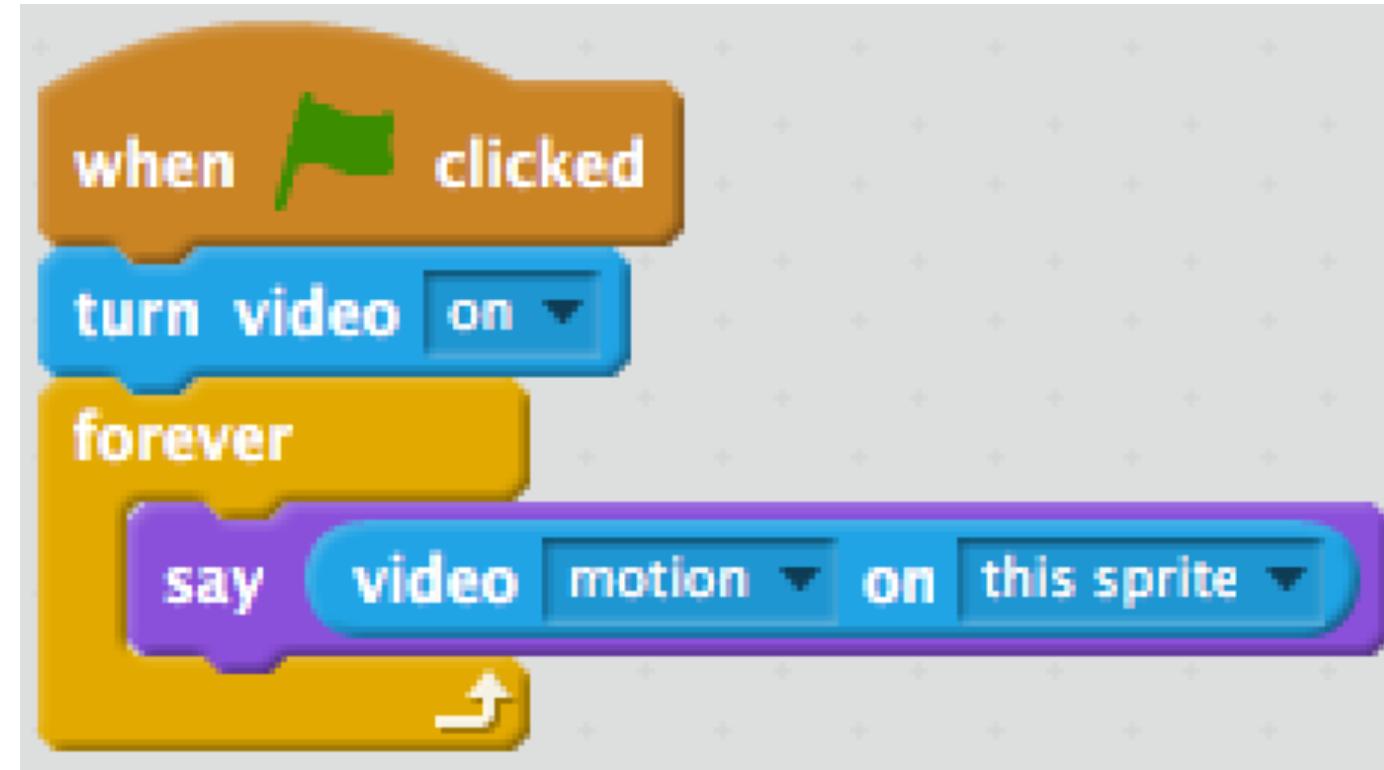


Programmering og Problemløsning

2.1: Fsharp

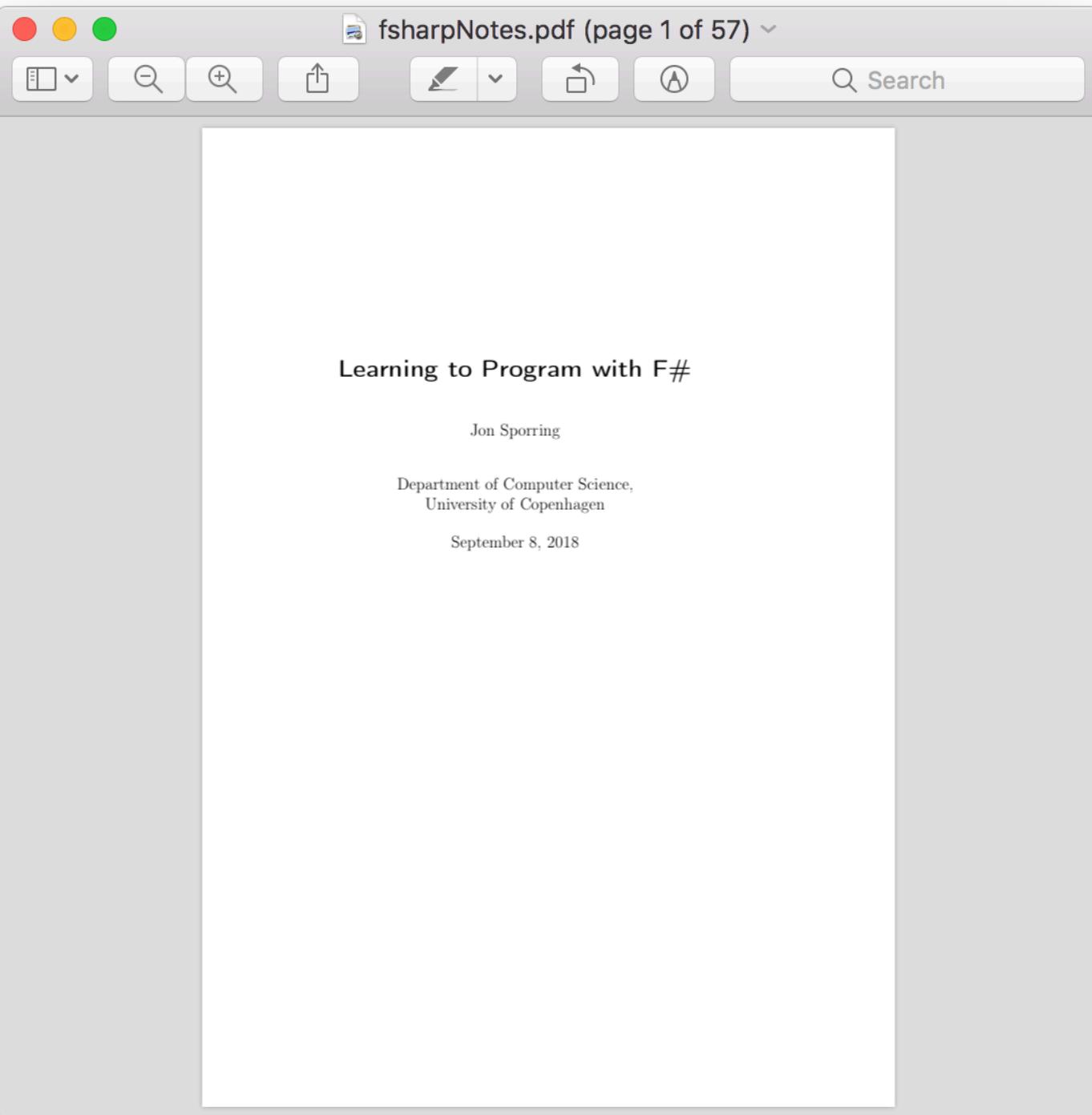
Nøglekoncepter

- Initial starttilstand
 - Løkker
 - Betingelser
 - Udtryk
 - Værdi
 - Variabel
 - Kommunikation
 - Tilfældigt tal
 - Initial starttilstand
-
- Udforsk blokke med testprogrammer
 - Design: tænk først, derefter programmér
 - Fejlfinding og retning (debugging)



Trin i problemløsning

- Forstå problemet
- Tænk løsninger igennem (design)
- Vælg og implementér
- Reflektér og evt. gentag



Programmeringsparadigmer

Imperativ



Objektorienteret



Deklarativ



Event-drevet



F# som lommeregner

357

```
let a = 357
```

864

```
let b = 864
```

357+864

```
let c = a + b
```

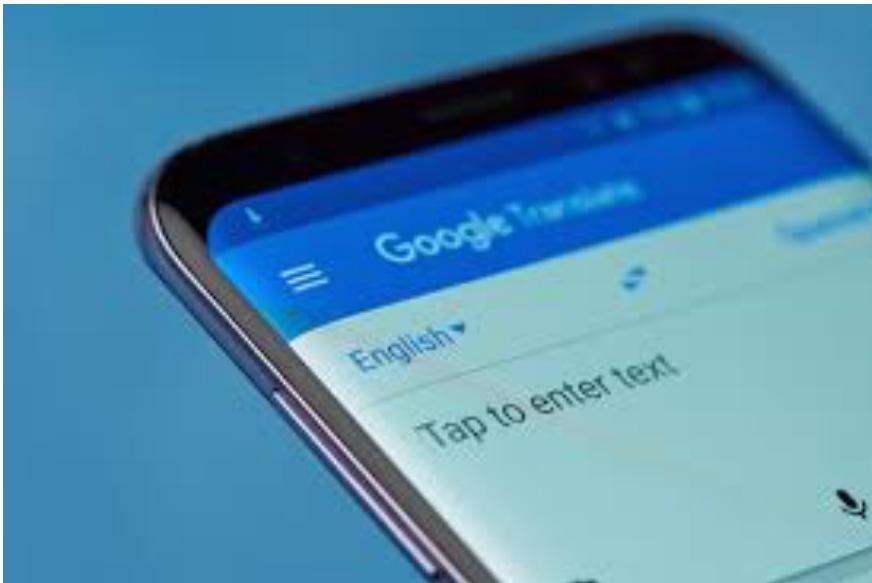
```
do printfn "%A" c
```

3 måder at køre (execute / run) programmet på:

- fsharpi -> indtast myFirstFsharp.fsx
- fsharpi myFirstFsharp.fsx
- fsharpc myFirstFsharp.fsx && mono myFirstFsharp.exe

Fortolker (interpreter) vs. Oversætter (compiler)

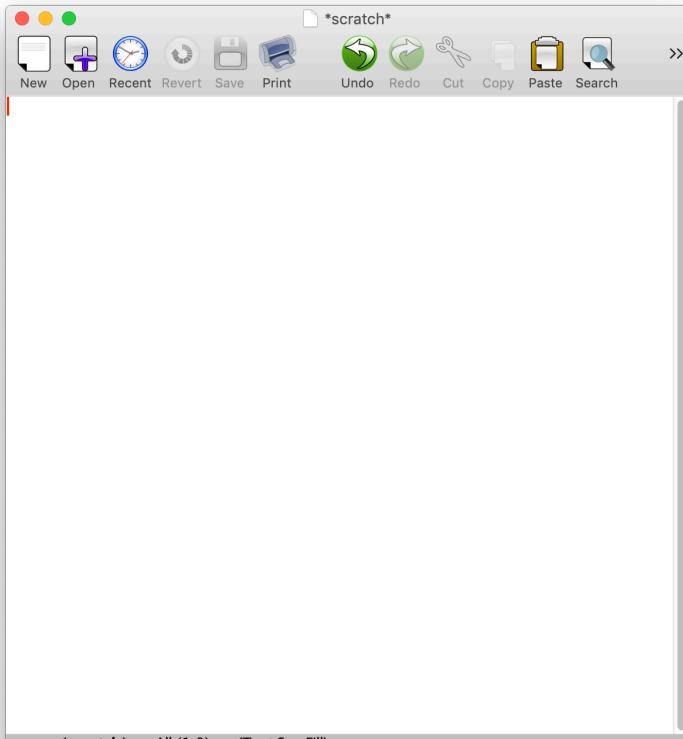
Interaktion



Kompilering



F#: Teksteditor, kompilering, kørsel

A screenshot of a terminal window titled '2.1ScratchNFsharp — bash — 80x24'. The window displays the following text:

```
[Jons-mac:2.1ScratchNFsharp sporring$ fsharpc gettingStartedStump.fsx
Microsoft (R) F# Compiler version 10.2.3 for F# 4.5
Copyright (c) Microsoft Corporation. All Rights Reserved.
[Jons-mac:2.1ScratchNFsharp sporring$ mono gettingStartedStump.exe
3
Jons-mac:2.1ScratchNFsharp sporring$ ]
```

Listing 3.1 `gettingStartedStump.fsx`:
A simple demonstration script.

```
1 let a = 3.0
2 do printfn "%g" a
```

F#

Listing 3.1 gettingStartedStump.fsx:
A simple demonstration script.

```
1 let a = 3.0
2 do printfn "%g" a
```

Command	Time
fsharpi gettingStartedStump.fsx	1.88s
fsharpc gettingStartedStump.fsx	1.90s
mono gettingStartedStump.exe	0.05s

Alt på computeren er relateret til binære tal

$$v = \sum_{i=0}^n b_i 2^i$$

<https://tinyurl.com/ycpxcto5>

Et heksal ciffer

Et oktal ciffer

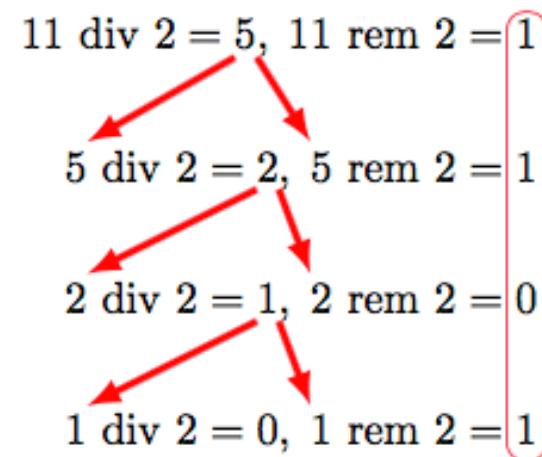
Dec	Bin	Oct	Hex	Dec	Bin	Oct	Hex
0	0	0	0	32	100000	40	20
1	1	1	1	33	100001	41	21
2	10	2	2	34	100010	42	22
3	11	3	3	35	100011	43	23
4	100	4	4	36	100100	44	24
5	101	5	5	37	100101	45	25
6	110	6	6	38	100110	46	26
7	111	7	7	39	100111	47	27
8	1000	10	8	40	101000	50	28
9	1001	11	9	41	101001	51	29
10	1010	12	a	42	101010	52	2a
11	1011	13	b	43	101011	53	2b
12	1100	14	c	44	101100	54	2c
13	1101	15	d	45	101101	55	2d
14	1110	16	e	46	101110	56	2e
15	1111	17	f	47	101111	57	2f
16	10000	20	10	48	110000	60	30
17	10001	21	11	49	110001	61	31
18	10010	22	12	50	110010	62	32
19	10011	23	13	51	110011	63	33
20	10100	24	14	52	110100	64	34
21	10101	25	15	53	110101	65	35
22	10110	26	16	54	110110	66	36
23	10111	27	17	55	110111	67	37
24	11000	30	18	56	111000	70	38
25	11001	31	19	57	111001	71	39
26	11010	32	1a	58	111010	72	3a
27	11011	33	1b	59	111011	73	3b
28	11100	34	1c	60	111100	74	3c
29	11101	35	1d	61	111101	75	3d
30	11110	36	1e	62	111110	76	3e
31	11111	37	1f	63	111111	77	3f

Dividér med 2 algoritmen

Fra binær til decimal

$$v = \sum_{i=0}^n b_i 2^i$$

Fra decimal til binær



<https://tinyurl.com/y7s5979a>