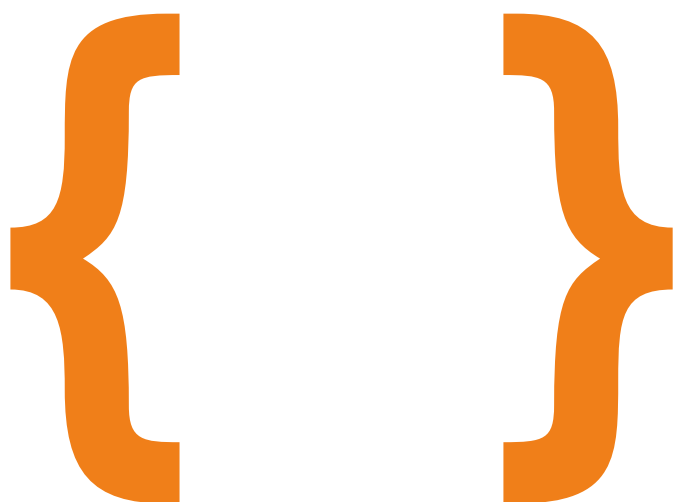


TOM HEINE NÄTT
JOSTEIN NORDENGEN

Oppgavesamling til Programmering i JavaScript

Oppdatert 23.12.2016



DATASERVICE
.NET


GYLDENDAL
UNDERVISNING

1. Introduksjon

1.1

Forklar med dine egne ord følgende begreper og programmeringselementer:

- Programmeringslogikk
- Instruksjon
- Hendelsesorientert programmering
- Utviklingsmiljø
- Syntax highlighting
- Kommentar
- `<script>`
- `innerHTML`
- `window.onload`
- camel case
- case-sensitiv

1.2

Ta for deg koden vi hadde for nettsiden som viste klokka. Forsøk å lage ulike feil i koden (som for eksempel å fjerne tegn, legge til tegn osv.), og bli kjent med feilmeldingene som da oppstår.

2. HTML, CSS og JavaScript

2.1

Forklar med dine egne ord følgende begreper og programmeringselementer:

- `onclick`
- `getElementById`
- Hendelsesparameter
- `createElement` og `appendChild`
- `<canvas>`

2.2

Lag en helt enkel nettside med to ulike knapper og en paragraf. Registrer en hendelse på hver av knappene. Når du trykker på den første knappen, skal paragrafen vise *Hei på deg*, og med den andre knappen skal *Ha det bra* vises.

2.3

Lag en nettside med flere spørsmål. Hvert spørsmål skal være på sin egen linje, og hver linje skal i tillegg til spørsmålet inneholde en "Sann" og en "Usann" knapp. Når du trykker på knappene skal en meldingsboks vise "Riktig" eller "Galt" ettersom hvilken knapp man trykket på.

2.4

Lag et skjema der du kan skrive inn to tall i to tekstbokser, og så få ut summen(+), differansen(-), produktet(*) og kvotienten(/) når man trykker på en av tilsvarende fire knapper.

2.5

Lag et "nummerisk tastatur" ved å plassere ut 10 knapper med teksten 0,1,2,3 osv. Lag så et tekstfelt, der du kan skrive inn tall som 4327 ved å trykke på knappene. Hver gang du trykker på en knapp, skal altså sifferet som står på knappen legges til i tekstboksen.

2.6

Bruk funksjonene i et canvas og tegn et hus med vindu, dør og trekantet tak ved hjelp av JavaScript. Før du skriver koden, kan det være lurt å tegne huset inn i koordinatsystemet på et ark. Husk at x-aksen peker mot høyre og y-aksen peker nedover.



2.7

Hvorfor er det viktig å skille utsende, innhold og funksjonalitet fra hverandre?

2.8

Hvorfor skal man ikke benytte JavaScript til å lage sikkerhetskritisk funksjonalitet?

2.9

Lag et lite program for å lage deg en handleliste. Programmet skal bestå av en tekstboks (<input>), en knapp(<button>) og en liste(). Hver gang det trykkes på knappen skal innholdet i tekstboksen legges til i lista, og tekstboksen skal tømme (sette egenskapen value til å være "")

3. Variabler og operatører

3.1

Forklar med dine egne ord følgende begreper og programmeringselementer:

- Variabel
- Deklarere
- Initialisere

- Variabelnavn
- Datatype
- string
- number
- boolean
- Virkningsområde
- Operator
- Presedens
- Modulo
- Inkrementering
- Dekrementering
- Konkaternering
- Konstant

3.2

a) Hvilke av disse variabelnavnene er ugyldige og hvorfor?

b) Hvilke av disse variabelnavnene bryter mot den anbefalte måten i skrive variabelnavn på, og hvorfor?

- minFineVariabe
- endaenvariabel
- 1001Natt
- min_Variabel
- kvartPå10
- minFineSide.no
- MinVariabel
- @Variabelen
- 90
- antallGangerUtført

3.3

Lag en applikasjon som leser inn en alder fra en tekstboks og viser hvor gammel du er om 5 år i en utskrift.

3.4

Lag et program som beregner utslipp av CO₂ fra en bil, per person i bilen. Brukeren skal kunne oppgi drivstofforbruket på bilen og hvor mange personer som er i bilen. Utslippet beregnes ut fra at bilen slipper ut 2500 gram CO₂ når den bruker 1 liter brennstoff. Benytt variabler for å lagre mellomregninger.

3.5

Fortsett med CO₂-programmet og vis en sammenligning av forbruket med andre transportmidler. Fly har et forbruk på 180 gram, tog bruker 70 gram, og buss bruker 28 gram.

3.6

Lag en enkel "kalkulator" der du kan skrive inn en verdi uten merverdiavgift i en tekstboks, trykke på en knapp, og så få ut verdien med merverdiavgift.

3.7

Utvid forrige oppgave med en ekstra knapp som gjør nesten det samme, men som går ut i fra at verdien i tekstboksen er med merverdiavgift, og dermed viser verdien uten merverdiavgift.

3.8

Modifiser eksempelet Historiegenerator slik at det forteller en annen og potensielt mye bedre historie ved hjelp av andre verdier og skjemaelementer.

3.9

Lag en kalkulator som utfører beregninger basert på brukerdata. Du velger selv hva som skal beregnes. Benytt variabler for å mellomregninger. For eksempel:

- BMI
- Skatt
- Importavgift eller omregistreringsavgift på bil
- Bit/s til Byte/s omformer
- Rentekalkulator
- Osv.

3.10

Lag en nettside der du kan skrive inn et tall i en tekstboks, og så addere dette til en sum i som lagres i en variabel når man trykker på en knapp. Totalsummen skal hele tiden vises i en paragraf.

4. Valgsetninger

4.1

Forklar med dine egne ord følgende begreper og programmeringselementer:

- Valgsetning
- Kontrollstruktur
- Betingelse
- If-test
- Kodeblokk
- true og false
- Valgoperator
- Logiske uttrykk
- Relasjonsoperator
- AND-operator
- OR-operator
- Negeringsoperator

4.2

Lag et program hvor brukeren kan skrive inn svaret på en matteoppgave som vises. Hvis svaret er riktig, skal teksten riktig vises i en paragraf, om ikke skal teksten feil vises.

4.3

Hva vil svaret (true/false) på disse logiske uttrykkene bli, dersom variablene x, y og z er definert som følger:

```
var x = 45;  
var y = 145;  
var z = 78;
```

- a) $x < 45$
- b) $x > y$
- c) $x > y - 100$
- d) $z > x \ \&\& \ z < y$
- e) $!(z > x \ \&\& \ z < y)$
- f) $z === x \ || \ z !== y$

4.4

Dersom vi fortsatt benytter variablene fra forrige oppgave, skriv logiske uttrykk for:

- a) x ulik y
- b) x lik y
- c) x lik y eller y større enn z
- d) x er større enn y og y er større enn z
- e) x er lik y og y er lik z ELLER z mindre enn x
- f) x lik y OG y lik z eller z større enn eller lik x

4.5

Gå ut i fra at du har følgende variabler definert (verdiene er eksempelverdier) som forteller om ulike sider ved været for foregående time (gjennomsnittsverdier).

```
var temperatur = 12.3;    // angitt i °C
var vindhastighet = 4.6; // angitt i m/s
var vindretning = "NØ"   // forkortelse for nord-østlig
var nedbor = 3.7         // angitt i antall mm
```

Skriv logiske uttrykk for følgende påstander:

- a) Det regner og er vindstille
- b) Det blåser nord-, nordøst- eller nordvestlig vind
- c) Det regner mer enn 5 mm ELLER det er frost og sørøstlig vind
- d) Det er kuling (13.9-17.1 m/s) ELLER orkan (over 32.6 m/s)

4.6

En forening har ganske spesielle opptakskrav. Vi ønsker nå å skrive et logisk uttrykk som angir om man er ønsket som medlem eller Ikke ønsket som medlem ut i fra følgende variabler:

```
var alder;           // angitt i antall hele år
var mann;            // verdien true angir mann og false angir kvinne
var hoyde;           // angitt i antall cm
```

Opptakskrav:

Man må være mann mellom 16 og 18 år ELLER kvinne som er mer enn 190 cm høy og nøyaktig 25 år gammel.

4.7

Ta utgangspunkt i det logiske uttrykket du skrev i forrige oppgave. Lag nå et enkelt brukergrensesnitt med en tekstboks for hver av variablene. For variabelen `mann`, kan du benytte en avkryssningsboks.

Sett også inn en knapp og en tekstboks etter disse inputfeltene, der du viser teksten *Ønsket som medlem* eller *Ikke ønsket som medlem* avhengig av om det logiske uttrykket stemmer eller ikke.

4.8

Gjett-på-tallet-spillet vi lagde som eksempel i dette kapittelet, har en stor svakhet ved at det ikke går an å starte spillet på nytt. Legg til en knapp med teksten *Start på nytt*, som starter spillet på nytt ved å sette `hemmeligTall` til et nytt tilfeldig tall, sette `forsokIgjen` tilbake til 10 og fjerne meldingen.

4.9

Lag en tekstboks, en knapp og en paragraf der du kan skrive inn et tall i tekstboksen og så få ut informasjon om dette tallet er positivt eller negativt i paragrafen.

4.10

Lag en tekstboks, en knapp og en paragraf der du kan skrive inn et tall i tekstboksen og så få ut informasjon om dette tallet er et partall eller oddetall i paragrafen. *Tips:* For å sjekke om et tall er partall, kan du benytte deg av modulusoperatoren. Dersom `tall % 2 == 0`, altså at vi ikke får noen rest når vi deler tallet på 2, så er det partall.

4.11

Lag to tekstbokser, en knapp og en paragraf. I paragrafen skal det skrives ut hvilket av tallene som blir skrevet inn i tekstboksene som er størst.

4.11

Lag en nettside som skal ta imot antall sekunder i en tekstboks. Utskriften til en paragraf skal være: [sekunder totalt] tilsvarer: [timer], [minutter] og [sekunder]. F.eks.: *7381 sekunder tilsvarer: 2 timer, 3 minutter og 1 sekund*. Pass på å få riktig syntaks på *time/timer, minutt/minutter, sekund/sekunder*.

4.12

Lag en nettside der man kan skrive inn navnet på en måned, og så få ut antallet dager i den valgte måneden i en paragraf når man trykker på en knapp.

4.13

Følgende kodebit gir deg en variabel som inneholder hvilken time i døgnet det er (0-23):

```
var tid = new Date();  
var timer = tid.getHours();
```

Lag en nettside som under oppstart setter en hyggelig velkomstmelding basert på tiden i en paragraf. Dette kan f.eks være "God morgen", "God ettermiddag" osv.

4.14

En spådamme ønsker å reise vekk på ferie, og vil derfor automatisere sine tjenester via en nettside. Du har fått i oppgave å lage denne funksjonaliteten.

Lag en nettside for å gi en spådom, hvor man kan skrive inn verdier i tekstbokser for navn, alder og høyde. I tillegg skal det være en nedtrekksliste for kjønn.

Spådommen skal vises på nettsiden når brukeren trykker på en knapp, og vil være basert på en magisk verdi. Denne verdien vil regnes ut forskjellig, avhengig av om det er en gutt eller en jente.

For gutter: $\text{<lengde på navnet>*<alder> - <hoyde>}$

For jenter: $\text{<alder>*<hoyde> - 3}$

Selve spådommen genereres etter følgende regel:

Dersom den magiske verdien er et partall (delelig med to): "Det vil gå deg godt her i verden... " "For at spådommen skal gå i oppfyllelse, må du betale inn 100 kr til følgende kontonummer: 1234.12.12345"

Dersom den magiske verdien er et oddetall: "Stakkars deg! Alt kommer til å gå deg galt..." "For at spådommen ikke skal gå i oppfyllelse, må du betale inn 100 kr til følgende kontonummer: 1234.12.12345"

Tips: Merk deg at det eneste som skiller den andre setningen i de to utfallene er ordet ikke...

5. Løkker

5.1

Forklar med dine egne ord følgende begreper og programmeringselementer:

- Løkke
- while-løkke
- for-løkke
- Uendelig løkke
- Iterasjon
- Nestet kontrollstruktur
- break
- continue

5.2

Forklar forskjellen og likheten mellom for-løkker og while-løkker. Hvilken tommelfingerregel finnes for når hver av dem bør bli brukt?

5.3

Lag for-løkker som lister ut følgende tallrekker mellom 0 og 100:

- a) Alle partall
- b) Alle oddetall
- c) Alle tall som er delige med 5
- d) Alle tall som er delige med 5 og partall
- e) Alle tall som er delig med 8 eller 3

5.4

Lag to tekstbokser, en paragraf og en knapp.

Når man trykker på knappen skal teksten i den første tekstboksen skrives ut så mange ganger (linje for linje) som man skrev et tall i den andre tekstboksen.

5.5

Skriv kode som produserer følgende utskrift (teksten "..." skal selvsagt erstattes med de virkelige utskriftene fra 8 til 99).

```
0*0=0
1*1=1
2*2=4
3*3=9
4*4=16
5*5=25
6*6=36
7*7=49
...
100*100=10000
```

5.6

En svakhet ved generatoren for sangen 5 monkeys jumping on the bed er at brukeren ikke selv kan velge hvor mange aper det var i utgangspunktet. Det hadde vart mer hensiktsmessig om brukeren kunne skrevet inn antallet i en tekstboks, og så trykket på en knapp for å generere sangen.

Gjør derfor denne endringen i eksempelet. Du må da flytte løkka fra `onload`-hendelsen til knappens `onclick`-hendelse, samt benytte verdien fra tekstboksen som startverdi i `for`-løkka.

5.7

Lag følgende grafiske brukergrensesnitt.

The image shows a simple web form. It has three text input fields stacked vertically. The first is labeled 'Pengebeløp:', the second 'Antall år:', and the third 'Bæste:'. Below these fields is a button labeled 'Regn ut'. Under the button is a large, empty rectangular area with a vertical scrollbar on the right side, intended for displaying the output of the calculation.

Lag deretter en *for*-løkke i knappens `onclick`-hendelse, som skriver ut linjer slik.
Etter 1 år har du xxx penger i banken.

Etter 2 år har du xxx penger i banken.

...

Etter yyy år har du xxx penger i banken.

Husk at i tekstbokser må vi benytte `\n`, og ikke som ellers, for å markere linjeskift.

5.8

a) Skriv ut alle tosifrede kombinasjoner av tallene 1, 2, 3, 4 og 5 slik som 11, 12, 13, 14, 15, 21, 22 osv. Et tips vil være å bruke nestede løkker, som begge løper over intervallet.

b) Legg til nok en nestet løkke med samme intervall, og skriv ut alle tresifrede kombinasjoner.

5.9

Lag/finn 10 bildefiler navngitt bilde1.jpg, bilde2.jpg, bilde3.jpg osv, og plasser disse filene i prosjektmappa.

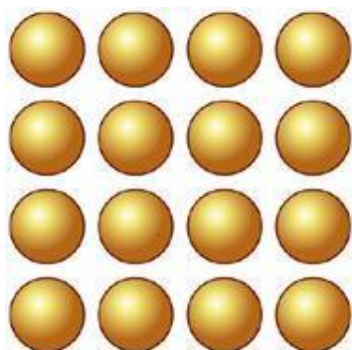
La så brukeren skrive enn et fra-nummer og et til-nummer i to tekstbokser, og vis de valgte bildene når en knapp trykkes.

TIPS: Benytt her en for-løkke til å legge til -tagger i dokumentet.

5.10

Lag en for-løkke inne i en annen for-løkke. Skriv kode i den innerste for-løkka slik at det blir tegnet sirkler som blir plassert i rader og kolonner:

```
for(var i = 0; i < 4; i++) {  
  for(var j = 0; j < 4; j++) {  
    // din kode her  
  }  
}
```



5.11 (Nøtt)

Lag en nettside der man kan vise en kalender på følgende form:

```
ma ti  on to  fr  lø  sø
1   2
3   4   5   6   7   8   9
10  11  12  13  14  15  16
17  18  19  20  21  22  23
24  25  26  27  28  29  30
31
```

Man skal her kunne opp gi hvilken ukedag kalenderen skal begynne på, og hvor mange dager det skal være i måneden.

TIPS: Benytt her en for-løkke og sett inn en `
` på strategiske plasser ved å benytte bl.a. `teller%7===0` som en test. For å få lik avstand mellom tegnene, så benytt fonten courier new eller lignende.

6. Arrayer

6.1

Forklar med dine egne ord følgende begreper og programmeringselementer:

- Array
- Indeks
- splice
- indexOf
- Flerdimensjonal array
- Assosiativ array
- Nøkkel
- split
- join

6.2

Gjør de endringene som trengs i eksempelet *statistikk over terningkast* for at det heller skal vise en statistikk mellom kron og mynt. Du kan fortsatt behandle utfallene som tall, der kron er 0 og mynt er 1. Det må imidlertid gjøres en "oversettelse" av verdiene til tekst under utskriften.

Tips: En ytre løkke styrer hvilket bokstavnummer du er på, en indre løkke skriver ut et antall bokstaver.

6.7

Lag en array som inneholder heltall. Skriv så ut alle heltall som forekommer flere ganger i arrayen, samt indeksene det gjelder. Sørg også for at samme kombinasjon kun listes ut en gang.

TIPS: Benytt to nestede løkker som sjekker om to ulike indexer inneholder samme verdi.

6.8

Lag et brukergrensesnitt med en tekstboks, en knapp og en etikett. Lag så en array med antall dager i hver måned som verdier. *Januar* skal da være på indeks 0, *februar* på indeks 1 osv. I tekstboksen skal du så kunne skrive inn nummeret på en måned og få en utskrift med antall dager i denne måneden når du trykker på knappen. Du kan se bort fra skuddår i denne oppgaven.

6.9

Lag et brukergrensesnitt med en tekstboks, en knapp og en paragraf for utskrift. Lag så en array med heltall. Skriv til slutt programkode til knappens `onclick`-hendelse som sjekker om et tall du skriver i tekstboksen, finnes i arrayen eller ikke.

6.10

Gå ut i fra at du har definert en array med navn på studenter. Denne definisjonen kan for eksempel se ut som følger:

```
var arr =  
["Hans", "Ole", "Nils", "Olav", "Per", "Knut", "Kari", "Line", "Pia"];
```

Skriv kode som plukker ut tre tilfeldige navn fra denne arrayen (Tips: `Math.random()`), og legger disse i en egen/ny array. Skriv deretter ut denne nye arrayen. Du må også sørge for at samme navn ikke blir plukket ut flere ganger.

6.11

Lag en enkel webside som inneholder en array med bokstaver (som til sammen former et ord). La så brukeren gjette på bokstaver i en tekstboks, og få tilbakemelding om bokstaven forekommer i ordet eller ikke.

6.12 (nøtt/stor)

Utvid forrige oppgave videre til å bli en enkel versjon av hangman. Lag en array med så mange plasser som det skal være forsøk, og ta vare på forsøkte bokstaver i denne. Vis så hele tiden de bokstavene av det hemmelige ordet som også finnes i arrayen med

forsøkte bokstaver. Om du har en variabel som inneholder antall feil, kan du lage og vise bilder med filnavn på følgende form: feil1.jpg, feil2.jpg, feil3.jpg osv.

6.13

Du har følgende tre assosiative arrayer, som inneholder en kobling mellom alle bokstaver i alfabetet og et ord. Under vises kun A til D for hver array for å spare plass:

```
var arr1 = {A: "gale", B: "snartenkte", C: "stinkende", D: "rykende" };  
var arr2 = {A: "buss", B:"nese", C: "glass", D: "tomat"};  
var arr3 = {A: "muffins", B: "tastatur", C: "kopp", D: "vase"};
```

Din oppgave blir nå å lage en "kallenavngenerator", der brukeren skal taste inn sitt virkelige navn i et skjema som ser slik ut:

Kallenavngenerator:

Fornavn:

Etternavn:

Systemet skal så plukke ut:

Første bokstav av fornavn: B

Første bokstav av etternavn: D

Siste bokstav av etternavn: A

Disse tre bokstavene skal så slås opp i hver sin array, for å hente ut de tre delene kallenavnet skal bestå av:

B i arr1 gir "snartenkte"

D i arr2 gir "tomat"

A i arr3 gir "muffins"

Delen fra arr1 skal gi det nye fornavnet. De to delene fra arr2 og arr3 skal settes sammen og gi det nye etternavnet: Fornavn: "Snartenkte" Etternavn: "Tomatmuffins" Til slutt skal det skrives ut en melding til brukeren:

Velkommen Bjarne Dunna! Du er herved døpt Snartenkte Tomatmuffins.

6.14

Ta utgangspunkt i at du har en array med alle bokstaver i det engelske alfabetet:

```
var alfabet = ["A", "B", "C", "D", "E", osv, "Z"];
```

Du har også en array som inneholder et utvalg bokstaver. Denne kan f.eks se slik ut:

```
var liste = ["F", "B", "C", "F", osv, "P"];
```

Lag programkoden som skriver ut alle bokstavene i alfabetet som IKKE forekommer i arrayen liste.

TIPS: Gå gjennom arrayen alfabet. For hver bokstav, gå gjennom arrayen liste og se om bokstaven finnes der.

6.15

La brukeren skrive inn en tekst som består av kommaseparerte tall mellom 0 og 9. Slik som:

0,4,4,2,3,1,4,6,8,4,2,1,4,5,3,2,1,3,4,8,8,5

Splitt så denne tekststrengen på komma, slik at du får en array med verdier.

a) Benytt en ny array til å telle opp hvor mange ganger hvert tall mellom 0 og 9 forekommer (tips: tallene er indeksene i denne nye arrayen, og verdien er antallet), og lag så en utskrift på dette.

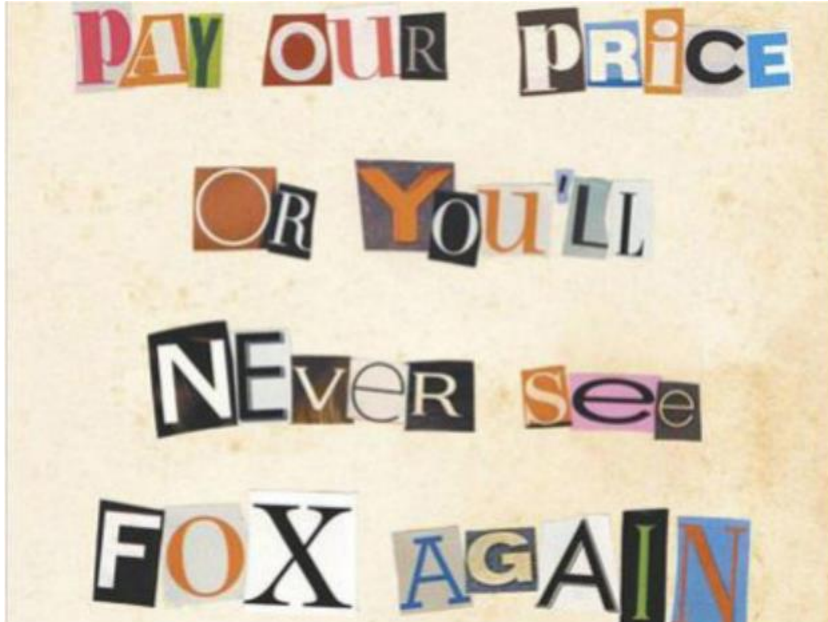
b) Presenter en utskrift av resultater. Vil du gjør litt ekstra så presenter resultatet som et liggende søylediagram, med ulik farge på partall og oddetall (altså tallene 0-9). Tips: Du kan lage hver søyle ved å lage en -tagg av et bilde på 1x1 pixel, og så sette bredden på dette bildet. Altrnativt kan du sette inn en du stilsetter med CSS.

6.16

Lag en webside der brukeren kan skrive inn en tekst, og så få ut en oversikt over hvor mange ganger de ulike ordene i teksten forekommer. (Tips: Dette vil kreve en assosiativ array, ettersom dere ikke på forhånd vet hvilket utvalg med elementer som finnes). Lag også en statistikk over hvor ofte ordene forekommer. Om du ønsker en ekstra utfordring, kan du også f.eks. vise de ti mest hyppige ordene, lengste og korteste ord osv. Bruk et liggende søylediagram for å presentere statistikken.

6.17

Len "kidnappingsbrevgenerator". Brukeren skal kunne skrive inn en tekst og generere et kidnappingsbrev av denne teksten. Det er da her tenkt att dere skal erstatte bokstaven i teksten med ett bilde av den samme bokstaven så det ser noe slikt ut (se ill. under)



7. Funksjoner

7.1

Forklar med dine egne ord følgende begreper og programmeringselementer:

- Funksjon
- Funksjonskall
- Funksjonsdefinisjon
- Parameter
- Argument
- Returverdi
- Globale og lokale variabler
- Rekursiv funksjon
- pow
- sqrt
- random

7.2

Lag ferdig følgende funksjonsdefinisjoner som :

a) konverterer fra tommer til centimeter (1 tomme = 2,54 cm)

```
function fraTommerTilCm(tommer)
```

b) sjekker om et tall er innenfor et intervall

```
function innenforIntervall(tall, fra, til)
```

c) finner gjennomsnittet av to tall

```
function gjennomsnitt(tall1, tall2)
```

d) finner ut om et tall er partall (Tips: Et tall er partall om det gir 0 i rest når vi skriver $\text{tall} \% 2$.)

```
function erPartall(tall)
```

e) finner absoluttverdien av et tall (Absoluttverdien er selve tallverdien uten fortegn, slik at 7 og -7 begge returnerer 7). Du får her ikke benytte deg av den innebygde abs-funksjonen.

```
function absoluttverdi(tall)
```

f) gir deg et tilfeldig tall mellom 0 og 100

```
function tilfeldigTallFraNullTilHundre()
```

g) gir deg et tilfeldig tall mellom fra og til

```
function tilfeldigTall(fra, til)
```

7.3

Lag et prosjekt der du får testet hver av funksjonene i oppgave 7.2. Prosjektet trenger ikke gjøre noe fornuftig.

7.4

Lag en funksjon som sjekker om en dato er gyldig basert på parameterne *dag*, *maaned* og *aar*. Du trenger ikke bekymre deg for skuddår i denne oppgaven, kun antall dager i hver måned under normale år. Gyldige år er mellom 0 og 2100.

Lag et brukergrensesnitt der du benytter funksjonen for å sjekke at en dato brukeren oppgir er gyldig eller ikke. Du kan benytte tre ulike tekstfelt (dag, måned, år).

7.5

Det er også mulig å lage funksjoner med arrayer som parametere. Starten på funksjonsdefinisjonen kan da for eksempel se ut som følger:

```
finnAntallNegativeTall(talliste)
```

Lag følgende funksjoner som arbeider med en parameter av typen array med heltall . Du skal til hver funksjon også lage et eksempel som viser bruken av den.

- a) Funksjon som finner summen av arrayen
- b) Funksjon som finner gjennomsnittet av arrayen
- c) Funksjon som finner det største tallet i arrayen
- d) Funksjon som finner antallet negative tall i arrayen
- e) Funksjon som finner ut om ett bestemt tall (angis som en parameter) er i arrayen eller ikke
- d) Funksjon som finner gjennomsnittet av elementene mellom en nedre og øvre index i en array:

```
function finnGjennomsnittUtvalg(array, startindex, sluttindex)
```

7.6

Skriv en funksjon som returnerer om to arrayer (parameterne) har samme sum (Tips: Benytt funksjonen som finner en sum av en array, som du har laget i forrige oppgave):

```
function sammeSum(listeA, listeB)
```

7.7

- a) Ved hjelp av en array med et utvalg bokstaver og en randomgenerator, lag en funksjon som trekker en tilfeldig bokstav.
- b) Lag en funksjon som genererer et tilfeldig ord ved hjelp av funksjonen i a). Her skal lengden på ordet være parameter.
- c) Lag en funksjon som genererer en tilfeldig setning ved hjelp av funksjonen i b). Her skal antall ord, og minste og lengste lengde på ordene være parametere.
- d) Lag et grensesnitt der brukeren kan fylle ut informasjon om antall setninger, minste og største antall ord i hver setning samt minste og største antall bokstaver i hvert ord. Når brukeren trykker på en knapp skal så en såkalt "dummy-tekst" genereres for brukeren basert på valgene.

7.8 (nøtt/stor)

Du skal lage et hangman-spill ved å benytte canvas. Forsøk å bygge opp spillet ved hjelp av funksjoner. En funksjon bør som en tommelfingerregel ikke ha mer enn 20-30 linjer kode som maks.

Følgende funksjonalitet er et minimum:

- For hver gang man trykker med musen på canvas-elementet skal neste del av hangman vises.
- Spilleren skal kunne skrive inn en bokstav og få en eller annen form på feedback på om bokstaven er med i ordet eller ikke.
- (merk at minimumsløsningen ikke krever kobling mellom tegningen i canvas og "ordgjettingen", og at det kan løses som to separate prosjekter for de som synes det blir for stort.)

Følgende funksjonalitet må med for å løse oppgaven fullt ut:

- Om bokstaven finnes i ordet skal denne/disse bokstavene bli synlige i løsningsordet der bokstaver ellers er kammouflert som _
- Om bokstaven ikke finnes skal neste del av hangman vises.
- Løsningsord skal velges blant en liste forhåndsdefinerte ord.
- Når siste del av hangman vises skal det stå "game over" i canvas-elementet. Klarer vi ordet skal vi vise en gratulasjon

Følgende funksjonalitet hadde vært ønskelig (for de som vil gjøre noe ekstra):

- Vanskelighetsgrad som er basert på antall bokstaver i ordet eller på tid(countdown)
- Poengsum som viser hvor godt spilleren løste oppgaven. Dette kan f.eks være brukt tid, antall benyttede forsøk eller vanskelighetsgrad

8. Finne og rette feil

8.1

Forklar med dine egne ord følgende begreper og programmeringselementer:

- syntaktisk feil
- semantisk feil
- logisk feil
- grensesnittfeil
- debugger
- stoppunkt (breakpoint)

8.2

Hvilke begreper vil du benytte til å beskrive følgende feil?

- a) Et semikolon mangler etter en instruksjon
- b) Et program som konverterer fra cm til tommer takler ikke negative verdier
- c) Man skriver `wile` istedenfor `while` når man ønsker å lage en løkke
- d) En nedtrekksliste inneholder dobbelt opp av elementer
- e) Dersom man ikke skriver en verdi i en tekstboks, så krasjer programmet
- f) Man har glemt hermetegn rundt en tekststreng

8.3

Finn frem tidligere eksempler som du syntes var vanskelige, og utfør koden skritt for skritt ved hjelp av en debugger for å se hvordan de fungerer.

9. Hendelser

9.1

Forklar med dine egne ord følgende begreper og programmeringselementer:

- hendelse
- sender
- lytter / lytterfunksjon
- hendelsesobjekt
- target
- currentTarget
- setInterval
- setTimeout

9.2

a) Gi en kort beskrivelse av sender-lytter-modellen.

b) Skriv kode for å registrere musklikk til en trykknapp. Ta utgangspunkt i koden og vis hva som er:

- Registrering av lytterfunksjonen
- Elementet som utløser/sender hendelsen
- Type/navn på hendelsen
- Navn på lytterfunksjonen
- Lytterfunksjonen
- Hendelsesobjektet

c) Hvorfor kan det av og til være nyttig å registrere flere lytterfunksjoner til det samme sender-objektet/elementet?

d) Hvorfor kan det av og til være nyttig å registrere den samme lytterfunksjonen til flere sender-objekter/elementer?

e) Hvordan avregistrerer (deaktiverer) vi en lytterfunksjon slik at den ikke lenger blir kjørt hvis hendelsen utløses?

9.3

Lag en overskrift. Skriv kode slik at overskriften endrer farge når du flytter musa over den. Når du flytter musa vekk fra overskriften skal den gå tilbake til normal farge

9.4

Lag en nettside som måler brukerens reaksjonstid. Nettsiden skal virke på følgende måte:

Et bilde av en sirkel og startknapp vises. Etter at brukeren trykker på knappen skal sirkelen forsvinne og det tar så ett sted mellom 2 og 5 sekunder før sirkelen igjen vises. Tiden på forsinkelsen skal være vilkårlig og variere for hver gang brukeren trykker på startknappen.

- Brukeren skal så fort som mulig klikke med musa på sirkelen etter at sirkelen er blitt vist. Tiden mellom visning av sirkelen og klikket er reaksjonstiden. Denne tiden skal vises for brukeren:



I denne oppgaven kan følgende kode være nyttig:

```
var d = new Date();  
var t = d.getTime();
```

Variabelen `t` vil bli fylt med antall millisekunder som er gått siden 1/1-1970 når koden ble utført. Ved å gjøre en slik uthenting når sirkelen vises, og en når brukeren trykker på sirkelen kan vi finne differansen mellom disse to antallene millisekunder.

10. Lyd, video og animasjon

10.1

Forklar med dine egne ord følgende begreper og programmeringselementer:

- programmert animasjon

10.2

Lag et program som spiller av en kort lyd hver gang du trykker på en tast.

10.3

Lag et program som spiller en kort lyd hvert 5 sekund.

Tips: `setInterval`.

10.4

Lag et program hvor du kan spille av forskjellige lyder ved å klikke på forskjellige knapper (såkalt soundboard). Lydene skal ikke spilles oppå hverandre.

10.5

Animer en -tagg slik at den roterer ved å endre CSS-egenskapen transform. Lag to trykknapper – en som gjør at du kan senke hastigheten, og en annen som gjør at du kan øke hastigheten på rotasjonen. Lag enda en trykknapp som skifter retning på rotasjonen.

11. Eksternt innhold

11.1

Forklar med dine egne ord følgende begreper og programmeringselementer:

- dynamisk innhold
- lokal webserver / testserver
- AJAX
- strukturerte data
- XML
- web storage
- client-side-script
- server-side-script
- JSON
- proxy-script

11.2

Lag et program som laster inn et bilde, en overskrift og en tekst fra en og samme tekstfil (bildet er da kun filnavn)

11.3

Finn en selvvalgt ekstern kilde for XML eller JSON og lag et stilig prosjekt som benytter dataene. Merk at dataene skal lastes via en proxy. Minimumsversjonen kan lages enkel, men gjør gjerne mye ut av dataprosesseringen.

11.4 (Stor/nøtt)

IT-Foreningen Enigma får stadig pågang fra bedrifter som ønsker å rekruttere nye ansatte gjennom bedriftspresentasjoner. Dette skaper behov for noen små webløsninger til å administrere presentasjonene.

Til dette formålet har Enigmas IT-ansvarlige laget fila *presentasjoner.dat* og fylt den med data:

812SuperIT2015-12-06 18:0025

614MediaHuset2016-01-14 16:1567

Osv..

a) Lag en opplisting med alle de presentasjonene som er fremover i tid.

Opplistingen skal ha følgende form:

812 - SuperIT

2015-12-06 18:00

Antall plasser: 25

614 – MediaHuset

2016-01-14 16:15

Antall plasser: 67

osv...

TIPS: Bruk Google og lær deg hvordan du kan sammenligne to datoer. Ikke håndter det som tekst.

b) Enigma ønsker også å ha en egen nettside med et påmeldingsskjema til presentasjonene. Dette påmeldingsskjemaet skal inneholde:

- nedtrekksliste for å velge presentasjon (hentes fra fila). Alternativt ha en tekstboks der man kan skrive inn koden (minimumsløsning)
- tekstboks for navn
- tekstboks for e-post

Skjemaet skal sende data til et script som heter *registrer.php* med URL på følgende form:
registrer.php?kode=812&navn=Ole Olsen&epost=ole.olsen@domene.no

Fila *registrer.php* skriver data til en fil kalt *paameldinger.dat* med følgende format:

*812xOle Olsenxole.olsen@domene.no**614xKari Karisenxkari.karisen@domene.no*

Osv...

c) Gjør endringer i oppgave a slik at utskriften blir som følger (her tenkt at det er 23 påmeldinger til presentasjon 812 og 67 påmeldinger til presentasjon 614):

812 - SuperIT

2011-12-06 18:00

Antall ledige plasser: 2 av 25 (meld deg på)

614 - MediaHuset

2012-01-14 16:15

Antall ledige plasser: 0 av 67

Linken "meld deg på" skal kun vises når det er ledige plasser på presentasjonen. Linken skal peke til påmeldingsskjemaet.

12. Objektorientering

12.1

Forklar med dine egne ord følgende begreper og programmeringselementer:

- klasse
- objekt
- arv
- abstraksjon
- innkapsling
- grensesnitt (for en klasse)
- konstruktør

12.2

Oppsummer kort hvilke fordeler du ser ved å benytte et objektorientert programmeringsspråk.