

Introduktion til Programmering og Problemløsning (PoP)

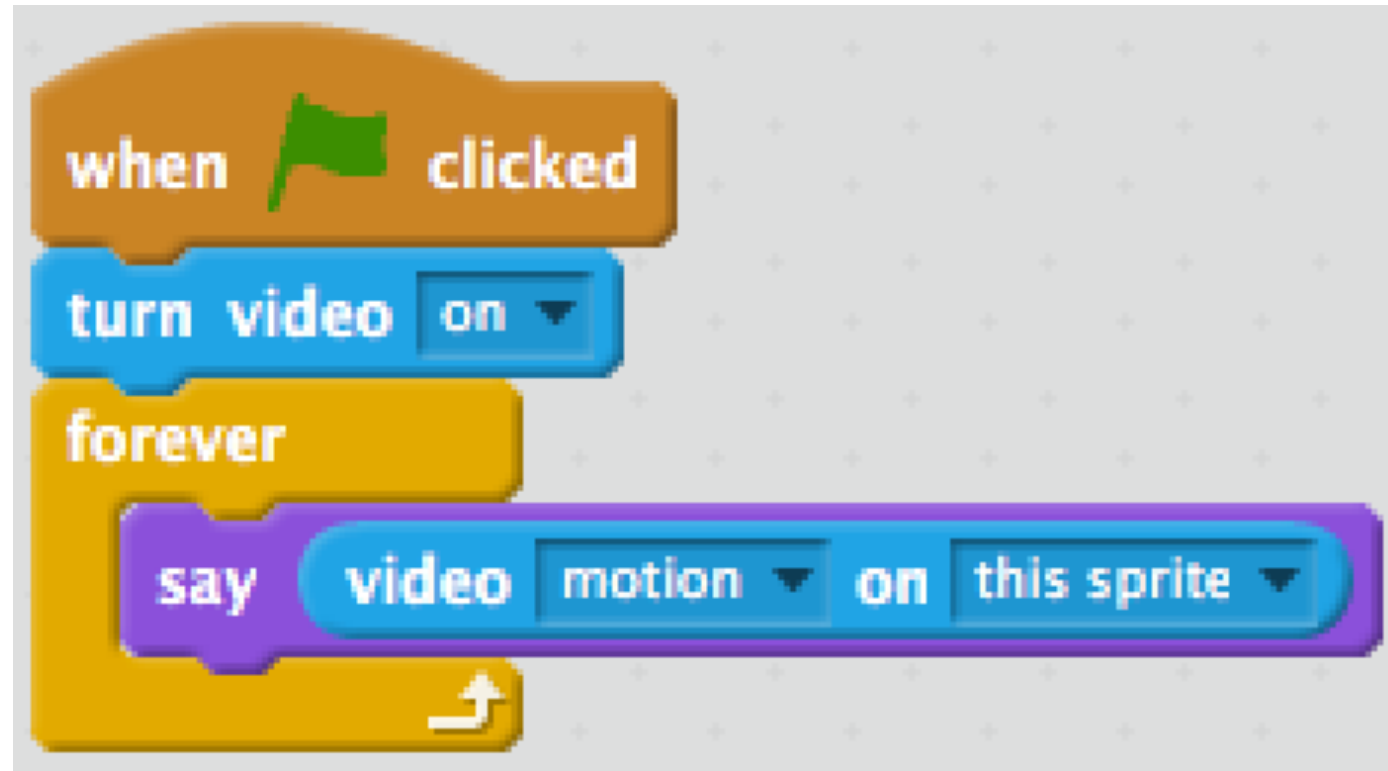
Jon Sparring
Department of Computer Science
2020/09/04

UNIVERSITY OF COPENHAGEN



Repetition fra sidste uge

- 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

F# som lommeregner

357

864

357+864

```
let a = 357
```

```
let b = 864
```

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

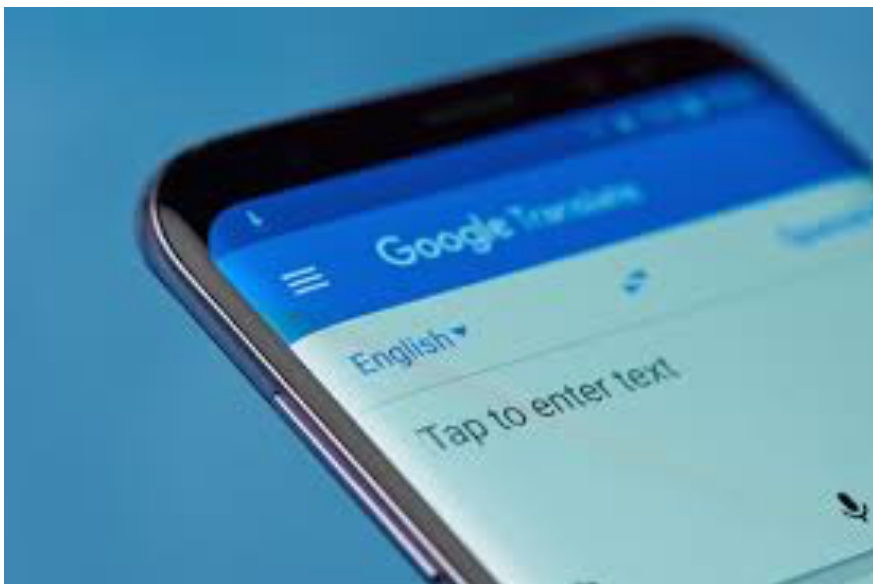
```
do printfn "%A" c
```

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)

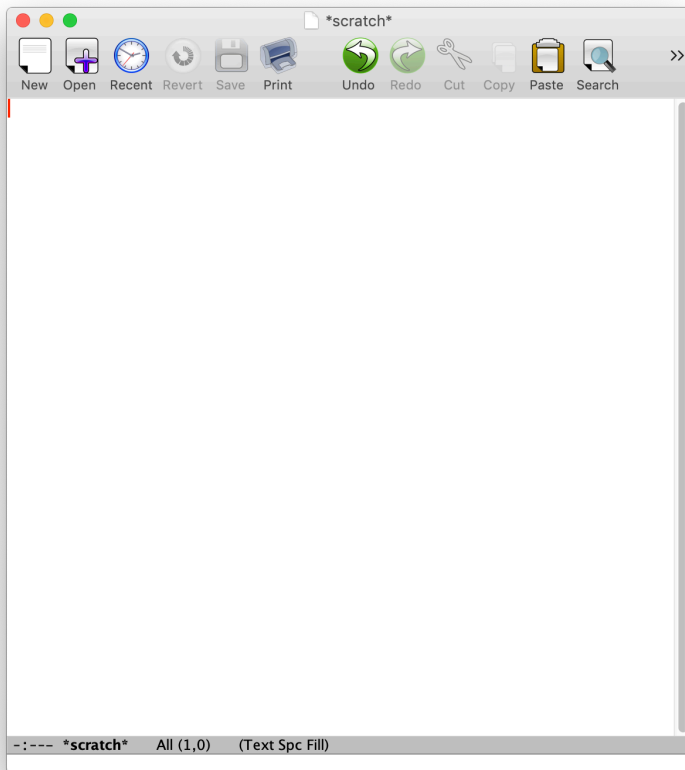
Interaktion



Kompilering



F#: Teksteditor, kompilering, kørsel



```
2.1ScratchNFsharp — -bash — 80x24
Jons-mac:2.1ScratchNFsharp sporring$ fsharp 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

Resumé

I denne video hørte du om:

- Repetition af programmeringselementer fra Scratch
- En struktureret tilgang til problemløsning
- F# som en lommeregner
- Forskellige måder at køre F# scripts
- Forskellen mellem et oversat (compiled) og et fortolket (interpreted) program