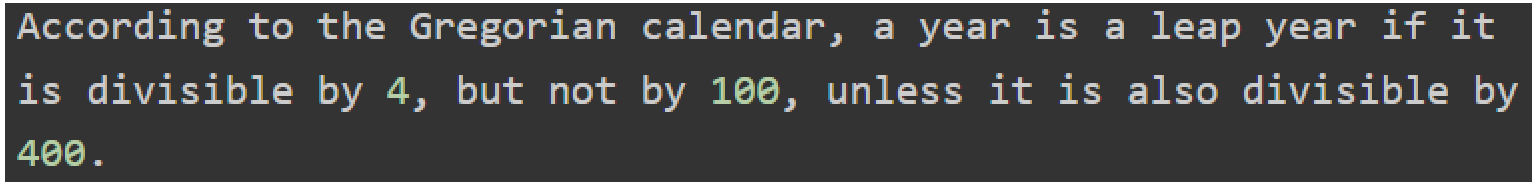
Programmeringssprogets præcedens-regler giver os mulighed for at udelade mange parenteser



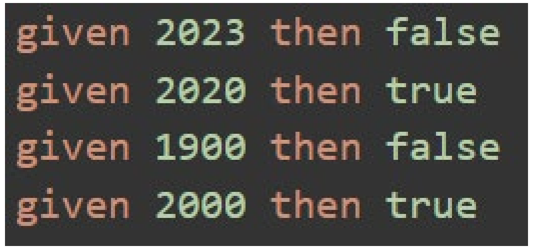


```
unary -
* / %
+ -
= < > <> <= >=
not
and
or
```

Syntaks for udtryk igen

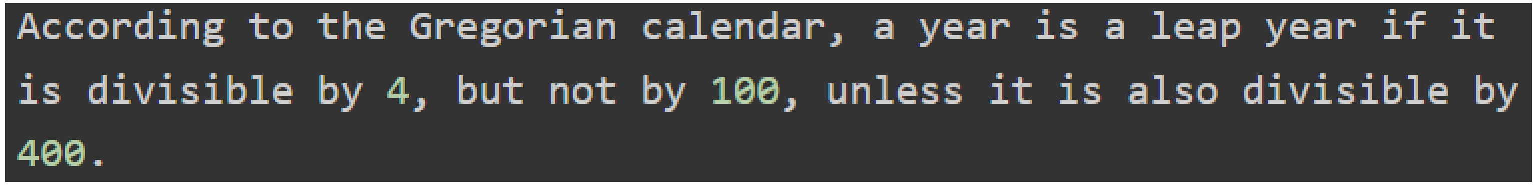


```
According to the Gregorian calendar, a year is a leap year if it
is divisible by 4, but not by 100, unless it is also divisible by
400.
```

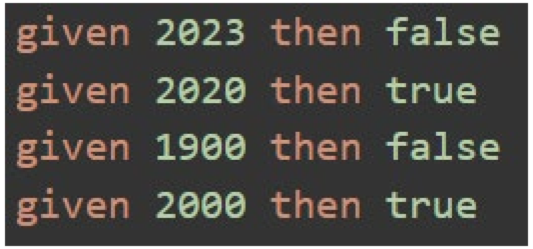


```
given 2023 then false
given 2020 then true
given 1900 then false
given 2000 then true
```

Syntaks for udtryk igen



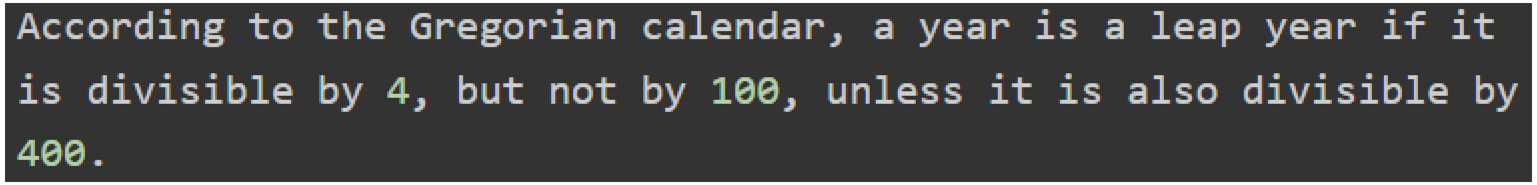
```
According to the Gregorian calendar, a year is a leap year if it
is divisible by 4, but not by 100, unless it is also divisible by
400.
```



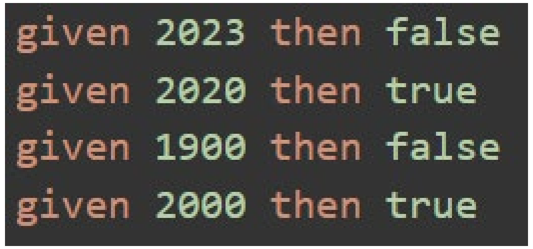
```
given 2023 then false
given 2020 then true
given 1900 then false
given 2000 then true
```

```
year % 4 = 0 and not year % 100 = 0 or year % 400 = 0
```

Syntaks for udtryk igen



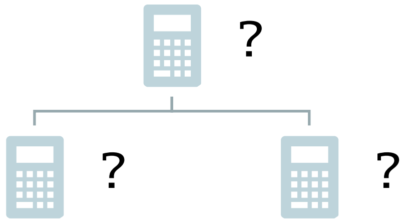
```
According to the Gregorian calendar, a year is a leap year if it
is divisible by 4, but not by 100, unless it is also divisible by
400.
```

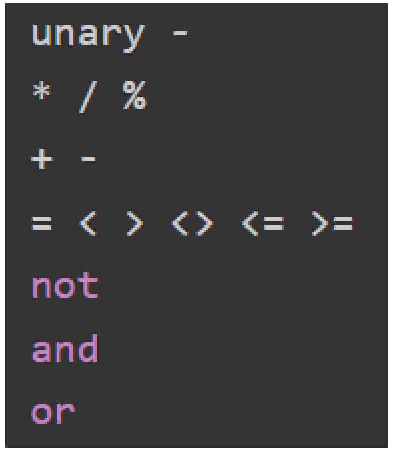


```
given 2023 then false
given 2020 then true
given 1900 then false
given 2000 then true
```

```
year % 4 = 0 and not year % 100 = 0 or year % 400 = 0
```

Med skuldermakker:
Sæt parenteserne,
tegn træet!





```
unary -
* / %
+ -
= < > <> <= >=
not
and
or
```

Syntaks for udtryk: opsamling

Hvor er du mest forvirret?

compound expression
output
programming
semantics
modulo
logical operator
atomic expression
statement
operator
literal
comment
text editor
relational
imperative programming
evaluation
short-circuiting evaluation
binary operator
syntax highlighting
code
parsing
expression
operator precedence
side effect
operator associativity
programming language
automated tests
high-level
variable
console
input
syntax
Booleans
program
keyword
unary operator

Fra syntaks til semantik

Et udtryk beskriver en værdi (et tal, en sandhedsværdi, en farve, en dato, …)

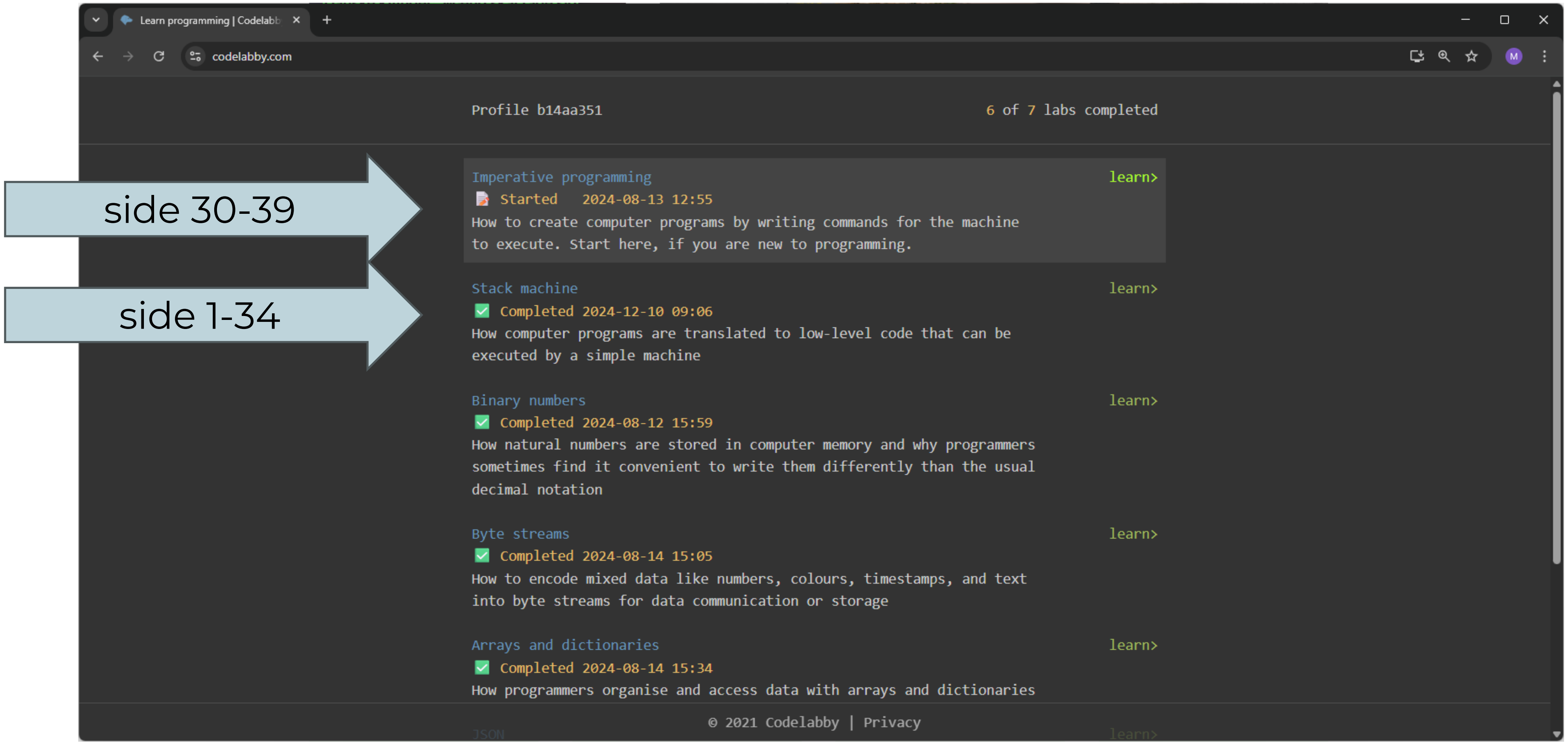
Programmeringssprogets semantik omfatter regler for, hvordan den værdi beregnes.

Beregningsprocessen kaldes **evaluering**.

I dag:

- I skal læse og forstå reglerne ("Imperative programming" side 30-39)
- I skal prøve at følge reglerne ved at håndoversætte udtryk ("Stack machine" side 1-34)

codelabby.com



Profile b14aa351

6 of 7 labs completed

Imperative programming learn>
📝 Started 2024-08-13 12:55
How to create computer programs by writing commands for the machine to execute. Start here, if you are new to programming.

Stack machine learn>
✅ Completed 2024-12-10 09:06
How computer programs are translated to low-level code that can be executed by a simple machine

Binary numbers learn>
✅ Completed 2024-08-12 15:59
How natural numbers are stored in computer memory and why programmers sometimes find it convenient to write them differently than the usual decimal notation

Byte streams learn>
✅ Completed 2024-08-14 15:05
How to encode mixed data like numbers, colours, timestamps, and text into byte streams for data communication or storage

Arrays and dictionaries learn>
✅ Completed 2024-08-14 15:34
How programmers organise and access data with arrays and dictionaries

© 2021 Codelabby | Privacy

JSON learn>

side 30-39

side 1-34

codelabby.com

I teams og med skuldermakker

Canvas:

samarbejdsstrukturer02.pdf

Hjemmearbejde

Se på Canvas under Plan for modul 02

Reaktioner på i dag

Spørgsmål?

Bekymringer?

Protester?

Kritik?