

i Informasjon om eksamen



EKSAMEN

Emnekode: ITF10213

Emnenavn: Innføring i programmering

Dato: 05/12-2017

Eksamenstid: 09.00-13.00, 4 timer

Hjelpemidler: 4 A4-sider (2 to-sidige ark eller 4 en-sidige ark) med egenproduserte notater (håndskrevne/maskinskrevne)

Faglærer: Tom Heine Nätt

Om eksamensoppgaven:

Oppgavesettet er inndelt i **3** deler med henholdsvis **10**, **2** og **2** oppgaver. Du er selv ansvarlig for å kontrollere at oppgavesettet er komplett.

Det er på hver del angitt hvor mye disse teller av totalen. Karakter fastsettes dog på basis av en helhetsvurdering av besvarelsen. Merk deg at del 1 bør gå forholdsvis raskt å svare på, så tidsmessig er det viktig å gi del 2 og 3 mer tid.

Merk deg at du også blir bedømt på hvor ryddig og god kode du skriver. At alle oppgaver fungerer betyr ikke automatisk at du får beste karakter.

Dersom noen oppgaver er så vanskelige at du ikke vet hvordan du skal løse de, gjør forenklinger og forklar nøye hva du har forenklet/tatt bort.

Les gjennom alle oppgavene før du begynner. Husk at det ikke er gitt at oppgavene står sortert etter økende vanskelighetsgrad.

Takk for et hyggelig semester, og god juleferie:-)

Lykke til!

Sensurfrist: 03/01-2018

Karakterene er tilgjengelige for studenter på Studentweb.

i Introduksjon - del 1

Her skal du kun svare på selve spørsmålet, og du trenger ikke forklare hvordan du kommer frem til svaret. NB! Les koden og selve spørsmålet nøye, for det kan være mange feller å gå i her... Det er ingen direkte syntaksfeil i koden, og alle koder vil kjøre og gi en utskrift.

Digital eksamen: Det skal ikke være behov for det, men om du gjerne ønsker er det en egen tekstboks til slutt i del 1 der du kan gi tekstlige kommentarer.

Vær nøye med å svare eksakt det innholdet i taggen *utskrift* vil bli.

1 Oppgaver - Del 1

Oppgave 1.1

Hva blir utskriften fra følgende kode?

```
var a = 3;  
var b = 7;  
var c = a + b;
```

```
document.getElementById("utskrift").innerHTML = c;
```

```
a = 4;  
b = 3;
```

Oppgave 1.2

Hva blir utskriften fra følgende kode?

```
var a = 5;
```

```
if(a < 10) {  
    document.getElementById("utskrift").innerHTML = "A";  
    a = 10;  
}  
else {  
    document.getElementById("utskrift").innerHTML = "B";  
}
```

Oppgave 1.3

Hva blir utskriften fra følgende kode?

```
var a = 5;
```

```
if(a < 10) {  
    a = 10;  
}  
if(a === 10) {  
    a = 5;  
}  
document.getElementById("utskrift").innerHTML = a;
```

Oppgave 1.4

Hva blir utskriften fra følgende kode?

```
var arr = [3,7,2,1,3];
```

```
var tall = arr[1] + arr[2];
```

```
document.getElementById("utskrift").innerHTML = tall;
```

Oppgave 1.5

Hva blir utskriften fra følgende kode?

```
var b = 5;
```

```
b++;
```

```
b += 1;
```

```
b = b - 1;
```

```
b *= 2;
```

```
document.getElementById("utskrift").innerHTML = b;
```

Oppgave 1.6

Hva blir utskriften fra følgende kode?

```
var res = 0;
```

```
for(var i = 0; i < 5; i++) {  
  res += i;  
}
```

```
document.getElementById("utskrift").innerHTML = res;
```

Oppgave 1.7

Hva blir utskriften fra følgende kode?

```
for(var i = 0; i < 5; i++) {  
  document.getElementById("utskrift").innerHTML += i + "!";  
}
```

Oppgave 1.8

Hva blir utskriften fra følgende kode?

```
var res = 0;
for(var i = 0; i < 9; i++) {
  if(res < 8) {
    res += i;
  }
}
```

```
document.getElementById("utskrift").innerHTML = res;
```

Oppgave 1.9

Hva blir utskriften fra følgende kode?

I dette tilfellet vill siste tre kodelinjer stå i "oppstarts-funksjonen", mens selve funksjonen er definert for seg selv.

```
function tullball(a,b) {
  if( b > a ) {
    return b-a;
  }
  else {
    return a-b;
  }
}
```

```
var a = tullball(3,4);
var b = tullball(2,6);
document.getElementById("utskrift").innerHTML = tullball(a,b);
```

Oppgave 1.10

Hva blir utskriften fra følgende kode?

```
var a = "8!2,4!2,9!3,12!4";

var b = a.split(",");
var c = 0;

for(var i = 0; i < b.length; i++) {
  var d = b[i].split("!");
  c += d[0] / d[1];
}
```

```
document.getElementById("utskrift").innerHTML = c;
```

2 **Kommentarer - del 1**

Dersom du har kommentarer til oppgavene i del 1 kan du fylle inn disse her.
Merk deg at dette ikke skal være nødvendig.

Har du ingen kommentarer, så sett et punktum i tekstboksen for å unngå advarsler om at denne er ubesvart.

Skriv ditt svar her...

Format	▼				↺		
	✎		Σ		✕		
							Words: 0

i Introduksjon - del 2

I oppgavene til denne delen trenger du **ikke** skrive et komplett HTML-dokument. Skriv kun den JavaScript-koden som etterspørres.

3 Oppgave 2.1

Du har tre arrayer (de inneholder mange hundre ord hver, selv om det her vises kun fire ord):

```
var subjekt = ["Ole", "katten", "ballen", "Kari"];
```

```
var verbal = ["liker", "savnet", "mistet", "malte"];
```

```
var adverbial = ["forsiktig", "raskt", "bestemt", "undrende"];
```

Du skal generere opp 50 setninger bygget på en bestemt form ved å velge ord tilfeldig fra de tre arrayene. Et siste ord kalt *direkte objekt* skal også plukkes fra arrayen *subjekt*. Det teller positivt om du sørger for at samme ord ikke velges som *direkte objekt* og *subjekt*.

Setningene skal se slik ut:

```
"subjekt verbal direkteObjekt adverbial, sa brura."
```

Altså:

```
Ole savnet Kari forsiktig, sa brura.
```

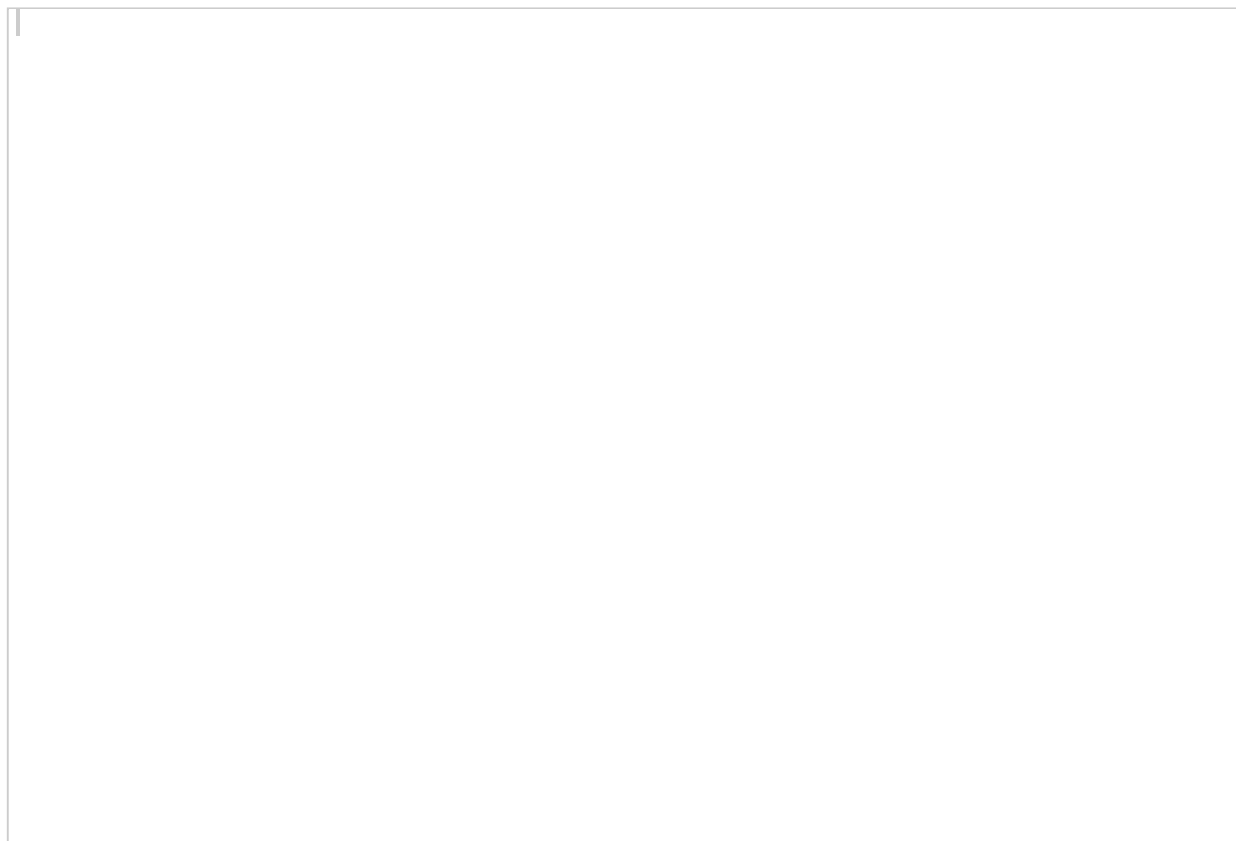
```
Katten mistet ballen bestemt, sa brura.
```

Du kan gå ut i fra at følgende funksjon som returnerer teksten med stor forbokstav finnes ferdig definert:

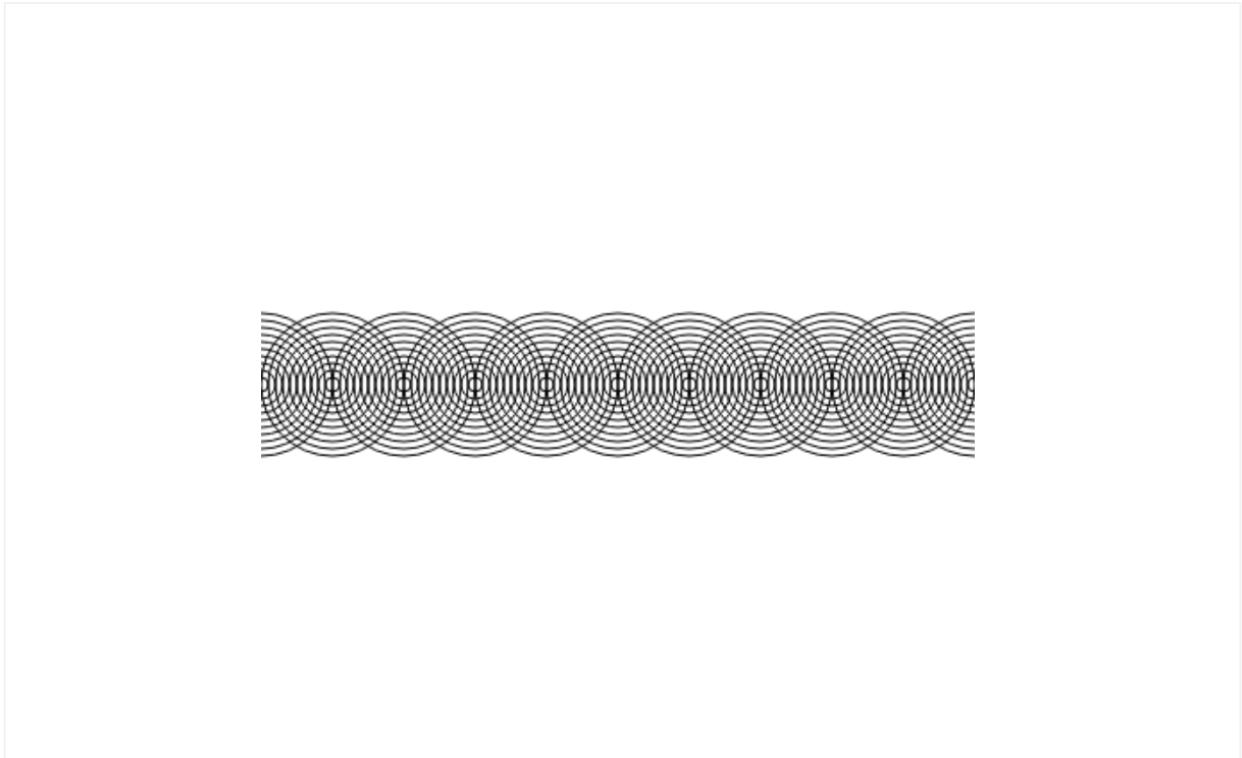
```
function storForbokstav(string)
```

Tips: For å unngå å plukke samme ord, kan du benytte en while-løkke som trekker et nytt ord så lenge det ikke er det samme som subjektet.

Skriv ditt svar her...

A large, empty rectangular box with a thin black border, intended for the user to write their answer. The box is positioned below the instruction text and occupies most of the page's width and a significant portion of its height.

4 Oppgave 2.2



I denne oppgaven kan du gå ut i fra at følgende `<canvas>`-tagg står i HTML-dokumentet:

```
<canvas id="tegneflate" width="400" height="400"></canvas>
```

Ved hjelp av løkker, produser følgende grafikk: (se illustrasjonen til oppgaven)

NB! Pga. måten Inspera viser illustrasjoner er det mye hvitt rundt grafikken. Grafikken dekker imidlertid **hele** canvaset i horisontal retning. De halve sirklene på hver side skyldes altså at kun deler av sirkelen tegnes innenfor canvaset.

Hint:

Tegn først én forekomst av "sirkelsamlingen". Dette er 10 sirkler med radius fra 4-40 punkter.

Få deretter denne "sirkelsamlingen" til å gjenta seg med sentrum fra $x=0$ og $y=200$ til $x=400$ og $y=200$ med 40 punkter økning i x -retning hver gang.

i Introduksjon - del 3

NB! I denne delen skal du lage komplette nettsider (med HTML-kode).

Digital eksamen: Ønsker du å dele opp oppgaven i flere eller andre filer enn det svar-boksene i den digitale eksamen legger opp til, så skriv filene i samme svar-boks, og marker med kommentarer når hver fil starter/slutter.

En foreleser sitter under eksamenssensurering og ønsker seg et enkelt system for å registrere "gullkorn" fra studentene som dukker opp i besvarelsene. Du skal nå lage dette systemet.

Alle gullkorn er registrert på følgende format i fila *gullkorn.dat*:

```
emnekode;semester;gullkorn
```

Et eksempel på en fil med fire gullkorn kan være (linjebryddene som her evt. vises pga. sidebredden vil ikke være til stede i fila):

```
ITF10213;H2009;Hadde jeg hatt tid, ville jeg løst resten av oppgavesettet helt riktig
```

```
ITF10511;H2014;Tilgjengelighet: ... en god opplevelse for alle uansett hemninger
```

```
ITF15015;V2015;Slemme datamenn
```

```
ITF10213;H2010;Etter å ha løst omtrent 10% av oppgavesettet står det: Jeg hadde dessverre ikke tid til å gjøre resten
```

Det finnes også en JSON-fil med alle emnekoder og tilhørende emnenavn. Denne fila har filnavnet *emner.json*. Denne kan se slik ut:

```
[  
  {"emnekode": "ITF10213",  
   "emnenavn": "Innføring i programmering"},  
  {"emnekode": "ITF10511",  
   "emnenavn": "Webutvikling"},  
  {"emnekode": "ITF15015",  
   "emnenavn": "Innføring i datasikkerhet"}  
]
```

Selv om oppgaven her viser fire gullkorn og tre emner, skal du lage et system som håndterer et ukjent antall gullkorn og et ukjent antall emner.

Skulle JSON-fila by på problemer, finnes det to mulige alternativer for å forenkle oppgavene. Å velge et av disse alternativene vil imidlertid ikke gi full uttelling på oppgaven.

Alternativ 1: I stedet for en JSON-fil kan du gå ut i fra at det er en semikolonseparert fil med navnet *emner.dat*. Fila vil da ha følgende format:

emnekode;fagnavn

og kan typisk se slik ut:

ITF10213;Innføring i programmering

ITF10511;Webutvikling

ITF15015;Innføring i datasikkerhet

Alternativ 2: En minimumsløsning er å håndtere emneinformasjonen som en array i selve programkoden. Gå altså ut i fra at du har følgende array tilgjengelig:

```
var emneinfo = [  
  {"emnekode": "ITF10213",  
   "emnenavn": "Innføring i programmering"  
  },  
  {"emnekode": "ITF10511",  
   "emnenavn": "Webutvikling"  
  },  
  {"emnekode": "ITF15015",  
   "emnenavn": "Innføring i datasikkerhet"  
  }  
];
```

5 Oppgave 3.1

Gullkorn

Innføring i programmering (ITF10213)

Hadde jeg hatt tid, ville jeg løst resten av oppgavesettet helt riktig (H2009)

Etter å ha løst omtrent 10% av oppgavesettet står det: Jeg hadde dessverre ikke tid til å gjøre resten (H2010)

Webutvikling (ITF10511)

Tilgjengelighet: ... en god opplevelse for alle uansett hemninger (H2014)

Innføring i datasikkerhet (ITF15015)

Slemme datamenn (V2015)

Lag en nettside som henter ut informasjonen fra fila *gullkorn.dat* og fila *emner.json* (evt. alternativ 1 eller 2) og presenterer informasjonen på følgende måte (se oppgaveillustrasjon)

Her vises nettsiden slik den ville blitt med de fire gullkornene og de tre emnene. Flere gullkorn/emner vil selvfølgelig gi en lengre opplisting.

Gullkornene er altså listet opp under de ulike emnene de tilhører. Under hvert emne kan gullkornene listes opp i den rekkefølgen du ønsker, men enklest er det nok å liste de opp i den rekkefølgen de står i fila.

Det er mange måter å løse dette på, men én mulighet er å først lese inn fila *emner.json*. Når denne er lest inn, så leser du inn fila *gullkorn.dat*. For hver oppføring i fila *emner.json* skriver du ut emnenavnet og går deretter gjennom informasjonen du hentet fra *gullkorn.dat*. De gullkornene som har samme emnekode som emnet listes ut. Når du er gjennom alle gullkornene gjør du det samme med neste emne.

6 Oppgave 3.2

Emne: ▾
År:
Semester: ▾
Gullkorn:

Foreleseren ønsker også at du lager et skjema for å registrere nye gullkorn i fila *gullkorn.dat*. Skjemaet skal vise emnenavnene i en nedtrekksliste. Denne må altså fylles med informasjonen fra fila *emner.json* (eventuelt alternativ 1 eller 2).

Før knappen *Registrer* sender data videre til et PHP-script, som skriver informasjonen til slutten av fila *gullkorn.dat*, skal du sjekke følgende:

- Sitatet kan ikke være blankt
- Årstallet må være høyere enn 2005 og mindre enn dagens årstall. (Tips: Hentes ut via `new Date` og `getFullYear()`)

Skulle noe av dette ikke stemme skal det vises en meldingsboks (alert) på nettsiden med passende feilmelding(er).

Når registreringen er fullført skal det vises "Registrering OK" i en meldingsboks (alert).

Husk at du også skal lage selve PHP-fila som skal benyttes til skrivning. Denne skal ha filnavnet *registrer.php*.

(Eksamenssystemet støtter kun én "kodeboks" til hver oppgave. Skriv PHP-fila i samme kodeboks, skilt med en kommentar. Se bort i fra at fargekodingen da ikke helt stemmer for PHP-koden)