Introduktion til Programmering og Problemløsning (PoP)

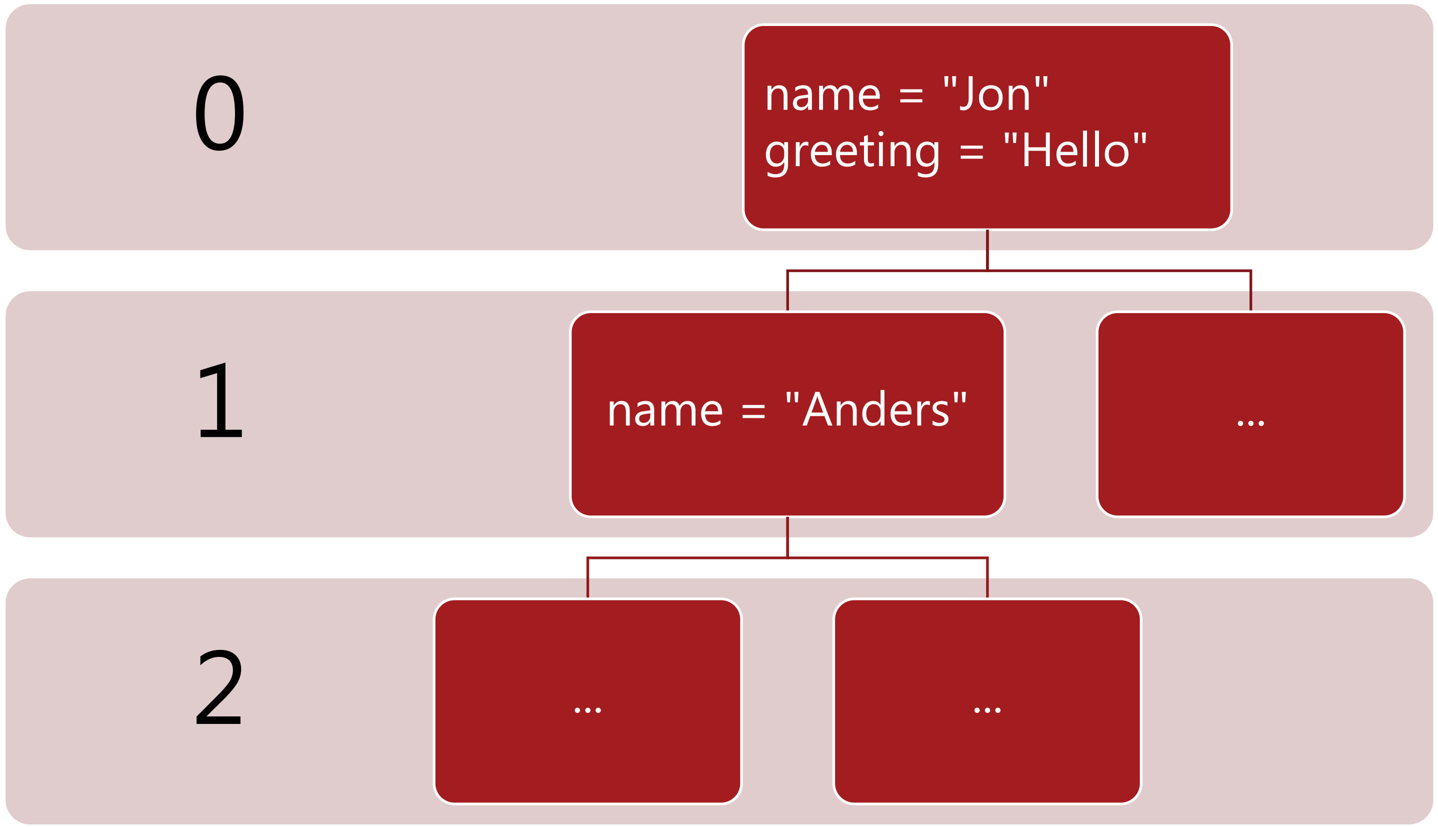
Jon Sporring
Department of Computer Science
2020/09/07

UNIVERSITY OF COPENHAGEN

Virkefelter (scope)

Virkefelter via parenteser

```
let greeting = "Hello"                          0
let name = "Jon"
do printfn "%s %s" greeting name
(
  let name = "Anders"                         1
  do printfn "%s %s" greeting name
)
do printfn "%s %s" greeting name
```



Navne (i yderste virkefelt) kan ikke overskrives

~~let name = "World"~~

~~let name = "Jon"~~

~~do printfn "Hello %s" name~~

Syntaks og virkefelter

Letvægtssyntaks

```
let name = "World"
do printfn "Hello %A" name
```

Valgfrit 'do'

```
let name = "World"
printfn "Hello %A" name
```

verbose syntaks

```
let name = "World" in do printfn "Hello %A" name
```

Funktioner

Organisering = nemmere at forstå og vedligeholde

Leksikografisk virkefelt

```
let greetings (name : string) : string =
  "Hello " + name
```

Indryk angiver funktionskroppen

```
let str = greetings "Jon"
printfn "%s" str
printfn "%s" (greetings "World")
```

```
> let greetings (name : string) : string =
-   "Hello " + name;;
val greetings : name:string -> string
```

```
let greetings name =
  "Hello " + name
```

```
let greetings name = "Hello " + name
```

```
let greetings name : string = "Hello " + name
```

```
let greetings (name : string) = "Hello " + name
```

Løs en andengradsligning (baglæns!)

```
let discriminant a b c =
  b ** 2.0 - 4.0 * a * c

let solution a b c sgn =
  let d = discriminant a b c
  (-b + sgn * sqrt d) / (2.0 * a)

let a = 1.0
let b = 0.0
let c = -1.0
let xp = (solution a b c +1.0)
printfn "0 = %fx^2 + %fx + %f => x_+ = %f" a b c xp
```

$$ax^2 + bx + c = 0$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Resumé

I denne video hørte du om:

- Letvægts og verbose syntaks
- Virkefelter
- Funktioner