

Eksempel i regneark: Din mads klimabelastning

Alle ved, at det belaster klimaet mindre at leve vegetarisk. Og endnu mindre, hvis man lever vegansk. Men hvor meget drejer det sig egentlig om? Det er ret nemt at udarbejde et regneark i fx Excel, hvor man bruger variable og funktioner til at holde styr på, i hvor høj grad man belaster miljøet med forskellige fødevarer.

På tabellerne på næste side kan man se, at jeg har indsat en funktion til hver række, hvor hver variabel er en celle. Eksempelvis er cellen i regnearket B7 en variabel, som jeg tildeler værdien 150 (g) – fordi det er rimeligt normalt at spise ca. 150 gram rødt kød til aftensmaden. C7 er en anden variabel, der henter en CO₂-belastningsværdi fra en tabel over fødevarer. I D7 er formlen for den samlede mængde CO₂ for den bøf, du har spist, nemlig de to tal ganget sammen, og ganget med 0,001 (så gram bliver omregnet til kg). Det giver miljøbelastningen af den fødevarer, man har indtaget i kg CO₂. I totalen i felt D12 har jeg taget summen ved at skrive =SUM(D5:D11).

- =SUM er en funktion, der er indbygget i Excel og de fleste andre regneark, og som tager summen på de variable fra D5 til D11.
- Kolon'et er en funktion i funktionen, der fortæller, at Excel skal udføre handlingen på alle værdierne fra D5 til D11.
- Lighedstegnet fortæller Excel, at det skal regne det ud, der står i cellen.

Sammenligner man de to totaler, kan man se, at man mere end halverer sin miljøbelastning ved ikke at spise oksekød og kødpålæg.

Madoversigt til beregning af miljøbelastning				
Køddag				
Vare	Mængde i g	CO ₂ -per kg	Miljøbelastning	
Mælk	100	1,2	0,12	CO2/kg: 1
Havregryn	60	0,8	0,048	km i bil: 5
Bøf	150	19,4	2,91	
Grøntsager (gnsn)	600	0,3	0,18	
Brød	200	0,8	0,16	
Kødpålæg	50	5	0,25	
Kaffe	600	0,2	0,12	
Total			3,788	
På et år			1382,62	
Vegetardag				
Vare	Mængde i g	CO ₂ -per kg	Miljøbelastning	
Mælk	100	1,2	0,12	
Havregryn	60	0,8	0,048	
Omelet	150	2	0,3	
Grøntsager (gnsn)	600	0,3	0,18	
Brød	200	0,8	0,16	
Ost	50	11,3	0,565	
Kaffe	600	0,2	0,12	
Total			1,493	
På et år			544,945	

Eksempel i Excel: Undgå at blive fattig som gammel, selv om du går på barsel

Du har måske børn og været mere på barsel end den anden forælder. Måske er du også gået ned i tid i en periode? Ud over, at du har fået mindre ind på kontoen i denne periode, så kommer du også til at mangle en del på din pensionsopsparing i forhold til den anden forælder. Men hvor meget?

Det kan man nemt lave et lille Excel-program, der kan regne ud. Og så slipper du også for dødkedelige møder med din pensionsrådgiver på et eller andet tåbeligt tidspunkt og kan i stedet lave en særskilt opsparing eller betale det beløb ind på din pension, der mangler fra fælleskontoen (eller den anden forælder, hvis I ikke har fælleskonto).

- Skriv i et felt, fx B3, hvad din månedsløn er (inklusive pension). Hvis børneroderiet strækker sig over otte år, bør det være lønnen sådan lidt over halvsvejs som tommelfingerregel. Brug evt. den løn, du starter med, og gang den så med inflationen, altså $30000 * (1,02)^5$. Du kan også sammenligne med dit barns anden forælders lønstigning over perioden.
- I et andet felt, B4, skriver du, hvor mange måneder, du går på barsel i alt (her 24 mdr. fordelt på to børn), i et tredje, B5, skriver du, hvor mange måneder, du går på deltid på grund af børnene eller huslige opgaver.
- Hvis du også går på deltid, fx 32 timer, kan du i et fjerde felt (B6) skrive $= (1 - 32/37)$.
- I sidste felt skriver du, hvor meget af din løn, der går til pension. Det er normalt mellem 12 % og 18 %.

I felterne foran A3 skriver du, hvad tallene betyder. Så skal du beregne pensionsbeløbene.

- I fx E4 ganger du beløbet med det antal måneder, du regner med at være på barsel. Det gør du ved at skrive $=B3*B4$. Så får du det samlede beløb over den tid, du går på barsel.
- I det fjerde felt, E5, ganger du det antal måneder, du går på deltid, med din løn og den andel "fri", du beregnede i B6, altså $=B3*B5*B6$.
- I det sidste felt, E6, tager du de to tal og lægger sammen ($=E4+E5$). Og så får du det samlede beløb, som du går glip af i din pension.

	A	B	C	D	E	F	G
1							
2	Grunddata						
3	Løn pt	30.000			Mistet pension		
4	Mdr på barsel	24		Under barsel	108.000,00		
5	Mdr på deltid	48		Under deltid	29.189,19		
6	Andel fri	14%		Total	137.189,19		
7	Pensionsbidrag	15%					
8							

Hjemmesider og HTML-kode

HTML er et sprog, som ikke er et egentlig programmeringssprog, men det bruges til at programmere hjemmesider og markere, hvordan tingene skal stå. Det kaldes også et "markup-language" eller på dansk "markeringssprog".

HTML er nemt at bruge, og man får resultatet at se lynhurtigt. Derfor er det også her, vi starter med at forstå, hvad kode er. Vi skal i gang med den første øvelse.

Du kan selvfølgelig også bare give den gas med at bygge din egen hjemmeside. Men hvis du ikke lige har et domænenavn (som fx *filo.dk* eller *kvindekenddinkode.dk*, som koster penge), kan du også programmere din helt egen side. Her kan du lægge alle dine yndlingslink, som du fx bruger som automatisk opstartsvindue, når du åbner din browser.

Jeg giver dig her nogle hints til, hvordan du kan programmere dit eget opstartsvindue med såkaldt dybe links til dine yndlingswebshops, streamingtjenester og sociale medier. Får du blod på tanden og vil lære mere HTML, kan jeg anbefale dig at låne en bog om det eller gennemføre en HTML-tutorial.

Åbn din editor og gem dit dokument som fx *browserstart.html*. Åben samme dokument ved at dobbeltklikke på det, så åbner det i et browservindue som *Internet Explorer* eller *Chrome*. Det kan du genopfriske hver gang, du gemmer nye ændringer i din editor. Se også, hvordan det ser ud på side 71.

Det er efternavnet (.HTML), der er vigtigt og får det til at åbne i en browser, hvis du bare dobbeltklikker på filen. Ligeledes vigtigt starter og slutter de fleste elementer i HTML med `<>` og `</>`. En hjemmeside starter og slutter fx altid med `<HTML>` og nederst på siden er den så afsluttet med `</HTML>`. Den slags markører definerer alt, hvad der sker på en hjemmeside. De kaldes også "tags", og du kan se dem i screendumpet på side 71. De mest almindelige får du her:

HTML-Tag (fedet)	Betydning
<H1> En overskrift </H1>	En overskrift på øverste niveau. Næste niveau hedder <H2> og niveauet under <H3> , osv.

	Laver linjeskift. Det gør HTML nemlig ikke automatisk.
 Denne tekst bliver kursiveret 	Kursiverer et ord eller en sætning.
	Visning af et billede med url-henvisning til billedet.
 Navn på link 	Link til en hjemmeside.
<iframe width="640" height="360" src="URL" allow="autoplay; encrypted-media" allowfullscreen></iframe>	Tillader et lille billede af fx en YouTube-video at spille direkte på hjemmesiden. Se eksempler på <i>kvindekenddinkode.dk</i>

Simple links

Først skriver du den tekst, der skal stå i vinduet. Så finder du de links, du skal bruge. Hvis du bare vil bruge helt almindelige links, som normalt bliver blå, fede og understregede, kan du lave dem således:

```
<a href=https://www.roskilde-festival.dk/>Roskilde Festival</a>
```

Dybe links

Med dybe (embedded) links bliver forsiden eller videoen vist på din side i mindre rammer, som du kan se i den lille karatevideo i screendumpet på næste side.

Det er smart, hvis du vil programmere en side, der samler links til fx yndlingsserier, webshoppingsteder eller bare de mest brugte links, som du så kan bruges som din personlige browser-startside. Hvorfor: Fordi den så bliver mere visuelt lækker end bare en samling link.

Men det kan også være, at du foretrækker en enkelt side med links, der står overskueligt i fx en tabel. I selve HTML-koden står tabeller i "rå" tekst, som godt kan virke uoverskuelige – til gengæld kan man gøre dem pænere end i fx Word.

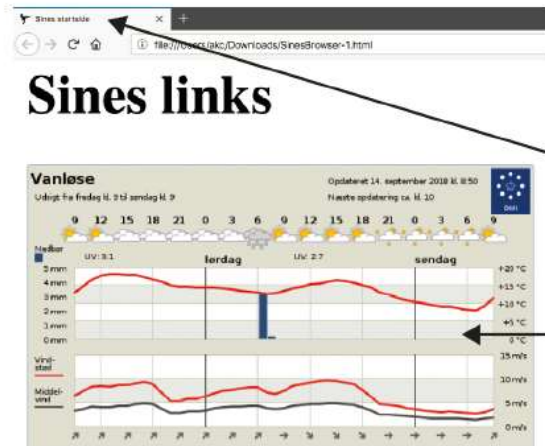
Hvis du skal lave dybe links fra fx You Tube, kan du højreklikke under videoen og klikke "Copy embed code". Så får du noget kode, der ser ud som fx:

```
<iframe width="640" height="360"  
src="https://www.youtube.com/embed/z5S579gboLs?list=PLjn3VZ_5uSwejeVmhZBu5s  
qlY1uvt5jyB" frameborder="0" allow="autoplay; encrypted-media"  
allowfullscreen>  
</iframe>
```

Og det kopierer du nu over i dit HTML-dokument, for det er noget rod at skrive selv.

Ud over at programmere links kan du også lave alt muligt andet gejl – både med hensyn til farve, størrelse, placering, faneblade, etc. Det er dine tags, der styrer, hvordan det kommer til at se ud.

Når du er færdig, kan du få din startside til at vise siden automatisk, når du starter din computer op. Det gør du som regel i browserens "indstillinger".



```
<HTML>
<HEAD>
<link rel="icon" href="https://npkaratekids.com/wp-content/uploads/2012/11/Karate_girl_2.jpg">
<TITLE>Sines startside</TITLE>
</HEAD>
```

```
<H1>Sines links</H1>
```

```
<p><span></span></p>
```

[Smykker paa udsalg](https://www.chanti.dk/p-smykkeudsalg)

```
<p><a href="https://www.chanti.dk/p-smykkeudsalg-oereringe" target="_blank"> Smykker paa udsalg </a></p>
```

[Videnskab.dk - Data og tal](https://videnskab.dk/topic/data-og-tal)

```
<p><a href="https://videnskab.dk/topic/data-og-tal"> Videnskab.dk - Data og tal</a></p>
```

Karatevideoer

```
<H3> Karatevideoer </H3>
```



```
<iframe width="200" height="100" src="https://www.youtube.com/embed/z5S579gboLs?list=PL-jn3VZ_5uSwejeVmhZBu5sqlY1uvt5jyB" frameborder="0" allow="autoplay; encrypted-media" allowfullscreen=""> </iframe>
```

```
</HTML>
```

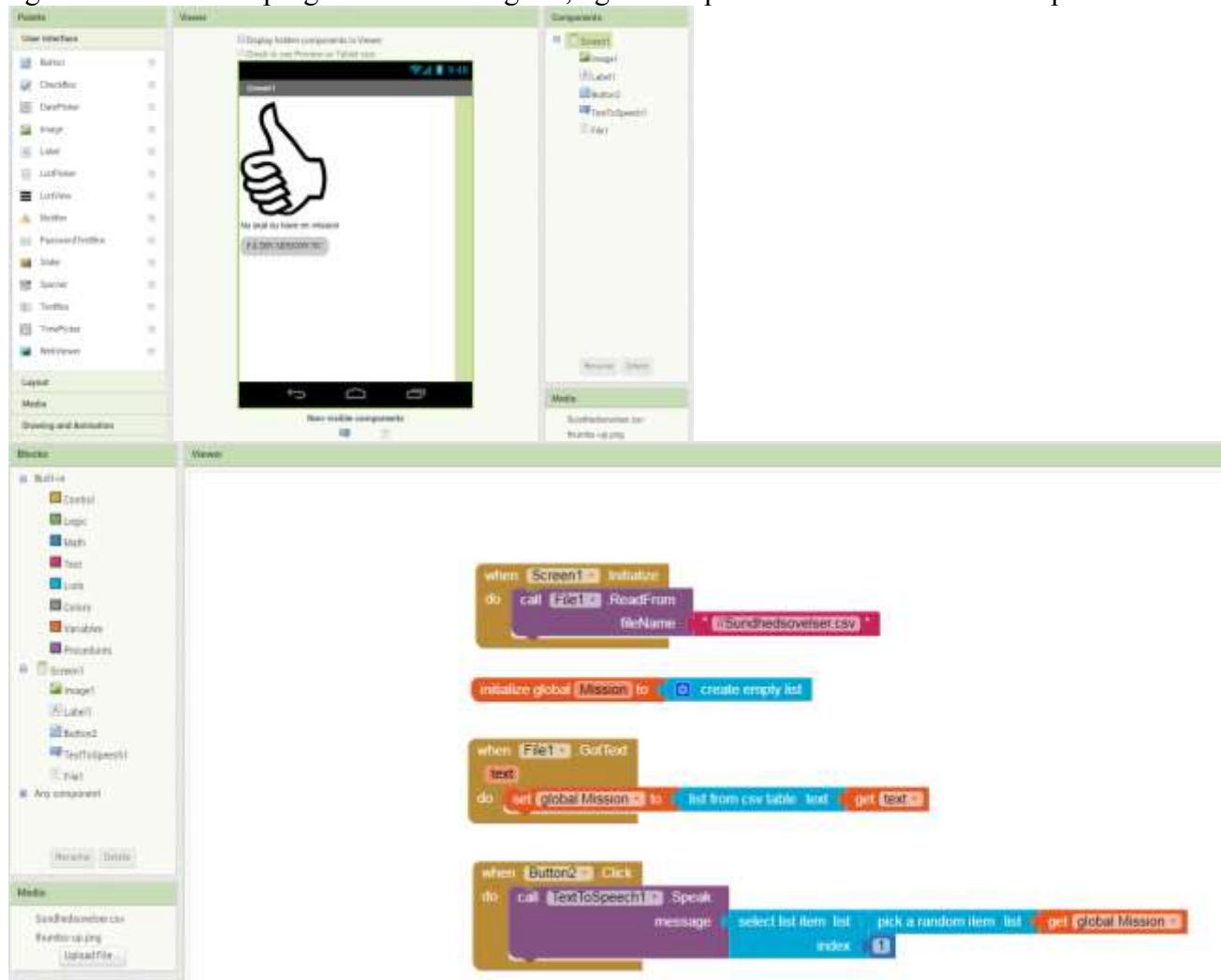
Øvelse: Din egen træningsapp i App Inventor

Der findes masser af træningsapps, men ofte passer de ikke lige helt til ens behov. Det er muligt selv at programmere en rimelig simpel træningsapp, hvor du fx har et dagsprogram eller ugeprogram, som du selv kan rette til i Excel eller Notepad. Og hvor du kan bruge nogle billeder, du synes er fede.

Til dette kan du bruge App Inventor, der er udviklet af MIT. Her bruger du såkaldte blokke til at kode med, og det er visuelt farvede klumper, man kan sætte sammen. Blokkene er designet så smart, at det er til at læse, hvad programmet gør. Og man laver ikke så nemt stavfejl som i tekstbaseret kode.

Du kommer i gang ved at gå ind på hjemmesiden ai2.appinventor.mit.edu, og du kan logge ind via dit google-id (eller oprette et ny id). Du har ikke brug for at downloade et program for at komme i gang – du starter bare et nyt projekt.

Mit oplæg til en træningsapp er at indkode en række øvelser, som kan udføres hurtigt, dér, hvor man lige er. Dem trækker programmet tilfældigt ud, og en computerstemme læser øvelsen op.



Design

Først laver du et design i *Designer*, som det ses i første screendump i øverste højre hjørne, hvor du sætter de knapper ind, der skal være. I denne app kan du i princippet nøjes med en knap til at få en mission på, et teksttiltale-modul, et labelmodul og en fil.

Jeg har også tilføjet en knap til at indlæse filen med, fordi det gør det nemmere at finde ud af, hvor det er gået galt, hvis der kommer fejl. Det er nemlig altid godt at have flere mellemtrin, end man har brug for i slutproduktet.

Hvis det virker uoverskueligt, så prøv at tegne appen, eller skriv ned linje for linje, hvad der skal ske. Så er det nemmere at implementere.

Hvad sker der i programmet?

I de første par blokke gør du data klar: Sørg for, at appen læser en tekstfil med øvelserne ind – fx en Excel- eller .txtfil, når man trykker på knappen. Det er den gule, lilla og pink blok, der tilsammen sørger for dette. Filen skal hedde .csv og kodning skal være *UTF-8*

- Man sætter sin *variable*, Mission, (orange blok) til en tom *liste* (ligesom at stille en tom skål frem)
- Man sørger for, at ens fil – som er af datatypen *tekst* (med *get text*) – bliver læst ind i den tomme liste (som i muffineksemplet hvor de *tørre* ingredienser blandes).

I den nederste blok sker det spændende.

Her skriver du, at når du trykker på knap 2 (den finder du under “components” i “vieweren”), ja, så bliver der udtrykt en tilfældig sætning fra din træningsfil, som bliver læst højt. De lyseblå blokke sørger for, at data er, som det skal være, og det mørkeblå 1-tal er indexet for, at det er fra første element. Det svarer til den første kolonne i et regneark, fx det Excellark du lige har læst ind. Det giver bedre mening nede i skattejagtseksemplet, hvor du også skal bruge 2. element, altså anden kolonne i Excelarket.

Betingelsen (gul) er, at du trykker på en knap, og *funktionen (lilla)* er, at du tager et tilfældigt element (en orange *variabel*) fra din liste, mens en anden *funktion (lilla)* læser teksten højt. Blokkene er kodet i forvejen, og du hiver dem ind i programmet med cursoren.

Der er ikke som sådan *gentagelser* eller *løkker* i denne simple app, men det kan der komme.

Når du først er i gang, kan det være, at du har andre ønsker end at hive en tilfældig mission ud. Det kan fx være, at du gerne vil træne i en bestemt rækkefølge eller opdele øvelserne i morgen- og aftenøvelser. Eller give dig selv point, som løbende tælles sammen, fx pr. dag eller uge. I den mere avancerede ende kan du bruge tidtagning og kilometertælling, som udnytter mobilens ur og GPS. Hvis du laver appen sammen med nogen, kan I skiftes til designe ugens øvelser, som skrives i et Excellark eller en anden editor og hives ind i appen.

Sådan bliver det en app

Det er vigtigt, at du tester programmet løbende. Specielt, når der ikke er en opskrift som her. Den nemmeste måde at teste på er ved (øverst midt for i første screen dump) at klikke *Connect* og derefter *AI-companion* eller ved at klikke i *Build* og derefter *App (qr code for .apk)*. Derefter følger du guiden. Hvis du bruger et offentligt netværk, er der ofte begrænsninger for, hvad du kan, og så er *Build* et bedre alternativ, selv om den er langsommere end *AI-companion*.

Når du er helt færdig, kan du gemme det som en app-fil, eller oploade til Google Play, hvis du er virkelig glad for den (og måske vil tjene penge på den). *.apk*-filen kan du downloade på *kvindekenddinkode.dk* og arbejde videre med den selv.

Øvelse: Skattejagtsapp

Her er en simpel opskrift på en skattejagt, hvor du bruger App Inventor. Det er en anelse mere komplekst end træningsappen. Til gengæld har du derefter en succes-opskrift på en fed skattejagt til ungerne, til vennerne på sommerhusturen eller til Roskilde Festival.

Indekser og tabeller

Når du skal programmere en skattejagt-app, kan du bruge, hvad du har læst om løkker, funktioner, variable og betingelser. Men du får lige et begreb mere: *indeks*, eller på engelsk, *Index*.

En almindelig liste med fx fem skattejagtsposter indeholder en række elementer med indekserne 1 til 5. Altså første skattejagt er den tekst, der er i en variabel med indeks=1, anden post har indeks=2 og så videre. Man kan sige, at indekset peger på den tekst, der er i variablen. Vær dog opmærksom på, at i mange programmeringssprog har første element i en liste indeks=0.

Mere avanceret bliver det, når der til hver post på skattejagten skal være et spor hen til næste post. Input til appen ville man lave i Excel som en tabel med to kolonner: I den første kolonne er opgaven og i anden kolonne er sporet. Så vil første post have indeks 1, 1, altså første række, første kolonne, mens sporet til næste post hedder 1, 2. Altså første række - anden kolonne. I Excel er der en indeksering, som du også arbejdede med tidligere. Her hedder rækkerne 1, 2, 3, 4.. osv., mens kolonnerne hedder A, B, C, D.. osv.

Filen med posterne skal gemmes som .csv-fil i Excel. Prøv at åbne den i Notepad eller Notepad++ og tjek, at den er komma-separeret. Hvis den ikke er det, retter du det bare inde i Notepad.

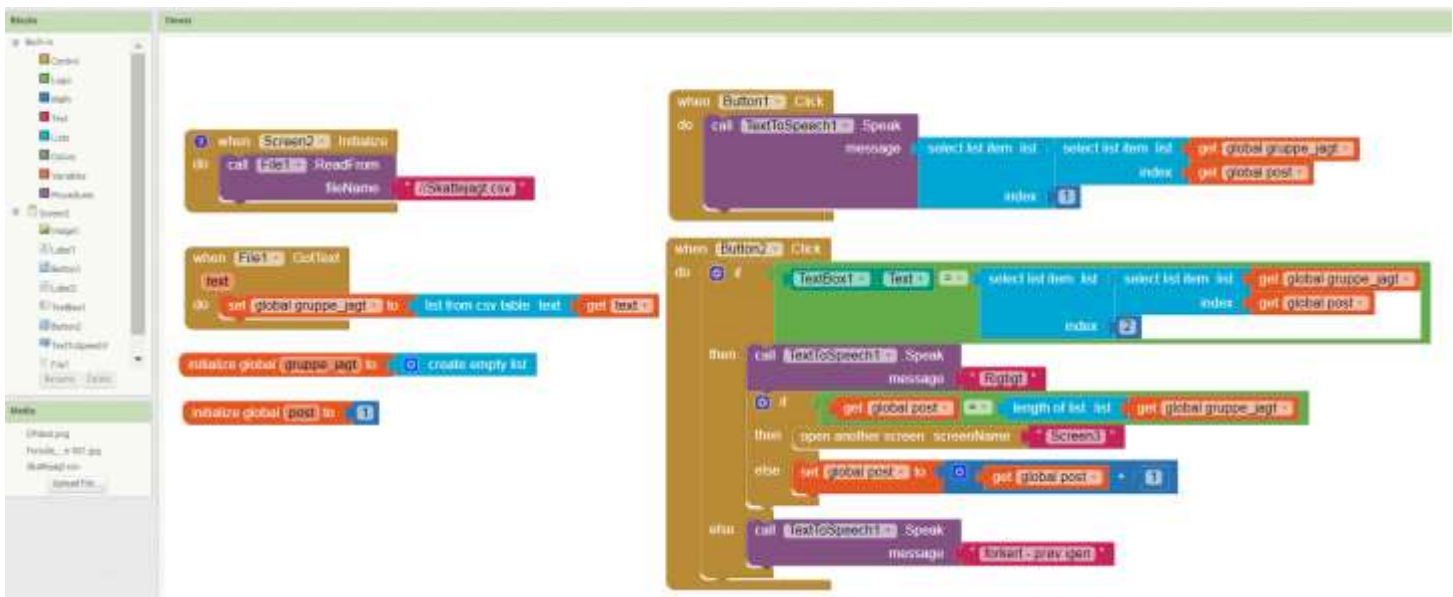
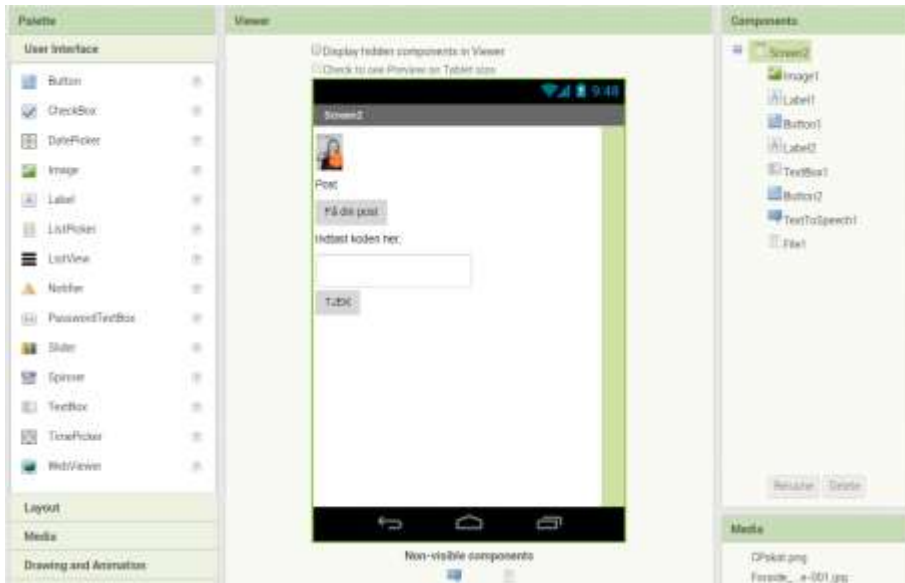
I screendumpet kan du se, at hver gang man trykker på knappen, får man en post læst højt, og hvis man svarer rigtigt, bliver der lagt 1 til post-indekset. Næste gang man trykker knappen, får man den næste post, som er den næste række i file.

Index	1 Spor til næste skattejagtspost	2 Koden på posten
1	Post 1: I busken i forhaven (indeks=1,1)	XYZ (indeks=1,2)
2	Post 2: I garagen (indeks=2,1)	FFF (indeks=2,2)
3	Post 3: På toilettet (indeks=3,1)	BUM (indeks=3,2)
Osv.

Appen holder styr på posterne

Ud over indekseringen, der i sig selv er lidt kompleks, skal man også bruge flere *screens* (nye skærme, der åbner sig på mobilen). Én som er velkomstskaerm, én til posterne og én til, når man er færdig og får sine point eller lignende. Neden for ser du de blokke, der koder for indgangsskærmen på telefonen.

Hele venstre side og den øverste blok til højre er omtrent ligesom sundhedsapp-koden, bortset fra at vi introducerer variabelen, *post*, der fungerer som indeks. Resten af koden til højre handler om, at programmet skal vurdere, om der er svaret rigtigt. I nederste højre hjørne kan du se, at man ved at svare rigtigt tæller post-tallet op, så man kan få et hint om, hvor næste post befinder sig. På den måde kan man lave mange poster uden at koden bliver længere.



Skattejagten kan både være børnevenlig som her, hvor man får hjælp, eller mere hardcore, hvor man skal svare rigtigt og får straffepoint for at gætte forkert. Jeg har ikke udfoldet hele programmet i denne øvelse. I stedet kan du finde koden sammen med de andre øvelser på kvindekenddinkode.dk og lege videre med den inde på App Inventor. Der er både med og uden pointtælling.