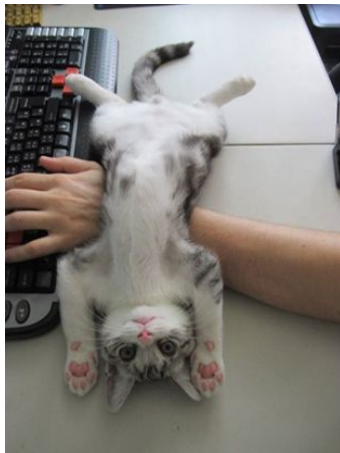


# 1. Programmering

- hvad er det?

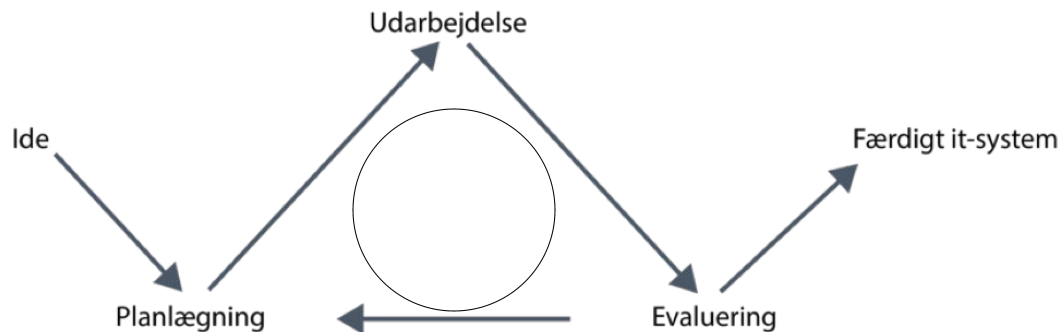
Informatik C  
August-september 2024



# Informatik C - kort introduktion

Informatik på C niveau:

- Informatik er et skaberfag hvor I skal arbejde ca. 50% praktisk og 50% teoretisk.
- I arbejder iterativt i informatik. Dvs. et projektarbejde, hvor arbejdsprocesserne gentager sig selv.



Hvilket IT bruger I? - snak  
sammen med sidemanden



# Informatik C – de faglige mål

- **Konstruktion af et it-system som løsning til en problemstilling**
  - Lave et mindre it-system helt eller delvist med lidt kode og/eller pseudokode
  - Et it-system består af tre lag (tre-lagsarkitektur): Præsentationslaget, datalaget, logiklaget.
- **It-systemers og menneskelig aktivitets gensidige påvirkning**
  - Hvordan påvirker it-systemer menneskers gøren og laden. Lever vi på en anden måde pga. it.
- **It-sikkerhed, netværk, arkitektur**
  - Hvordan et it-system er bygget op (arkitektur)
  - Hvordan internettet er bygget op (netværk)
  - It-sikkerhed (din og andres privatliv på nettet f.eks. ect.).
- **Repræsentation og manipulation af data**
  - Hvordan gemmer man data og finder det igen
  - Databaseopbygning
- **Programmering**
  - Simpel programmeringsforståelse, såsom for-løkker, if-else-sætninger osv.
- **Interaktionsdesign**
  - Design af præsentationslaget: hensigtsmæssig placering af knapper, tekst, valg af farve, nem navigering osv.
- **Innovation**
  - Hvad er et innovativt it-system?
  - Modeller til at kunne identificere innovation

# Faglige mål vi arbejder med i dette forløb:

- **Konstruktion af et it-system som løsning til en problemstilling**

- Lave et mindre it-system helt eller delvist med lidt kode og/eller pseudokode

- Et it-system består af tre lag (tre-lagsarkitekturen): Præsentationslaget, datalaget, logiklaget

- **It-systemers og menneskelig aktivitets gensidige påvirkning**

- Hvordan påvirker it-systemer menneskers gørem og laden. Lever vi på en anden måde pga. it.

- **It-sikkerhed, netværk, arkitektur**

- Hvordan et it-system er bygget op (arkitektur)

- Hvordan internettet er bygget op (netværk)

- It-sikkerhed (din og andres privatliv på nettet f.eks. ect.)

- **Repræsentation og manipulation af data**

- Hvordan gemmer man data og finder det igen

- Databaseopbygning

- **Programmering**

- Simple programmeringsforståelse, såsom for-løkker, if-else-sætninger osv.

- **Interaktionsdesign**

- Design af præsentationslaget: hensigtsmæssig placering af knapper, tekst, valg af farve, nem navigering osv.

- **Innovation**

- Hvad er et innovativt it-system?

- Modeller til at kunne identificere innovation

Video: "You should learn to program"



# Introduktion til logbog

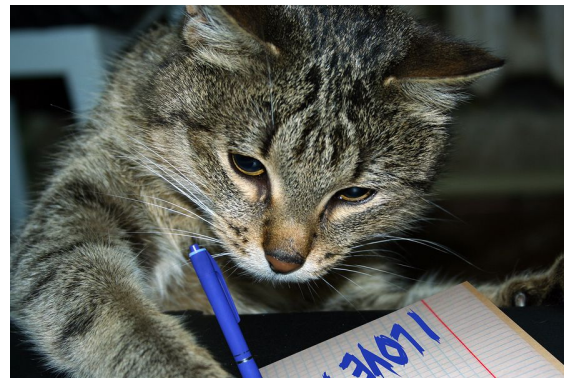
**Program:** Word, Google Docs eller lignende.

**Krav:** Skal kunne omdannes til pdf.

**Store filer?** Lægges i jeres skole-drev og der linkes til det i logbogen.

**OBS!** I er selv ansvarlige for backup af jeres logbog

**Op til karaktergivning:** I afleverer jeres logbog i opgaver i Lectio, når vi nærmer os karaktergivning. Jeg skal nok også sige til. Logbogen skal ikke være færdig. Jeg vil gerne se jeres produkter og arbejdsgang.

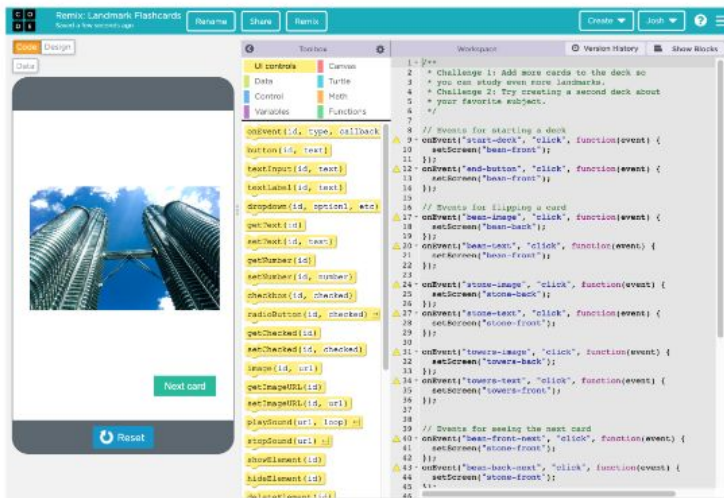


# Introduktion til logbog

## Informatik 2cd - C-niveau - Michala Dahl Mensing

14.08.2024:

I dag har jeg lært ~~SÅÅÅÅ~~ meget af min lærer Michala! Her kan du se, hvad jeg har bygget i App Lab:



The screenshot displays the App Lab workspace for a project named "Remix: Landmark Flashcards". The interface is divided into three main sections:

- Design:** Shows a mobile app preview with a background image of the Petronas Towers. A "Next card" button is visible at the bottom right, and a "Reset" button is at the bottom center.
- Toolbox:** A list of UI components such as "UI controls", "Data", "Control", "Variables", "Canvas", "Turtle", "Math", and "Functions".
- Workspace:** A code editor containing JavaScript code for handling events and screen transitions. The code includes comments and functions for starting a deck, flipping a card, and navigating between screens.

```
1 // **
2 * Challenge 1: Add more cards to the deck so
3 * you can study even more landmarks,
4 * Challenge 2: Try creating a second deck about
5 * your favorite subject.
6 //
7
8 // Events for starting a deck
9 onEvent('start-deck', 'click', function(event) {
10   setScreen('base-front');
11 })
12 onEvent('end-button', 'click', function(event) {
13   setScreen('base-front');
14 })
15
16 // Events for flipping a card
17 onEvent('base-image', 'click', function(event) {
18   setScreen('base-back');
19 })
20 onEvent('base-text', 'click', function(event) {
21   setScreen('base-front');
22 })
23
24 onEvent('tower-image', 'click', function(event) {
25   setScreen('tower-back');
26 })
27 onEvent('tower-text', 'click', function(event) {
28   setScreen('tower-front');
29 })
30
31 onEvent('tower-image', 'click', function(event) {
32   setScreen('tower-back');
33 })
34 onEvent('tower-text', 'click', function(event) {
35   setScreen('tower-front');
36 })
37
38
39 // Events for seeing the next card
40 onEvent('base-front-next', 'click', function(event) {
41   setScreen('atone-front');
42 })
43 onEvent('base-back-next', 'click', function(event) {
44   setScreen('atone-front');
45 })
46
```



# Robotøvelsen!

Øvelsen går ud på, at I skal lave en **model** af en **virkelig handling** som i dette tilfælde er at børste tænder.

I skal **skrive ned**, hvad der sker **trin for trin**, når man **børster tænder**. De første trin skal være:

1. **Stå foran badeværelsesskabet.**
2. **Tag venstre arm frem**
3. **Brug venstre arm og venstre hånd til at åbne badeværelsesskabet**
4. **Osv.**
5. **Osv.**

Bagefter **afprøver** vi, om **modellen** kan få en **robot** til at børste tænder.  
Brug **5-7** minutter på at skrive jeres trin ned.

# Refleksionskrivning

## Skriv i logbog:

- Hvad tænker du om forskellen mellem den menneskelige måde at tænke på (kognition) og computerens måde at tænke på (computational thinking). Fik robotøvelsen sat nogle tanker i gang om dette hos dig?
- Tror du, at der vil være forskel på den menneskelige kognition og computerens kognition i fremtiden?
- Tror du menneskeheden kan opnå mere, hvis vi alle får en forståelse for computerens tankegang og evt. lærer programmering?

Skriv i 3 minutter.



My army is ready, we attack at  
nightfall



# Forløbets programmeringsopgave:

Din lokale restaurant/café/juicebar/takeaway-sted/kantine/ect. har brug for hjælp:

Tiden er knap, kunderne er mange og personalet er i undertal. I skal derfor hjælpe firmaet med et stk. IT-system.

- Man skal kunne bestille varer gennem IT-systemet!

Opgaven er individuelt men I må gerne hjælpe hinanden. I må også gerne vælge den samme type vare.

# Forløbets programmeringsopgave:

I skal nu:

- Brainstorme alene eller sammen med sidemanden om hvilken restaurant/butik/ect. I vil hjælpe.
- Eksempler: Boblejuicebar, kantinen, pizzarestauranten nede på hjørnet, sushirestauranten på Algade.... osv. osv.
- Skriv idéerne ned i logbogen.
- Skriv det endelig valg ned i logbogen.



# De vigtigste begreber inden for programmering i Informatik C:

- Variabler
- Løkker
- Forgreninger
- Funktioner
- Sekvenser



# Variabler



# Variables

The screenshot shows a web development IDE interface. At the top, there's a teal header with "Untitled Project" and "Saved a few seconds ago". Below this are buttons for "Rename", "Share", "Remix", and "Create". The main workspace is divided into three sections: "Code", "Design", and "Data". The "Code" section is active, showing a list of code blocks in the "Toolbox" and a "Workspace" area. The "Toolbox" lists categories: UI controls, Data, Control, Variables, Canvas, Turtle, Math, and Functions. The "Workspace" area contains the following code blocks:

```
var x = 0;  
var x;  
x = 0;  
var x = prompt("Enter a va  
var x = promptNum("Enter a  
console.log(message)  
console.clear()  
var str = "Hello World";  
str.substring(start, end)  
str.indexOf(searchValue)
```

At the bottom, there's a "Run" button and a "Debug Console" section with "Show Debug Commands" and "Debug Console" options.



# Variabler - deklarerer/opret

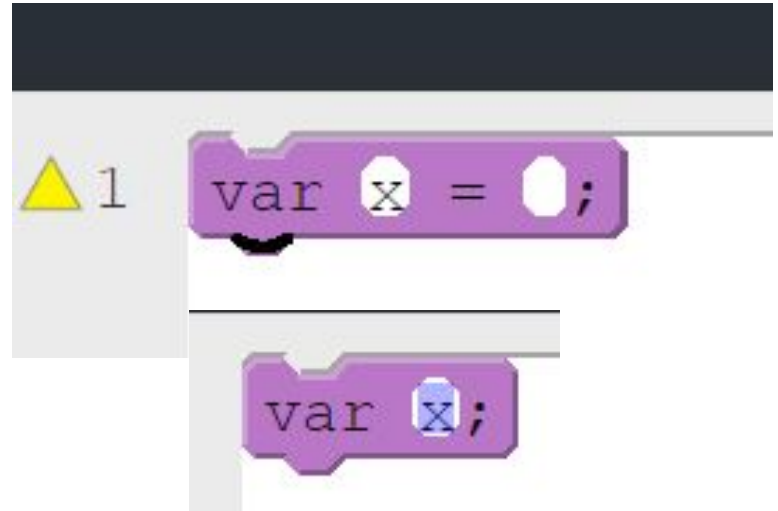
I App Lab:

Deklarerer først med: VAR

**Kun VAR første gang!** (pakker kassen ud)

X = erstat med logisk navn (skriver navn på kassen)

Tildel værdi eller vent (Put noget i kassen eller vent - =)



## Variabler - efter oprettelse

1. linje: opretter variabel og kalder den for: navn
2. linje: putter en værdi ind i min variabel: en streng (i gåseøjne"). (Streng er ord, sætninger)
3. Ændrer værdien.

KUN ÉN TING AD  
GANGEN

```
1 var navn;  
2 navn = "Michala Dahl Mensing";  
3 navn = "Julemanden";
```

# Variabler

En variabel er et *navngivet sted*, hvor du kan gemme noget.

Man kan sammenligne en variabel med *en kasse med en etiket på*. I kassen kan du lægge forskellige ting, fx tal, bogstaver m.m. Du kan ændre tallene eller bogstaverne, fjerne dem eller udskifte dem med nye.

En variabel skal have et navn, som du bestemmer - uden mellemrum.

<https://informatikforalle.ibog.forlagetcolumbus.dk/?id=285#c385>



Jetpack Cat

# Variabler

Hvad kan man putte i en variabel:

- Ord = **strings**                      **OBS: altid i gåseøjne “”**
- Sætninger = **strings**              **OBS: altid i gåseøjne “”**
- Lister = **arrays**
- Tal = **tal**
- Kode = **objekter**

Hvor meget kan der være i en variabel af gangen?

EN TING!!!



# App Lab

Untitled Project  
Saved a few seconds ago

Rename

Share

Remix

Create ▼

Terminatormic ▼



Code Design Data

screen1 ▼

▶ Run

Toolbox



UI controls Canvas  
Data Turtle  
Control Math  
Variables Functions

Workspace

🕒 Version History

</> Show Text

```
onEvent(id, type, callback)  
button(id, text)  
textInput(id, text)  
textLabel(id, text)  
dropdown(id, option1, etc)  
getText(id)  
setText(id, text)  
getNumber(id)  
setNumber(id, number)  
checkbox(id, checked)  
radioButton(id, checked)
```

English ▼



Show Debug Commands

Debug Console

🗑️ Clear

Watchers

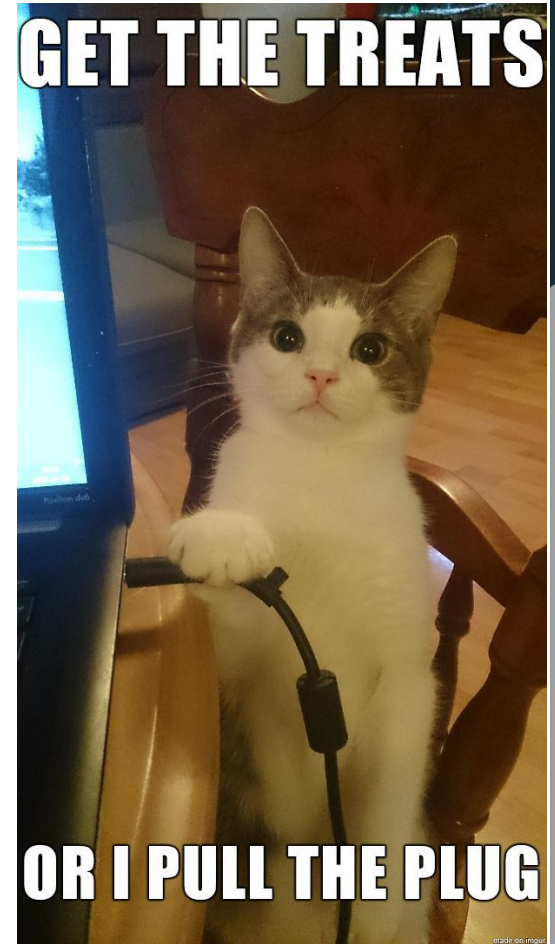


# App Lab



# App Lab - tutorial

<https://studio.code.org/s/applab-intro/lessons/1/levels/1>



# App Lab - design af bestillingsapp





# Skriv i logbog

- Hvad har jeg lavet/lært i dag?
- Skriv i jeres logbog i 5 minutter - indsæt gerne skærmpoint, hvis I har.



# Løkker



# Løkker

- Løkker er **gentagelser** af kode
- Smart hvis vi skal bruge den samme kode mange gange.

```
1 for ( var i = 0 ; i < 4 ; i++ ) {  
2  
3 }
```

opretter variabel med navn "i" - skal bruges til at tælle med

definerer hvor mange gange vi skal gentage - her 4 gange. Der står: så længe variabelen "i" er mindre end 4, så skal du blive ved med at udføre koden nedenunder.

Variablen "i" skal plusse sig selv med 1.

(Dvs.: nu står der 0 i variabelen, men vi plusser med 1 og så står der 1 i variabelen "i".)

# Forgreninger

The image features a woman with dark hair wearing a white and black striped sweater, positioned on the left side of a purple background. To her right are two educational graphics. The top graphic shows a row of ten human icons: four orange, two pink, and three teal. A pink bracket is drawn under the first six icons, with the text "PG 13" centered below it. The bottom graphic is a screenshot of a Scratch script editor, showing a script with several "say" blocks. The visible text in the script includes:

```
when green flag clicked  
say [ ] for 2 sec  
say [ ] for 2 sec  
say [ ] for 2 sec  
say [ ] for 2 sec  
say [ ] for 2 sec
```

# Forgreninger

- Også kaldet if/else.
- Bruges til at **vælge imellem nogle valg** i programmeringen - enten sker der dét eller også sker der dét her/enten går vi den ene vej ellers går vi den anden vej.
- Man kan tilføje uendeligt antal else if
- Man kan også bare nøjes med en if, uden else eller else if.

```
1 var brugernavn = "Michala";  
2 if (getText(▼ "user") == "Michala") {  
3     write("Rigtigt brugernavn!");  
4 }  
5
```

```
1 if ( ) {  
2  
3 } else {  
4  
5 }
```

```
1 var vejret = "solskin";  
2 if (vejret == "solskin") {  
3     write("Tag solbriller på");  
4 } else {  
5     write("Lad solbrillerne blive hjemme.");  
6 }
```

# Funktioner



# Funktioner

- En måde at **gentage** en sekvens/kode på.
- Man skriver et lille “program” inde i koden.
- Man **opretter/skriver** sit lille program og giver det et navn.
- Når man skal bruge programmet, **kalder** man på det.

```
7 function tjekVejret () {  
8  
9 }  
10 myFunction ( );
```

Skriver programmet og giver det et navn. (Her mangler jeg at skrive programmet.)

Her kalder jeg programmet, så det bliver udført.

# Funktioner

```
2 function tjekVejret () {  
3     if (vejret == "solskin") {  
4         write("Tag solbriller på");  
5     } else {  
6         write("Lad solbrillerne blive hjemme.");  
7     }  
8 }  
9 tjekVejret();
```



# Sekvenser (sequences)

- Robotlegen - tandbørsteøvelsen
- Kan sammenlignes med en opskrift man følger minutiøst
- F.eks.:
- `var navn = "Michala";`
- `var besked = "Hej ";`
- `var personligBesked = besked + navn;`

# Tutorials i App Lab

<https://studio.code.org/s/csp4-2022/lessons/1/levels/1>

<https://studio.code.org/s/csp4-2022/lessons/3/levels/1>

<https://www.youtube.com/watch?v=GDV4f2brxfE>

# "Informatik for alle": "Guide til App Lab"

<https://studio.code.org/projects/applab/U4jhmmFLLwdDf7n-sChKkpH-CmfjuDYTrV6FqJoSw6I>

## First Semester

me in october:

me in january:

```
34 write("Du vandt en sekser!");
35 } else {
36 write("Desværre, der er kun gevinster på seksere. Prøv igen.");
37 }
38 //Funktion:
39 function login() {
40 var navn = prompt("Brugernavn: ");
41 var kode = prompt("Kodeord: ");
42 if (navn=="admin" && kode=="1234") {
43 write("Du har skrevet det rigtige!");
44 setScreen(▼"login");
45 } else {
46 write("Dit brugernavn og kodeord er ikke korrekt.");
47 }
48 }
49 login();
50 function udskrivNavn (navn) {
51 setText(▼"label1", "Du er nu logget ind " + navn);
52 }
53 udskrivNavn (mitnavn);
54
```



# Michalalas bestillingsapp - et løsningsforslag



<https://www.youtube.com/watch?v=GDV4f2brxfE&t=680s>

App Lab - design af bestillingsapp



## Rutediagram og pseudokode

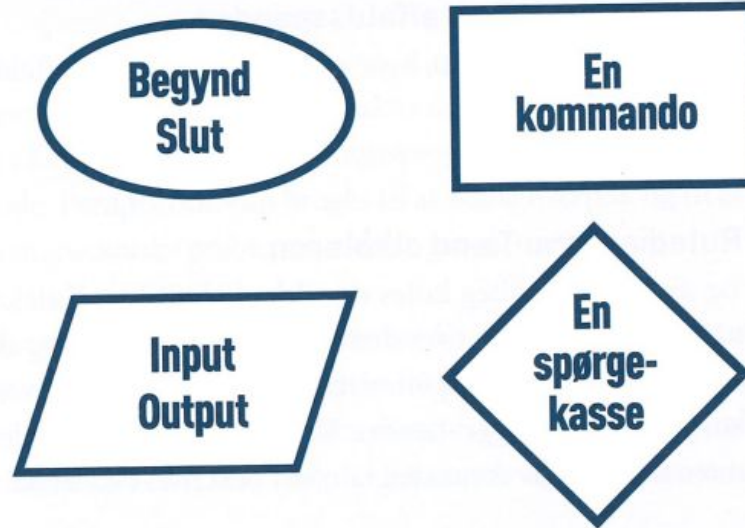
- Læs kapitel 3 fra bogen “Informatik C” t.o.m. s. 37 og besvar arbejdsspørgsmål. (Ligger på modulet).

# Rutediagram



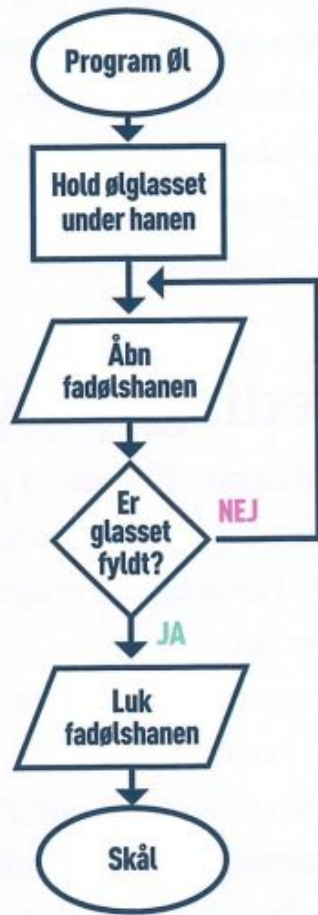


# Rutediagram



Figur 3.4: De vigtigste figurer i et rutediagram.





Figur 3.5: Rutediagram for øl-program.

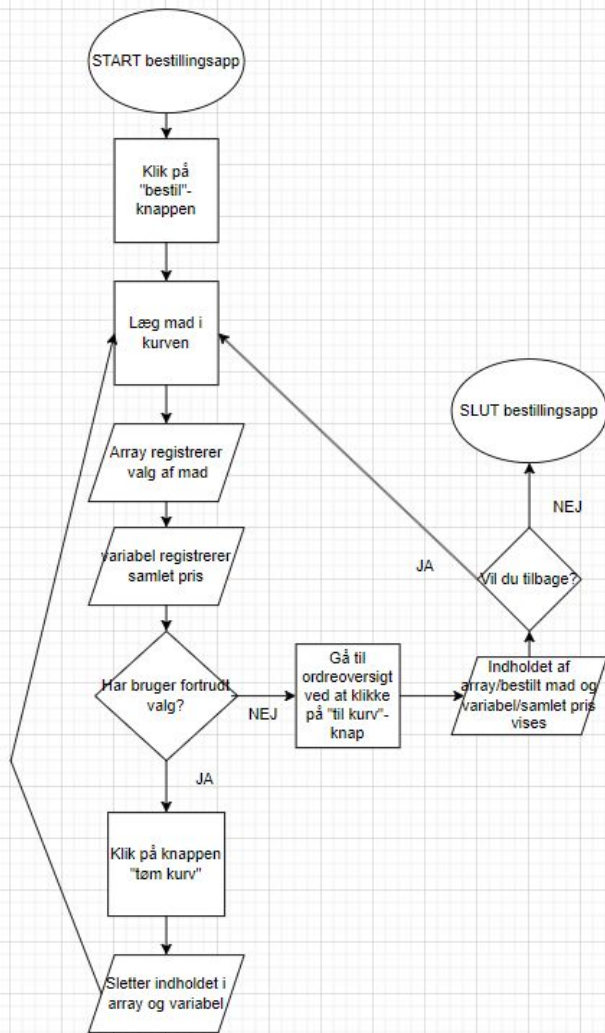
## Rutediagram

Kommando: en funktion eller programmeringshandling

Input/output: data der går ind og ud efter en kommando.

Input repræsenterer de oplysninger eller værdier, der er nødvendige for at udføre en bestemt kommando eller proces.

Eksempel: For en kommando som "Tilføj vare til kurv" kan input være oplysningerne om den valgte vare, og output kan være opdateringen af indkøbskurvens status.



Nu skal I lave eget rutediagram  
over egen app.



Speeddating!



# Prototyper

**Hvad er det:** tidlig skitse af et it-produkt.

**Formål:** Hurtigt danne sig overblik over it-produktet og hurtig måde at arbejde med idéudvikling af app. Og vise brugerne og/eller medudviklere, hvad man forestiller sig at ville bygge.

**Forskellige slags:**

**“Kvaliteten”/omfanget:**

- Low fidelity
- High fidelity

**Til hvilket formål:**

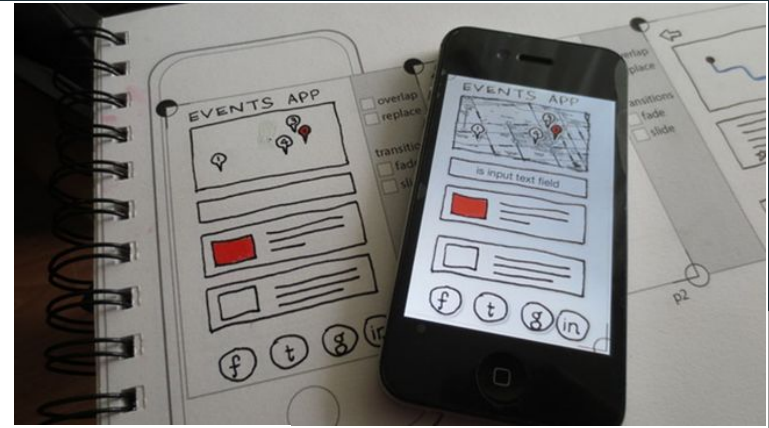
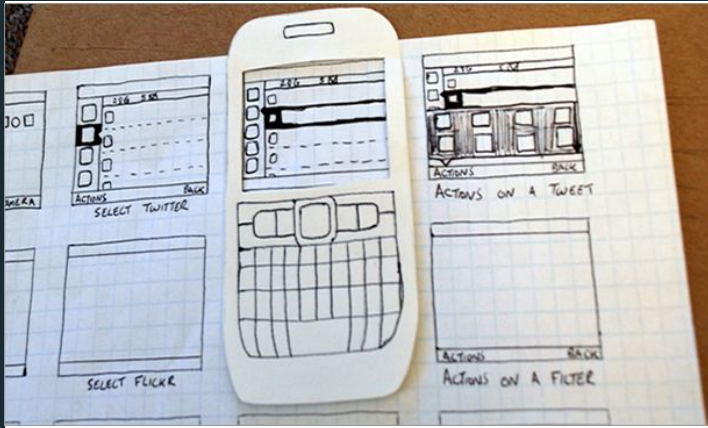
- Evolutionær prototyping
- Throw away prototyping
- Rapid prototyping



# Prototyper – Lofi vs. hifi

- **Lofi / Low fidelity prototyper:** papir, træ metal, simple, billige materialer, hurtig at lave – fremstilles tidligt i forløbet.
- **Hifi / High fidelity prototyper:** ligner det endelige produkt, mere funktionalitet, tidskrævende og kræver flere ressourcer, men kan bruges til at teste mange flere aspekter af produktet, f.eks. det følelsesmæssige aspekt – fremstilles som regel sent i forløbet.





iPhone Prototype - 1983

It lacked the portability of a mobile phone, but showed that Apple was looking to take the initiative in touchscreen technology long before the iPhone was even a glimmer in Steve Jobs' imagination.

Evolutionær,

throw away,

rapid

- **Evolutionær:**

I evolutionær prototyping bliver man ved med at udvikle prototyper, der forfølger samme spor som, fx hvordan skal brugergrænsefladen virke.

- **Throw away:**

Man laver en mockup/skitse/rapid prototype. Så prøver man den af med målgruppen, får noget feedback og bliver klogere. Derefter skaber man en ny type prototype og får mere viden. Hver gang man gennemfører en prototypesession, smider man den gamle version af prototypen væk (men man beholder naturligvis den viden, som man har fået).

- **Rapid:**

Man fremstiller hurtigt nogle mock-ups eller prototyper. Det er illustrationer af, hvordan produkter eller løsninger kan komme til at se ud. Disse skabes og vises til brugerne, så man hurtigt kan få noget feedback fra brugerne. Det kan bruges til at teste et koncept. Hvis konceptet ikke virker på papir, vil det ikke virke digitalt.



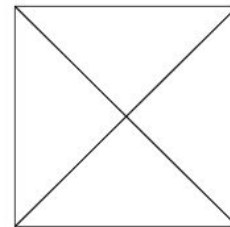
# Lofi, rapid prototyping

I skal nu skitsere jeres app-idé på whiteboards.

I skitserer alle skærme: fx

1. skærm: forside med billede
2. skærm: oversigt over varer og mulighed for at klikke dem i kurv
3. skærm: ordreoversigt/kurv
4. skærm: frivilligt om hvor mange ekstra skærme I ønsker.

Kattens kantineservice



Klar til at bestille

# Hurtig intro til Tænke-højt-test

- **Tænke-højt-metoden er et stærkt værktøj til at afdække problemer på en hjemmeside eller på skærbillederne i en app. Metoden giver et indblik i, hvorledes brugerne navigerer på en side, hvordan ikoner og menupunkter opfattes.**
1. Man lader testbrugeren prøve produktet efter anvisning og siger ikke noget! Hvis testbrugeren går i stå og ikke kan finde vej, så viser det en fejl i systemet.
  2. Man laver en række opgaver. I jeres tilfælde fx:
    - A) klik på startknappen på forsiden.
    - B) Vælg mindst ét produkt, du skal putte i kurven
    - C) Gå til ordreoversigten.
    - D) Find selv på flere...





Fortsæt med min bestillingsapp i  
App Lab

<https://www.youtube.com/watch?v=GDV4f2brxFE>



# Walk and talk - 15 minutter

- Hvad er dit yndlingsprogram/-app?
- Kan du huske og beskrive dit første møde med en computer?
- Hvor meget bruger du din smartphone og til hvad?
- Tror du robotterne kommer til at overtage verdensherredømmet?



# Skriv i logbog

- Hvad har jeg lavet/lært i dag?
- Skriv i jeres logbog i 5 minutter - indsæt gerne skærmpoint, hvis I har.

