

Koding for alle – i JavaScript

Terje Kolderup

Koding for alle i JavaScript

CAPPELEN DAMM AKADEMISK

© CAPPELEN DAMM AS, Oslo, 2020

ISBN 978-82-02-67969-9

1. utgave, 1. versjon

E-utgaven baserer seg på 1. trykte utgave, 2020

Illustrasjoner: Side 119 og 204: Terje Kolderup; side 127: Wikimedia Commons/Popperipopp

Omslagsdesign: Cappelen Damm

Omslagsillustrasjon: <https://stock.adobe.com/no/enterprise-conditions>

Sats/design: [Bøk Oslo AS](#)

Elektronisk tilrettelegging: [Bøk Oslo AS](#)

Materialet i denne publikasjonen er omfattet av åndsverklovens bestemmelser. Uten særskilt avtale med Cappelen Damm AS er enhver eksemplarframstilling og tilgjengeliggjøring bare tillatt i den utstrekning det er hjemlet i lov eller tillatt gjennom avtale med Kopinor, interesseorgan for rettighetshavere til åndsverk.

Forfatteren har mottatt støtte fra Det faglitterære fond.

www.cda.no

akademisk@cappelendamm.no

Innhold

Første og andre kapittel er en introduksjon til HTML og CSS. Tredje kapittel handler om versjonsstyring og om hvordan man kan dele det man har laget med verden. Resten av kapitlene handler om JavaScript.

Alle kodeeksempler i boken finnes også på denne lenken:

getacademy.github.io/KodingForAlle/eksempler

Forord	11
Fokus på kundeverti via prototypebygging	12
Perspektivet er praktisk snarere enn teoretisk	12
Eget kapittel om problemløsningsteknikker	13
Fra nybegynner til <i>frontend</i> -utvikler	13
 1	
HTML	15
Min første nettside - ved hjelp av HTML	15
HTML	18
Vinlotterix - Prototypebygging 1 - bare HTML	19
Koderedigeringsprogram	21
 2	
CSS	23
CSS-klasser	27
Vinlotterix - Prototypebygging 2 - CSS	28
CSS - fullskjerm layout ved hjelp av CSS Grid	30
Vinlotterix - Prototypebygging 3 - CSS Grid	35
Responsivt webdesign	37

3

Å publisere det man har laget	39
--	-----------

4

Grunnferdigheter i JavaScript	49
--	-----------

 Ferdighet 1 - alert	49
 Ferdighet 2 - onclick	51
 Ferdighet 3 - Egne funksjoner	52
 Ferdighet 4 - Endre innholdet av en tagg ut fra id	53
 Ferdighet 5 - Endre CSS for en tagg ut fra id	55
 Ferdighet 6 - Enkel prototypebygging med JavaScript	56
 Vinlotterix - Prototypebygging 7 - JavaScript	58
 Ferdighet 8 - Nøkkelordet this og en forsmak på parametre	61
 Ferdighet 9 - Sette sammen tekst av flere deler	64
 Ferdighet 10 - Logging til konsollen	65
 Ferdighet 11 - Endre andre ting enn innerHTML: value, style og classList	66
 Ferdighet 12 - Alternative eventhandlere: onchange osv.	68
 Ferdighet 13 - Debugge i nettleseren Google Chrome	69
 Ferdighet 14 - Vær nysgjerrig og slå opp eller test deg frem til det du lurar på!	71

5

Variabler	73
------------------------	-----------

Lagre tall i variabler	73
Lagre tekst i variabler	75
Lagre objekter i variabler	76
Bygge HTML ved å sette sammen tekst av flere deler	78
Variabler - innenfor og utenfor funksjon	79
Sette variabel til resultat av et regnestykke	82
Sette variabel til tilfeldig tall	83

6

Valgsetninger	85
----------------------------	-----------

Hvis og ellers	86
Flere alternativer	88
Sammenlikne tekster	91
Sammenlikne objekter	93
Logisk og	97
Logisk eller og mer om logiske uttrykk	99
Logisk ikke og kombinasjoner av og og eller	102

7

Funksjoner	107
Funksjoner med returverdi	110
Scope - let og const	114

8

Problemløsningsteknikker	117
Omformuler problemet	118
Del opp problemet	121
Begynn med det som er kjent	123
Forenkle problemet	124
Se etter analogier	126
Eksperimenter	128
<i>21st century skills</i>	129

9

Model-view-controller	133
Model	134
View	135
Controller	136

10

Løkker	139
while	140
for	141
while til slutt	143

11

Lister	145
Slette eller bytte ut elementer i en liste	148
Sortere og reversere	148
Søke	149
Filtrere	150
Aggregeringer	150
Referanser	152

12

Funksjonell programmering	153
Funksjonene <code>filter()</code> , <code>map()</code> og <code>reduce()</code>	156

13

Objekter	159
Lister med objekter	163
Vinlotterix	168
Slette eller bytte ut elementer i en liste	171
Sortere og reversere	172
Søke og filtrere	175
Lage en liste basert på en annen liste	175
Aggregeringer	176

14

Tilbakekall, promises og asynkrone kall	177
Nettarbeider	182
Promises	185
Nøkkelordene <code>async</code> , <code>await</code> , <code>try</code> og <code>catch</code>	188

15

Backend med Google Firebase	191
<code>onSnapshot</code>	196
Endre og legge til dokumenter	197
Lese flere dokumenter fra en collection	198
Vinlotterix	200

16

Mer objektorientert JavaScript med klasser	203
Innkapsling	207
Samarbeid mellom klasser	212
Arv	217

17

En liten webhistorie	219
Statiske nettsider	219
Dynamiske nettsider	220
SPA	221

18

SPA-rammeverket Vue.js	223
Vinlotterix med browser-Vue	228
Komponenter og egenskaper	233
Vinlotterix med browser-Vue og komponenter	237
Hendelser	244
Vue-filer, Webpack og Node.js	246
Data på toppnivå med hendelser og egenskaper	254
Vinlotterix med vue-filer, Webpack og Node.js	257

19

REST API	265
Sende med data	267

20

Kodegenerering og metadata	269
Metadata	274
Metadata til brukergrensesnitt	280
Produktkatalog	284

Stikkordregister	297
-------------------------------	---------------------

Forord

Målet med denne boken er først og fremst å introdusere programmering til interesserte som ikke har noen forkunnskaper. Da er det mange språk å velge mellom. Dette er årsakene til at JavaScript er valgt i denne boken:

1. Skriptspråk er enklere å lære enn kompilerte og sterkt typede språk som C# og Java.
2. Man trenger bare en nettleser for å kjøre programmet sitt, og det har de fleste fra før.
3. Sammen med HTML og CSS kan man lett lage morsomme og visuelt tiltalende applikasjoner.
4. Når man bygger for web, er det lett å dele det man har laget, med alle man kjenner, via en lenke. (Mange nettsteder tilbyr gratis vertstjenester, *hosting* på engelsk, for eksempel GitHub Pages – pages.github.com.)
5. Kompetansen man bygger, er verdifull i arbeidsmarkedet. På grunn av populariteten til webapplikasjoner bør alle utviklere kunne HTML, CSS og ikke minst JavaScript.

Forfatteren underviser i programmering til daglig, og progresjonen i boken er et resultat av erfaringene fra dette. Stoffet er delt opp i så små skritt at alle kan oppleve progresjon. Bokens mål er å komme over kneika det er å lære seg grunnleggende programmering. Det gir den en bred målgruppe:

- studenter på fagskoler, høyskoler og universiteter
- elever med valgfag programmering i grunnskole og videregående skole
- deltakere på Lær kidsa koding (en bevegelse som jobber for å lære barn IT og programmering)
- alle som vil skaffe seg grunnleggende programmeringskompetanse – uansett om det er på fritiden eller i jobben

De viktigste fortrinnene som denne boken har, sammenlignet med andre introduksjonsbøker til programmering, er følgende:

1. den tar små skritt,
2. har fokus på kundeverti via prototypebygging,
3. har et mer praktisk enn teoretisk perspektiv,
4. har eget kapittel om problemløsningsteknikker, og
5. passer for nybegynnere til *frontend*-utviklere.

Fokus på kundeverti via prototypebygging

På hvert nivå viser vi hvordan vi kan levere verti for en kunde ved å lage prototyper. Dette er en smart måte å jobbe på også i virkelige prosjekter, for det er ofte uklart for kunden nøyaktig hvordan applikasjonen som ønskes, skal se ut og oppføre seg. Å lage det i en prototypeversjon er mye raskere enn å lage det helt ferdig, og man kan dermed gå raskere til kunden og få tilbakemelding på om man er på rett spor. Og om man ikke er det, har man ikke kastet bort mye tid.

Gjennom slik jobbing med prototyper bygger man også en felles forståelse med kunden. Man finner et felles språk som lar begge parter forstå hverandre, og man blir enige om hva som skal lages. Selve prototypen er ofte i seg selv en god beskrivelse av hva som skal gjøres. Den hindrer at utviklerne svir av mange timer på noe kunden ikke ønsker. I prinsippet kan den være en del av en kontrakt mellom kunde og leverandør.

Perspektivet er praktisk snarere enn teoretisk

Pedagogikken er enkel. Først viser jeg *hvordan* man får til noe, deretter forklarer jeg hvorfor og hvordan det fungerer. Det er normalt at man ikke forstår etter bare å ha lest om det. Det viktigste er å få nok hjelp til å komme i gang med å *gjøre det selv*. Da får man erfaringer og læring.

Det å lese noe nytt i en bok kan sammenliknes med å støpe i betong. Først er betongen flytende og kan ikke bære noe som helst. Å få egne erfaringer er herdingen, det som får betongen til å størkne. Først når den har størknet, kan man bygge oppå den. Slik er det med denne boken også. Å lese hele boken uten

å programmere selv, blir som å legge lag på lag med flytende betong. Les en porsjon. Prøv det selv. Lek med det og få erfaringer. Etter en stund kan du lese en ny porsjon.

Eget kapittel om problemløsningsteknikker

Det er helt vanlig å stange hodet i veggen når man programmerer. Denne boken har et eget kapittel med helt konkrete teknikker man kan bruke når man står fast.

Fra nybegynner til *frontend*-utvikler

Boken krever ingen forkunnskaper, men dekker likevel det aller meste man må kunne for å jobbe som *frontend*-utvikler. Går du på finn.no og ser etter jobb som IT-utvikler, er *frontend*-utvikler en av underkategoriene. En slik jobb krever selvsagt et bredt spekter av kompetanse, men alt det viktigste er dekket i denne boken.

HTML

Når vi ser på nettsider, gjør vi det ved hjelp av en nettleser. Jeg bruker Google Chrome. Andre eksempler på nettlesere er Firefox, Internet Explorer og Safari.

Du kan bruke hvilken du vil, men aller enklest blir det om du bruker samme nettleser som meg. Den er gratis, og du kan laste den ned fra **www.google.com/chrome**.

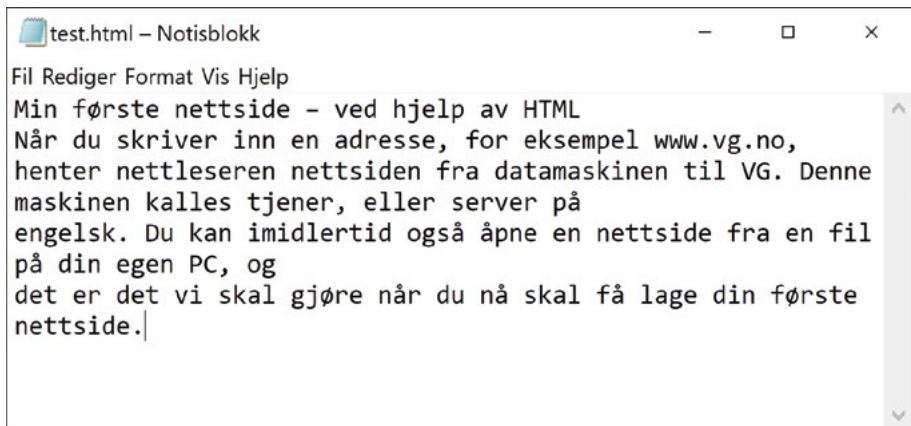
Min første nettside – ved hjelp av HTML

Når du skriver inn en adresse, for eksempel www.vg.no, henter nettleseren nettsiden fra datamaskinen til VG. Denne maskinen kalles tjener, eller *server* på engelsk. Du kan imidlertid også åpne en nettside fra en fil på din egen PC, og det er det vi skal gjøre når du nå skal få lage din første nettside.

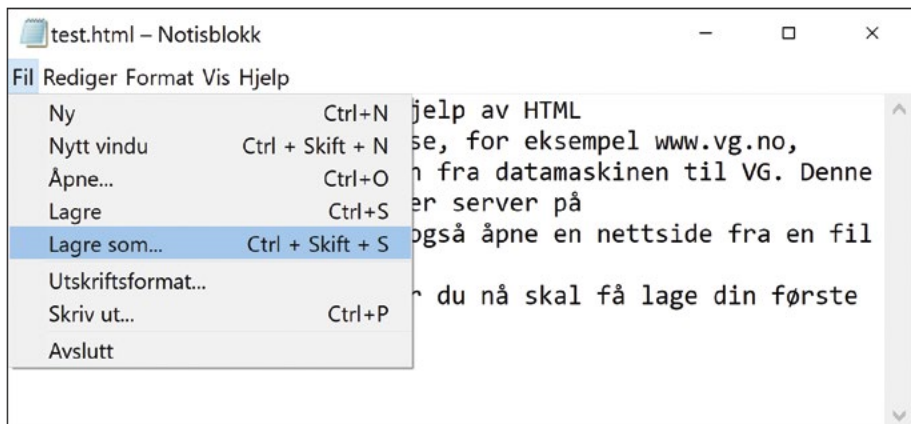
Til å lage denne filen trenger du et redigeringsprogram for tekst. Ikke en tekstbehandler, som Word, men et enda enklere program som lar deg redigere rene tekstfiler uten formatering. Windows har et innebygget program som heter Notisblokk (Notepad om du har engelsk Windows). På Mac er det innebygget et tekstredigeringsprogram som heter TextEdit.

Det beste er å laste ned og installere et eget redigeringsprogram som er laget for å redigere kode. Visual Studio Code er gratis og veldig bra. Det kan lastes ned fra **code.visualstudio.com**. Det er likevel fullt mulig å bruke de helt enkle redigeringsprogrammene.

Start redigeringsprogrammet, og skriv eller kopier litt tekst i vinduet. Jeg kopierte inn litt tekst fra denne boken, og under ser du hvordan det ser ut:



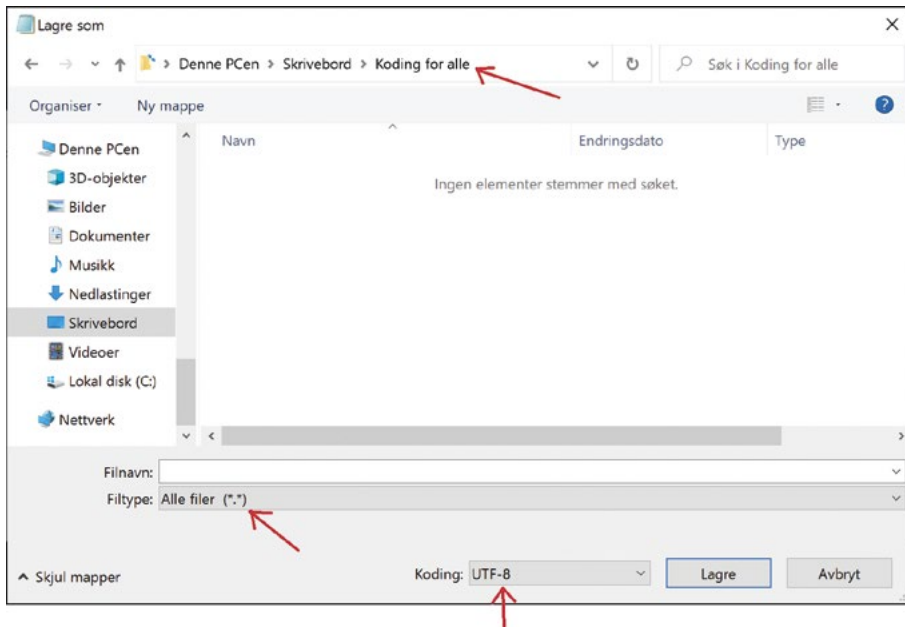
Dette er din første nettside! For at vi skal kunne se den i nettleseren, må vi først lagre den som en fil. Åpne Fil-menyen, og velg **Lagre som...** som vist på bildet under.



Velg så et filnavn, for eksempel **test.html**. Filnavnet må slutte på **.html**. I tillegg må du velge:

1. filtype: alle filer
2. koding: UTF-8
3. hvor filen skal lagres, altså en mappe

Alle de tre tingene markert med rød pil i bildet på neste side:



Til slutt trykker du på **Lagre** nederst, nest lengst til høyre. Neste skritt er å åpne denne filen i Google Chrome eller en annen nettleser. Start nettleseren din. Hold Ctrl-tasten inne, og trykk på **O**. Da får du opp en dialogboks for å velge fil. Gå til mappen du valgte over, og velg filen din **test.html**, eller hva du eventuelt kalte den. Under ser du hvordan det ser ut hos meg.



Legg merke til at formateringen fra tekstdokumentet har blitt borte. Teksten «flyter» sammen.

HTML

I HTML har ikke linjeskiftene i tekstfilen så mye å si. Vi bruker tagger. En tagg består av start-tag, innhold og slutt-tag. En tagg kan for eksempel være `h1`, som vi kan tenke står for «header 1» eller «overskrift nivå 1».

Prøv å bytte ut første linje i html-filen din med dette:

```
<h1>Min første nettside</h1>
```

Lagre filen. Gå tilbake til nettleseren, og klikk F5 for å lese inn filen på nytt. Hos meg ser det da slik ut:



Du har nå skrevet din første HTML-tag! Prøv selv taggene `p`, `b` og `i`. De står for *paragraph* (avsnitt), *bold* (fet skrift) og *italic* (kursiv). Ifølge HTML-standarden er det noen tagger alle HTML-dokumenter skal ha. Dette er vist under:

```
<!DOCTYPE html>
<html>
<head>
  <title>Arkfaneoverskrift</title>
</head>
<body>
  <h1>Overskriften</h1>
  <p>Dette er et avsnitt.</p>
  <p>Dette er et avsnitt.</p>
</body>
</html>
```

Alt som skal vises på skjermen ligger inne i body-taggen. Taggen `p` står for *paragraph*, eller avsnitt på norsk. Taggen `head` kan inneholde flere overordnede ting. Vi ser mer på det senere i boken. Et eksempel er `title`, som styrer hva som står på nettleserfanen.

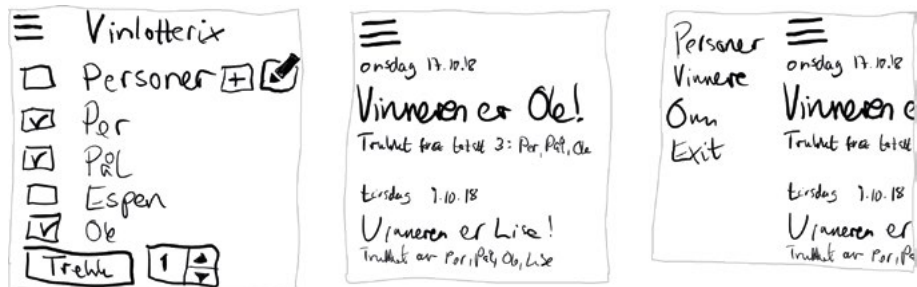
Både `body` og `head` ligger inne i en `html`-tag, og øverst ligger det en såkalt HTML-deklarasjon som forteller nettleseren hva slags type dokument dette er.

Vinlotterix – Prototypebygging 1 – bare HTML

Vi skal følge en case gjennom hele denne boken. Først skal vi lage en prototype, og deretter skal vi videreutvikle denne til en funksjonell applikasjon. Vi trenger et relativt enkelt oppdrag, som likevel inneholder nok kompleksitet.

Mange bedrifter har noe de kaller vinlotteri. Gjennom arbeidsuken kjøper alle lodd. Torsdag etter jobb samler ukens ansvarlige alle pengene og drar på vinmonopolet og kjøper inn vin. På fredagen gjøres det en trekning for hver flaske. Vi skal lage en applikasjon som kan trekke vinnerne, og den skal hete Vinlotterix.

Første skritt er å lage prototype på papir. Tegn opp skjermbildene helt konkret. Her er mitt første utkast:



Så kan vi bruke HTML til å lage en såkalt *statisk prototype*. Det er altså fortsatt bare en tegning, men vi bruker HTML istedenfor blyant.

≡ Vinlotterix ☐ Personer +  ☒ Per ☐ ☒ Pål ☐ ☐ Espen ☐ ☒ Ole ☐ Trekk! 1 ▲ ▼	≡ Vinlotterix onsdag 17.10.18 Vinneren er Ole! Trukket fra totalt 3 personer: Per, Pål og Ole	Personer Vinnere Om Exit	≡ Vinlotterix onsdag 17.10 Vinneren Trukket fra
--	---	---	--

Her kom jeg på at det kan være lurt å kunne ta bort en person helt også. Derfor er det med et kryss i en firkant etter hvert navn.

HTML-koden ser slik ut for skjermbilde 1:

[illegible]

Merk at det her brukes en del spesialtegn. Søk etter «unicode symbols» på Google, og du vil få opp hvilke tegn som finnes.

Man bruker en HTML-tabell bare for å få en ramme rundt.

Non-break-space eller er et «hardt» mellomrom, det vil si et som fyller plass uansett. Det er en primitiv måte å høyrejustere noe på. I neste kapittel skal vi lære bedre teknikker for rammer og justeringer, men med bare HTML er dette det beste vi har.