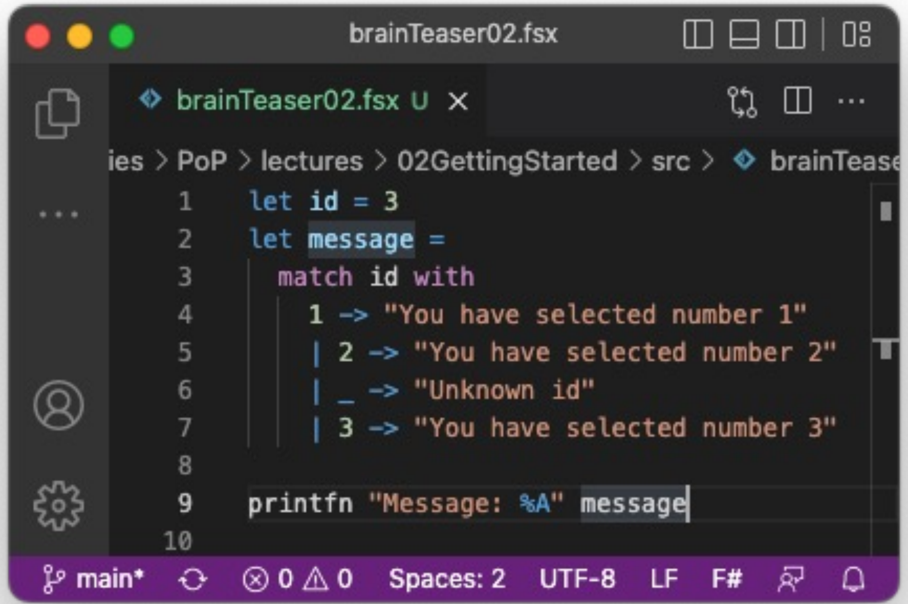
Introduktion til Programmering og Problemløsning (PoP)

Håndkøring

Jon Sporring
Department of Computer Science
2022/09/12

UNIVERSITY OF COPENHAGEN

Hvad skriver programmet



```
let id = 3
let message =
  match id with
    1 -> "You have selected number 1"
    | 2 -> "You have selected number 2"
    | _ -> "Unknown id"
    | 3 -> "You have selected number 3"

printfn "Message: %A" message
```

- ○ "You have selected number 2"
- ○ "You have selected number 1"
- ○ "You have selected number 3"
- ○ "Unknown id"
- ○ Andet

```
% dotnet fsi brainTeaser02.fsx
/Users/jrh630/repositories/PoP/lectures/02GettingStarted/src/brainTeaser02.fsx(7,7): warning
FS0026: This rule will never be matched

Message: "Unknown id"
```

Funktioner

Organisering = nemmere at forstå og vedligeholde

Leksikografisk virkefelt

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

Indryk angiver funktionskroppen

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

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

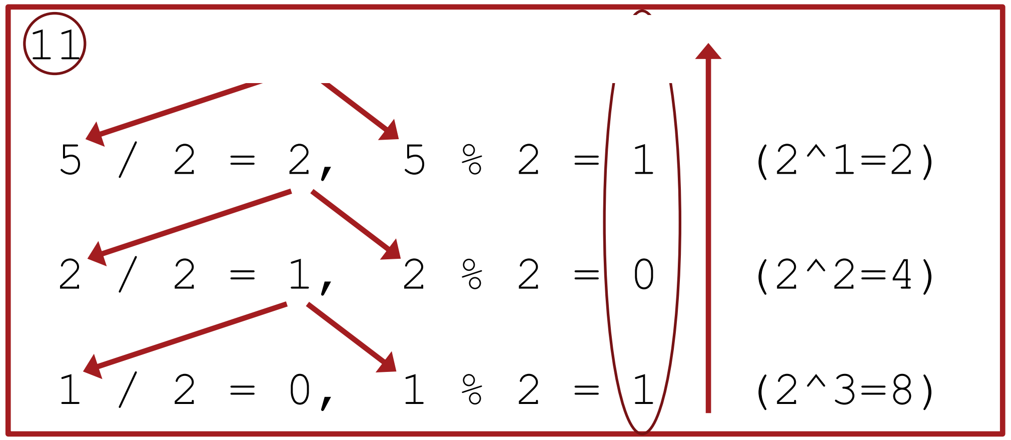
```
let greetings name =
  "Hello " + name
```

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

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

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

Decimal til Binær: Divider med 2

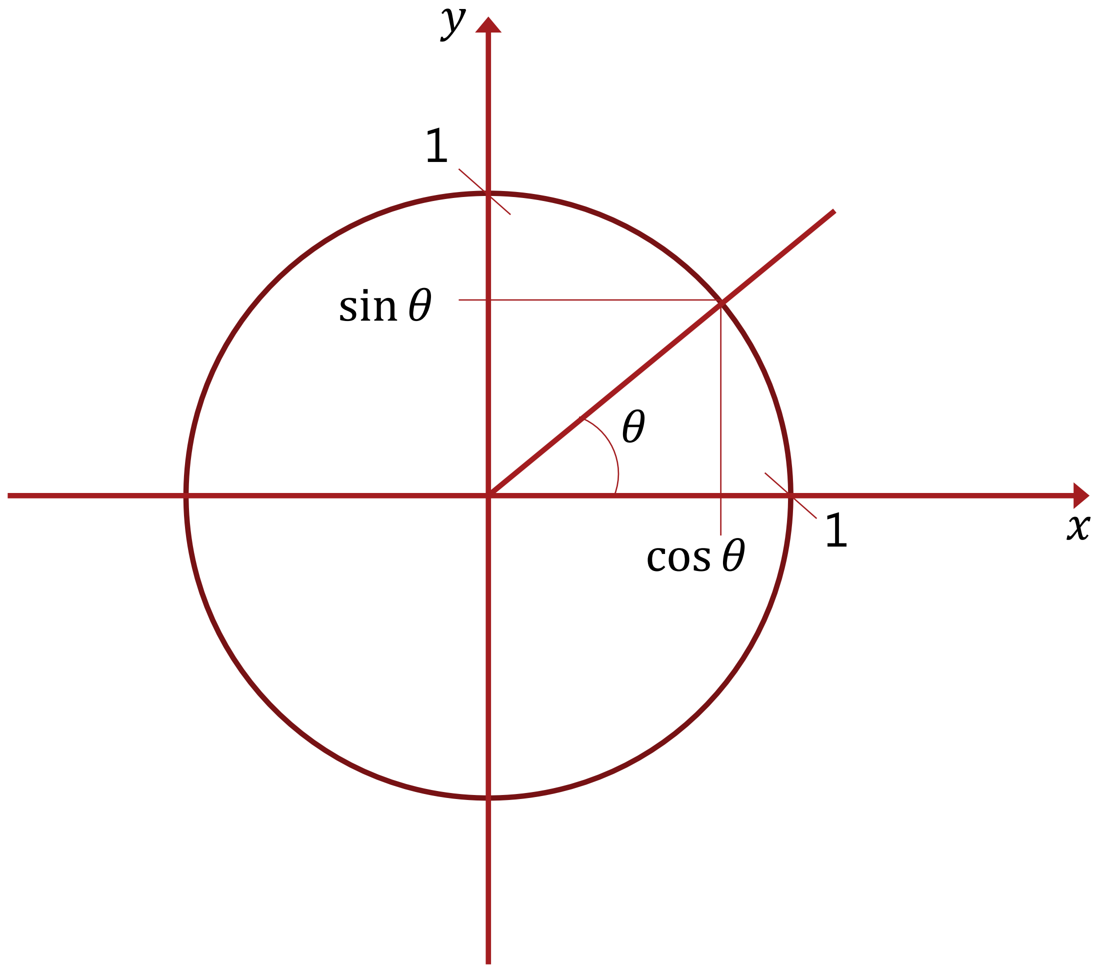


```
let rec divideByTwo (n: uint) : string =
  match n with
    0u -> ""
    | _ -> (divideByTwo (n/2u)) + (string (n%2u))

let N = 11u
let str = divideByTwo N
printfn "%A_10 = %A_2" N str
```

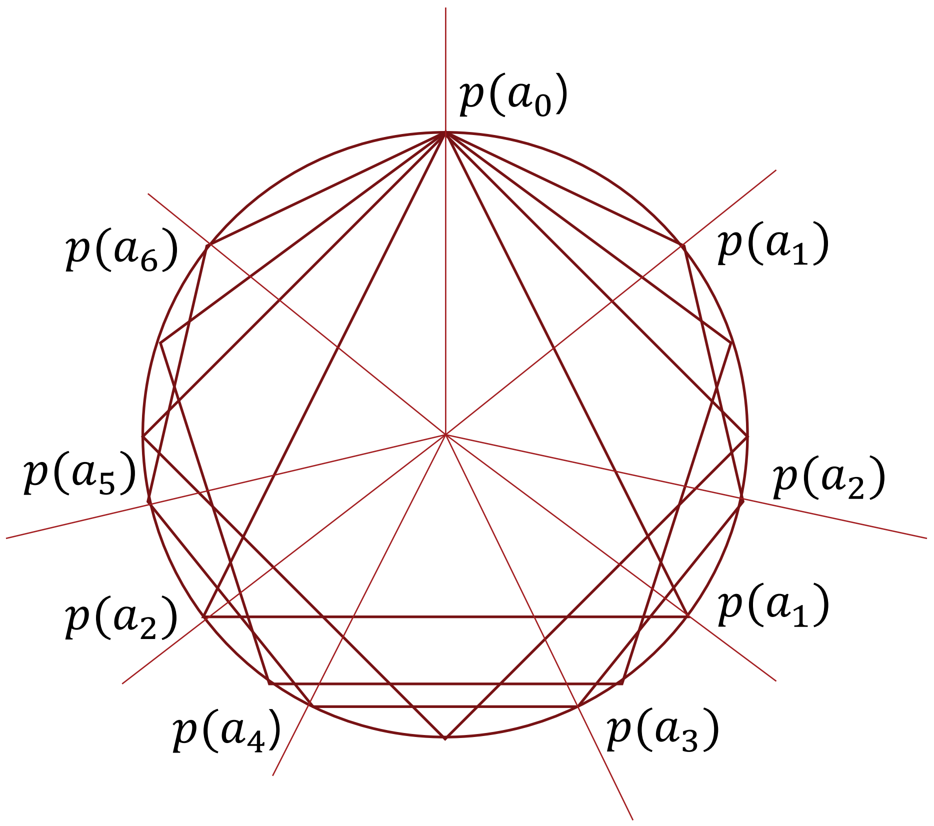
```
let divideByTwo (n : uint) : string =
  let mutable i = n
  let mutable str = ""
  while i > 0u do
    str <- string (i % 2u) + str
    i <- i / 2u
  str

let N = 11u
let str = divideByTwo N
printfn "%A_10 = %A_2" N str
```

Dec	Bin
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111
16	10000
17	10001
18	10010
19	10011
20	10100
21	10101
22	10110
23	10111
24	11000
25	11001
26	11010
27	11011
28	11100
29	11101
30	11110
31	11111

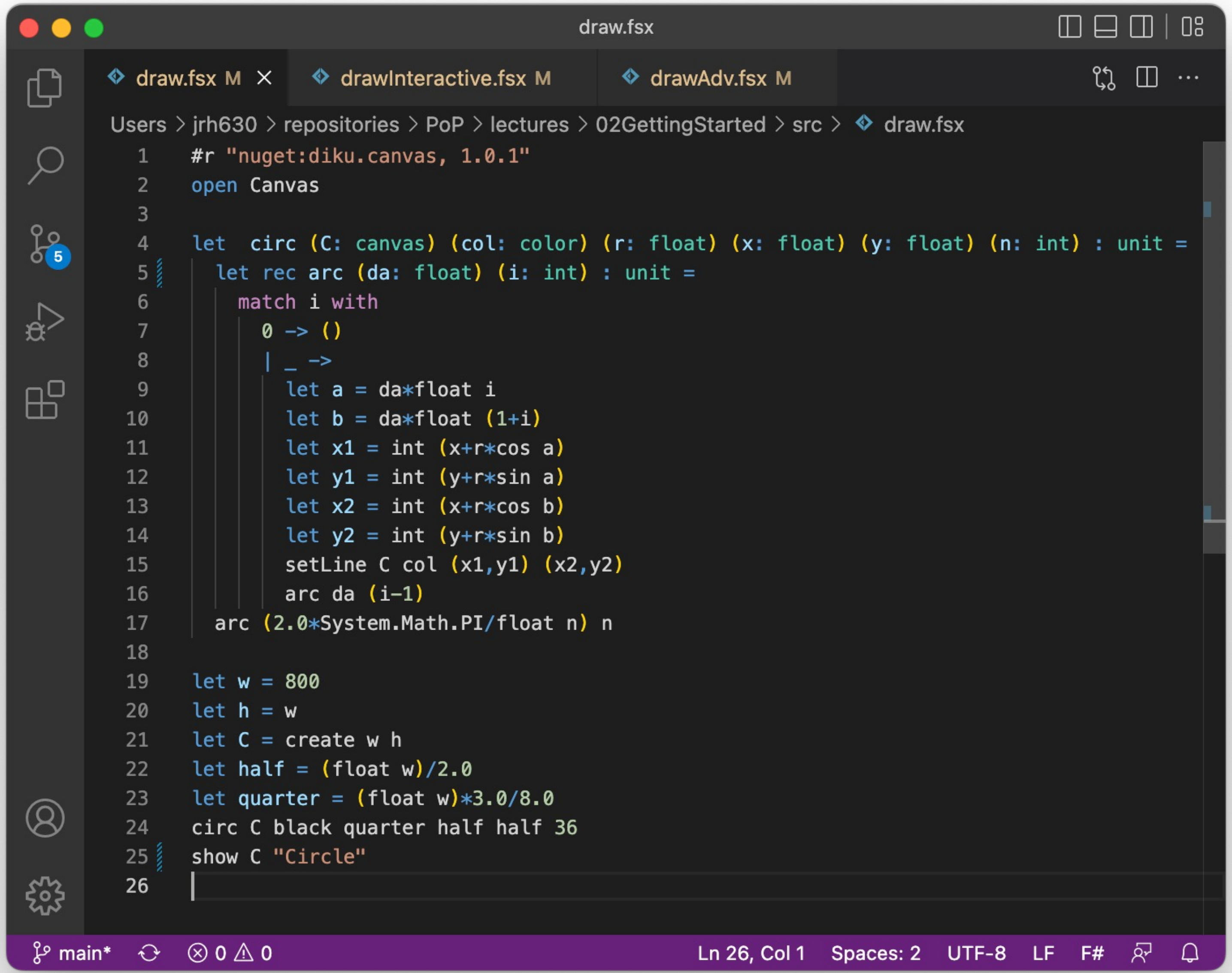
Canvas: tegn en cirkel med rette linjestykker



$$p(a) = (\mathrm{r}\cos a,\ r\sin a)$$



Canvas: tegn en cirkel med rette linjestykker



```fsharp
#r "nuget:diku.canvas, 1.0.1"
open Canvas

let  circ (C: canvas) (col: color) (r: float) (x: float) (y: float) (n: int) : unit =
  let rec arc (da: float) (i: int) : unit =
    match i with
      0 -> ()
      | _ ->
        let a = da*float i
        let b = da*float (1+i)
        let x1 = int (x+r*cos a)
        let y1 = int (y+r*sin a)
        let x2 = int (x+r*cos b)
        let y2 = int (y+r*sin b)
        setLine C col (x1,y1) (x2,y2)
        arc da (i-1)
  arc (2.0*System.Math.PI/float n) n

let w = 800
let h = w
let C = create w h
let half = (float w)/2.0
let quarter = (float w)*3.0/8.0
circ C black quarter half half 36
show C "Circle"
```

Canvas: tegn en interaktiv cirkel

Interactive Canvas

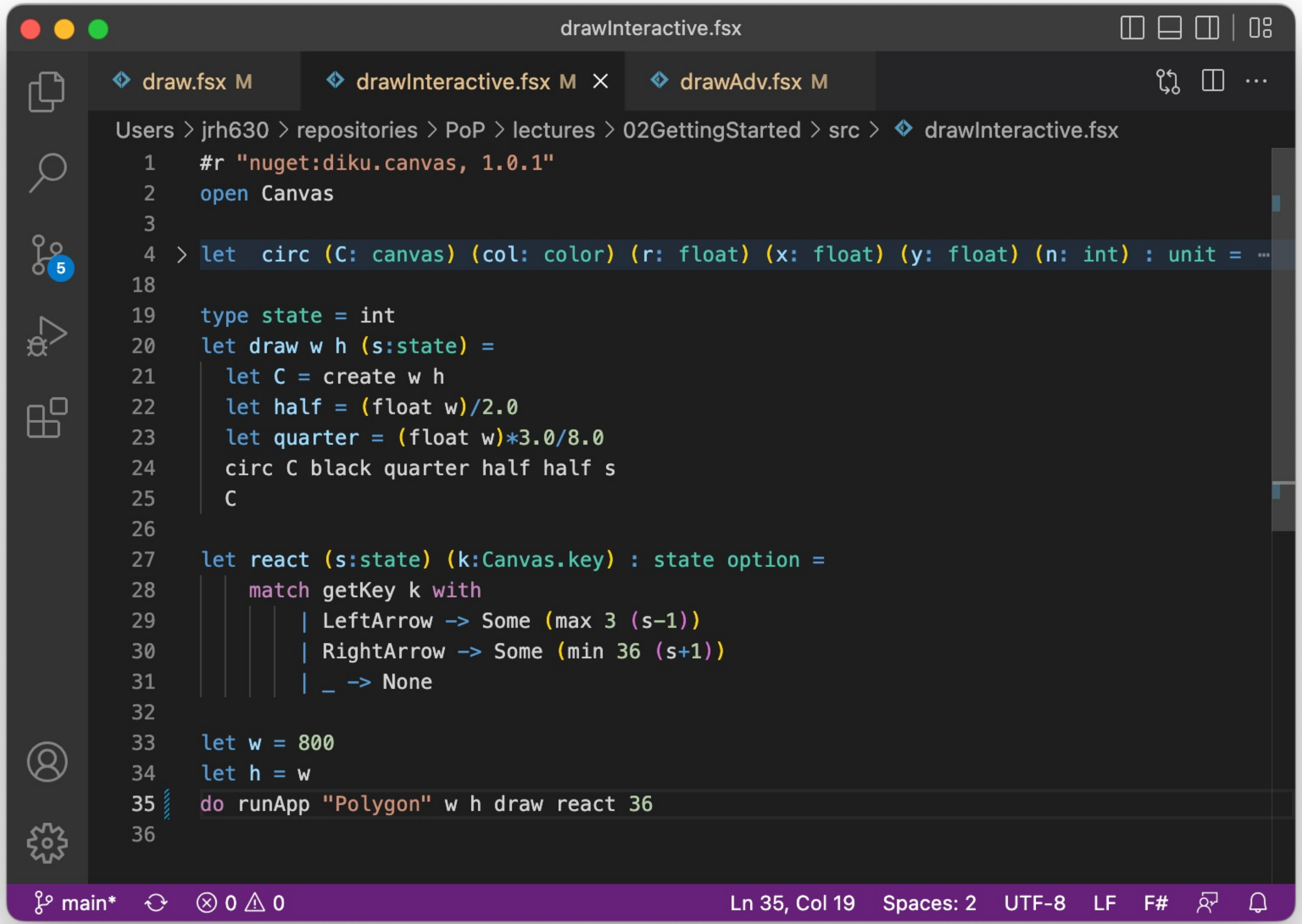
```
val runApp     : string -> int -> int
             -> (int -> int -> 's -> canvas)
             -> ('s -> key -> 's option)
             -> 's -> unit
```

Eksempel

```
type state = int
let draw (w: int) (h: int) (s: state) = …
let react (s: state) (k: Canvas.key) = …
do runApp "Text" 300 300 draw react 0
```

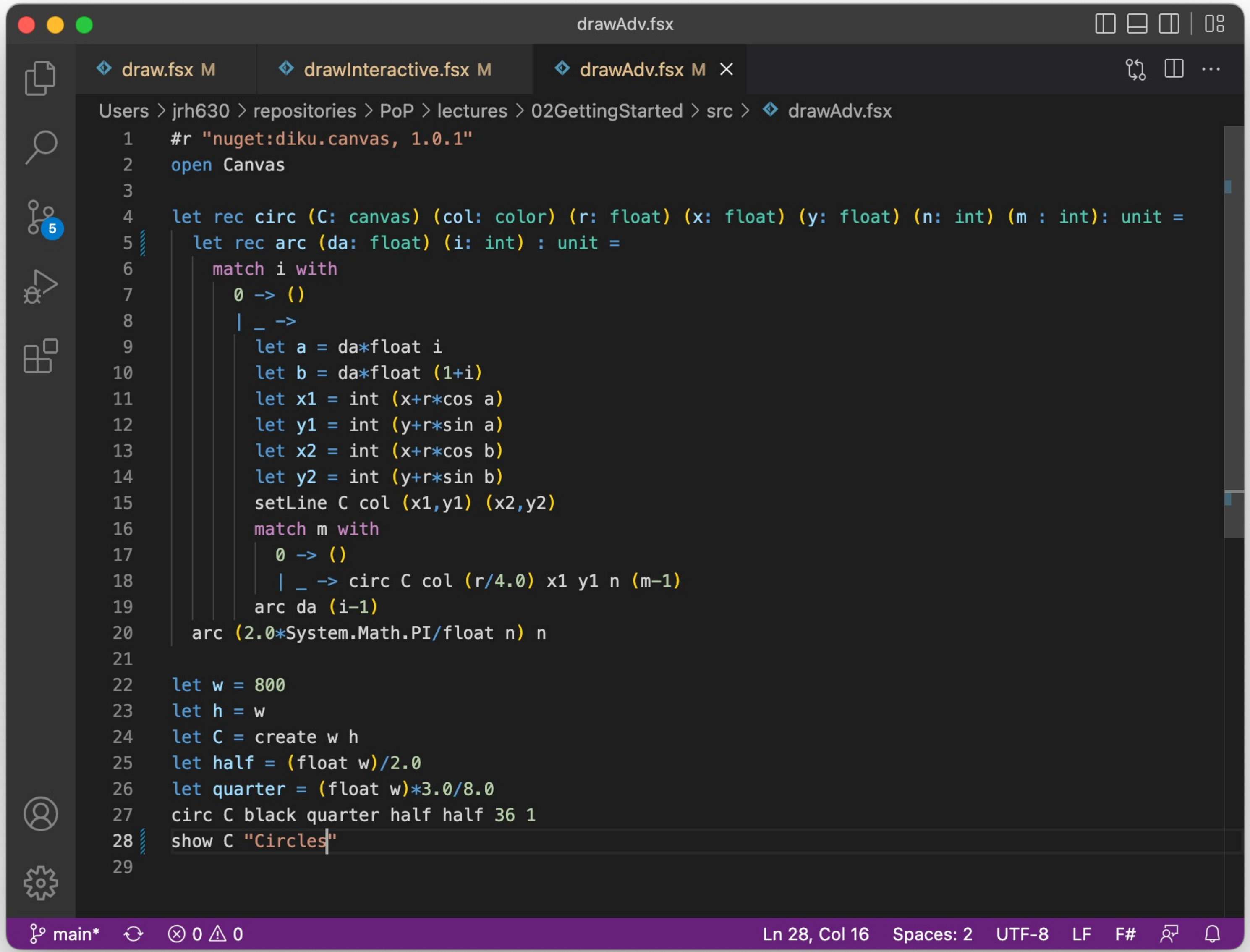
Hvad gør runApp?

```
let runApp txt w h draw react init =
  let mutable s = init
  draw w h s

  while true do
    let k = userKeyPress ()
    s <- react s k
    draw w h s
```



```
#r "nuget:diku.canvas, 1.0.1"
open Canvas

let  circ (C: canvas) (col: color) (r: float) (x: float) (y: float) (n: int) : unit = ⋯

type state = int
let draw w h (s:state) =
  let C = create w h
  let half = (float w)/2.0
  let quarter = (float w)*3.0/8.0
  circ C black quarter half half s
  C

let react (s:state) (k:Canvas.key) : state option =
    match getKey k with
        | LeftArrow -> Some (max 3 (s-1))
        | RightArrow -> Some (min 36 (s+1))
        | _ -> None

let w = 800
let h = w
do runApp "Polygon" w h draw react 36
```

Canvas: tegn en cirkel rekursivt



drawAdv.fsx

draw.fsx M | drawInteractive.fsx M | drawAdv.fsx M

Users > jrh630 > repositories > PoP > lectures > 02GettingStarted > src > drawAdv.fsx

```
#r "nuget:diku.canvas, 1.0.1"
open Canvas

let rec circ (C: canvas) (col: color) (r: float) (x: float) (y: float) (n: int) (m : int): unit =
  let rec arc (da: float) (i: int) : unit =
    match i with
      0 -> ()
      | _ ->
        let a = da*float i
        let b = da*float (1+i)
        let x1 = int (x+r*cos a)
        let y1 = int (y+r*sin a)
        let x2 = int (x+r*cos b)
        let y2 = int (y+r*sin b)
        setLine C col (x1,y1) (x2,y2)
        match m with
          0 -> ()
          | _ -> circ C col (r/4.0) x1 y1 n (m-1)
        arc da (i-1)
  arc (2.0*System.Math.PI/float n) n

let w = 800
let h = w
let C = create w h
let half = (float w)/2.0
let quarter = (float w)*3.0/8.0
circ C black quarter half half 36 1
show C "Circles"

```

main* ⊗ 0 ⚠ 0 Ln 28, Col 16 Spaces: 2 UTF-8 LF F#

Resumé

I dag har vi talt om:

- Match-with og wildcard
- Funktioner
- Dividér med 2 algoritmen på funktionel og imperativ form
- Canvas cirkeltegning og interaktion