

## i Informasjon om eksamen



## EKSAMEN

**Emnekode:** ITF10213

**Emnenavn:** Innføring i programmering

**Dato:** 28/11-2018

**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**, **3** 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:** 19/12-2018

Karakterene er tilgjengelige for studenter i Studentweb.

## i Introduksjon - del 1 (15 %)

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.

Koden som presenteres vises uten *window.onload* og tilhørende funksjon. Du kan likevel anta at koden ville vært plassert inne i en "oppstarts-fuksjon". Funksjoner i oppgavene vil bli plassert som globale funksjoner.

## 1 Oppgaver - Del 1

### Oppgave 1.1

Hva blir utskriften fra følgende kode?

```
var a = 4;
var b = 5;
var c = a + b;
b = 1;
```

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

### Oppgave 1.2

Hva blir utskriften fra følgende kode?

```
var a = 4;
var b = 5;
```

```
if( b >= a ) {
    document.getElementById("utskrift").innerHTML = b;
}
else {
    document.getElementById("utskrift").innerHTML = a;
}
```

### Oppgave 1.3

Hva blir utskriften fra følgende kode?

```
var a = 4;
var b = 5;
```

```
if( b >= a ) {
    document.getElementById("utskrift").innerHTML = b;
}
if ( a >= b ) {
    document.getElementById("utskrift").innerHTML = a;
}
else {
    document.getElementById("utskrift").innerHTML = b;
}
```

### Oppgave 1.4

Hva blir utskriften fra følgende kode?

```
var arr = [1,2,3,4,5];
```

```
document.getElementById("utskrift").innerHTML = arr[1];
```

**Oppgave 1.5**Hva blir utskriften fra følgende kode? 

```
var arr = [1,2,3,4,5];
var b = arr.length;
document.getElementById("utskrift").innerHTML = arr[b-1];
```

**Oppgave 1.6**Hva blir utskriften fra følgende kode? 

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

**Oppgave 1.7**Hva blir utskriften fra følgende kode? 

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

**Oppgave 1.8**Hva blir utskriften fra følgende kode? 

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

  document.getElementById("utskrift").innerHTML += (i % 2) + "#";
}
```

**Oppgave 1.9**Hva blir utskriften fra følgende kode? 

```
function test(a,b,c) {
  if(c === true) {
    return a+b;
  }
  else {
    return a-b;
  }
}
```

```
var riktig = 3 > 4 && 7 > 2;
document.getElementById("utskrift").innerHTML = test(3,4,riktig);
```

**Oppgave 1.10**Hva blir utskriften fra følgende kode? 

```
var a = "0#4!2#3!3#4";
var sum = 0;
var deler = a.split("!");
for(var i=0; i < deler.length;i++) {
  var tall = deler[i].split("#");
```

```
sum += tall[0]*tall[1];  
}  
document.getElementById("utskrift").innerHTML = sum;
```

Maks poeng: 10

## 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. Det skal svært mye til før en kommentar her vil påvirke bedømmingen av et svar gitt på selve oppgaven.

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

**Skriv ditt svar her...**

Format - | **B** *I* U  $x_2$   $x^2$  |  $T_x$  | | | |  $\Omega$  | |  $\Sigma$  | ABC |

Words: 0

Maks poeng: 0

## i Introduksjon - del 2 (35 %)

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

## 3 Oppgave 2.1 (15 %)

Du har en array med tall. Denne kan f.eks. være definert som følger:

```
var tall = [2,4,2,3,1,2,5,6,5];
```

Skriv ut alle tall i arrayen som kun forekommer en gang. Skriv ut ett tall på hver linje.

For eksempelarrayen over, ville denne utskriften blitt (Det er ingen krav til rekkefølge i utskriften):

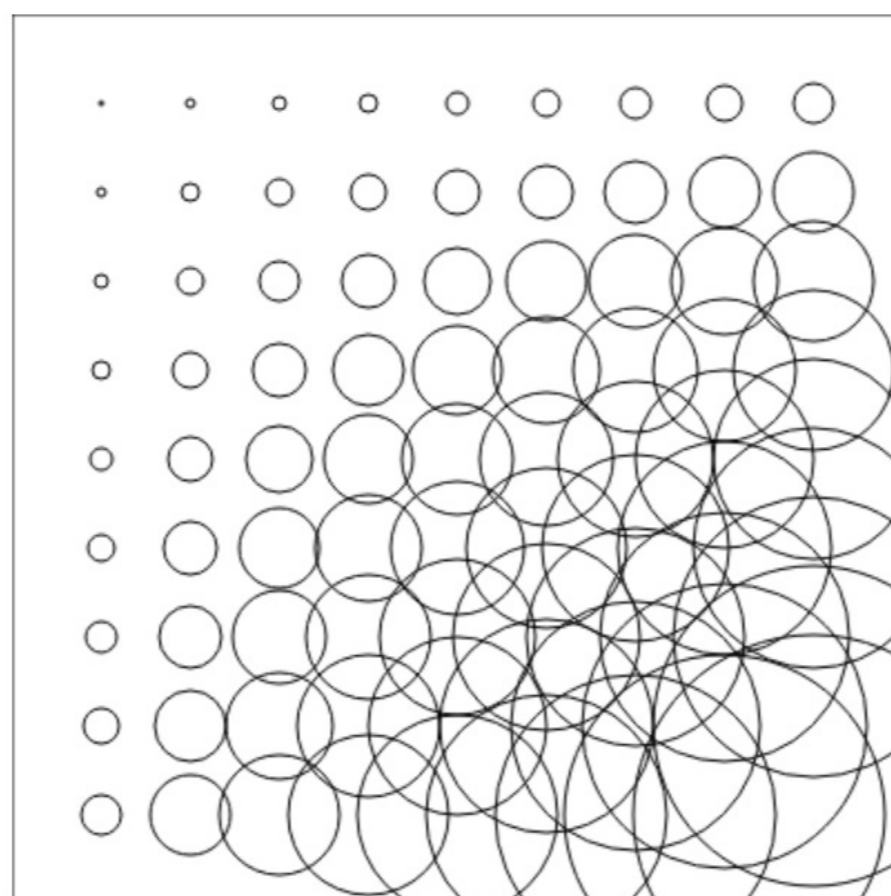
```
4  
3  
1  
6
```

**Tips:** De fleste måter å løse oppgaven på vil kreve to løkker (nestede løkker) - Én som styrer hvilket tall vi jobber med nå, og én som finner ut om det finnes flere forekomster av dette tallet i arrayen.

**Skriv ditt svar her...**

Maks poeng: 15

#### 4 Oppgave 2.2 (15 %)



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. Den sorte firkantede rammen (som du også skal tegne) går fra 0,0 til 400,400. De

halve sirklene skyldes da altså at kun deler av sirkelen tegnes innenfor canvaset.

**Hint:**

Det skal altså tegnes sirkler i et "rutenett" på 9 x 9 sirkler. De ytterste sirklernes senter skal være 40 punkter fra kantene. Radiusen til sirklene angis ved å ta posisjonsnummeret til sirkelen fra høyre og fra toppen, og gange sammen.

Første sirkel øverst til venstre har derfor radius 1 (1\*1). Sirkelen øverst til høyre har radius 9 (1\*9). Sirkelen nederst til høyre har radius 81 (9\*9).

Husk også å tegne den sorte firkantede rammen.

**Skriv ditt svar her...**

Maks poeng: 15

**5 Oppgave 2.3 (5 %)**

Du skal i denne oppgaven lage en funksjon som arbeider med tidspunkt. Denne funksjonen skal også benyttes i oppgave 3.1, men det er ikke vesentlig nå. Det er heller ikke nødvendig at funksjonen blir ferdig for å klare og løse 3.1

Sist i dette oppgavesettet er det et vedlegg med litt praktisk informasjon om tidspunkt og tidsangivelser i JavaScript.

Du skal nå lage en funksjon som tar i mot to tekststrenger som representerer et start- og slutt-tidspunkt på ISO-format, sammen med et Dato-objekt som representerer et annet tidspunkt:

**function aktiv(start,slutt,tidspunkt)**

Funksjonen skal returnere **true** dersom tidspunktet ligger mellom start og slutt, **false** ellers.

Typisk bruk kan være:

```
var forelesningsstart = "2018-11-05T14:15:00+01:00";  
var forelesningsslutt = "2018-11-05T16:00:00+01:00";
```

```
var naa = new Date();
```

```
if (aktiv(forelesningsstart,forelesningslutt,naa) === true) {  
    alert("Kom deg på forelesning!");  
}
```

**Skriv ditt svar her...**

--	--

Maks poeng: 5

## **i Introduksjon - del 3 (50 %)**

**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.

Merk deg at denne introduksjonen gjelder både oppgave 3.1 og oppgave 3.2. Enkelte opplysninger vil kun være aktuelle for en av oppgavene. Du bør likevel lese hele denne introduksjonen før du går i gang. Minner også om vedlegget til slutt i oppgavesettet som beskriver datoangivelse i JavaScript.

En foreleser ønsker å lage en enkel nettside som studenter kan benytte for å få oversikt over avtaler som gjør at foreleser ikke er på kontoret. Denne nettsiden skal potensielt vises på en infoskjerm på kontordøra.

Systemet bygger på en enkel tekstfil med navn *avtaler.dat* i følgende format:

*starttidspunkt;sluttidspunkt;tittel;sted;icon*

Et eksempel på fil (som også blir benyttet i videre skjermbilder) er følgende:

```
2018-09-24T08:00:00+01:00;2018-10-08T16:00:00+01:00;Fødselspermisjon;Hjemme;DAYOFF  
2018-11-21T14:15:00+01:00;2018-11-21T16:00:00+01:00;Forelesning;Aud Max;LECTURE  
2018-12-03T08:00:00+01:00;2018-12-02T10:00:00+01:00;Personalmøte;D1-057;MEETING  
2018-12-03T12:00:00+01:00;2018-12-02T13:00:00+01:00;Foredrag for VGS;D1-052;LECTURE  
2018-12-03T14:00:00+01:00;2018-12-02T20:00:00+01:00;Bursdagfeiring;Hjemme;BIRTHDAY  
2018-12-05T08:00:00+01:00;2018-12-03T16:00:00+01:00;Uvell;Hjemme;SICK  
2018-12-01T11:00:00+01:00;2018-12-01T13:00:00+01:00;Møte i eksamensgruppa;D1-038;MEETING
```

Merk deg at linjebrydd (mer enn for hver oppføring) kan forekomme pga visningsbredden i eksamensverktøyet. Disse linjebryddene vil ikke være tilstede i fila.

Siste felt i hver oppføring henviser på et icon. I samme mappe som filene du skal lage i 3.1 og 3.2 vil det ligge PNG-filer med samme navn. Disse vil imidlertid være skrevet med små bokstaver, slik at vi finner filene *dayoff.png*, *lecture.png* osv.

Det finnes også en JSON-fil (som skal benyttes i oppgave 3.2) med alle icon og tilhørende ledetekst. Denne fila har navnet *status.json*, og kan se slik ut:

```
[
  {
    "status": "Fridag",
    "kode": "DAYOFF"
  },
  {
    "status": "Bursdag",
    "kode": "BIRTHDAY"
  },
  {
    "status": "Hjemmekontor",
    "kode": "HOMEOFFICE"
  },
  {
    "status": "Forelesning",
    "kode": "LECTURE"
  },
  {
    "status": "Møte",
    "kode": "MEETING"
  },
  {
    "status": "Syk",
    "kode": "SICK"
  }
]
```

Skulle JSON-fila by på problemer, kan du i stedet benytte en tekstfil med navnet *status.dat*. Dette vil imidlertid ikke gi full uttelling:

*beskrivelse;kode*

Fylt med data vil den typisk se slik ut:

*Møte;MEETING*  
*Syk;SICK*  
*Forelesning;LECTURE*  
osv.

## 6 Oppgave 3.1 (30 %)



## Kalender for Tom Heine Nätt

### Nå



Foredrag for VGS (D1-052) - Fra 12:00 til 13:00

### I dag



Personalmøte (D1-057) - Fra 08:00 til 10:00



Bursdagfeiring (Hjemme) - Fra 14:00 til 20:00

Lag en nettside som henter ut informasjon fra fila *avtaler.dat* og presenterer informasjonen på følgende måte (se oppgaveillustrasjon).

Her vises nettsiden slik den ville blitt vist 03/12-2018 kl 12.30 med avtalefila slik den ble presentert i introduksjonen til denne delen av oppgavesettet.

Her hentes altså alle avtaler inn. De som er aktive på nåværende tidspunkt presenteres under "Nå", de som gjelder dagen i dag presenteres under "I dag". Merk deg at avtaler som gjelder nå (kan potensielt være flere) ikke skal vises under "I dag".

Det er noen krav til oppgaven:

- \* Navnet på foreleser skal oppbevares i en global variabel
- \* Filnavnet på avtalefila skal oppbevares i en global variabel

Som hjelp til oppgaven kan du gå ut i fra at du har følgende funksjoner ferdige og tilgjengelige:

#### **function aktiv(start,slutt,tidspunkt)**

Tar i mot to ISO-formaterte datostrenger og et dato-objekt og avgjør om avtalene disse gjelder er aktiv på angitt tidspunkt. Returnerer true/false. Denne funksjonen ble laget i oppgave 2.3, men du kan gå ut i fra at den er ferdig uavhengig av om du fikk den til eller ikke der.

#### **function innenforDato(start,slutt,dato)**

Tar i mot to ISO-formaterte datostrenger, samt et dato-objekt. Avgjør om avtalen disse tidspunktene gjelder vil være aktiv i løpet av datoen som dato-objektet angir. Returnerer true/false

#### **function formaterKlokkeslett(datoString)**

Denne funksjonen tar i mot en ISO-formatert datostreng og returnerer selve klokkeslettet på formen HH:MM. Siffer under 10 vil bli skrevet som 0X, altså 05:12, 10:04 osv.

#### **function sorterAvtaler(avtaleliste)**

Denne funksjonen tar i mot en array med avtaler (altså en array der hver linje på fila *avtaler.dat*, som vist i introduksjonen, er ett element i arrayen) og returnerer en array med avtalene sortert i kronologisk rekkefølge (sortert på tid) basert på starttid.

#### **Tips:**

Det vil være besparende i tid og arbeid om du lager en funksjon om tar i mot opplysninger om en avtale og gjør disse om til en utskrift. Denne funksjonen kan da gjenbrukes både for "Nå" og "I dag".

#### **Tips:**

Det kan være nyttig å lage ferdig de to taggene der avtalene for "Nå" og "I dag" skal ligge. Når du går gjennom avtalene fra fila kan du da "hekte de på" riktig tagg og slipper med kun en gjennomgang av fildata.

**Skriv ditt svar her...**

--	--

Maks poeng: 30

**7 Oppgave 3.2 (20 %)**

Titel:	<input type="text" value="Juleavslutning"/>
Rom:	<input type="text" value="D1-057"/>
Status:	<input type="text" value="Møte"/>
Fra:	<input type="text" value="19.12.2018 14.00"/>
Til:	<input type="text" value="19.12.2018 21.00"/>
	<input type="button" value="Registrer"/>

Foreleseren ønsker også at du lager et skjema for å registrere nye avtaler i fila *avtaler.dat*. Skjemaset skal vise de ulike statusene fra fila *satus.json* i en nedtrekksliste. Denne må altså fylles med informasjon fra fila i det siden lastes (eventuelt den alternative fila *status.dat*)

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

- \* Tittel må være 5 tegn eller mer
- \* Rom må være 5 tegn eller mer
- \* til-dato kan ikke komme før fra-dato.

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). Deretter skal skjemaet tømmes slik at det er klart til en ny avtale (dato i til- og fra-boksene kan få stå igjen).

Til registrering av dato/tid benyttes HTML5-elementet **<input>** med **type** satt til *datetime-local*:

```
<input type="datetime-local"/>
```

Når man henter ut egenskapen **value** fra denne inputboksen, får man en ISO-formatert datostreng som kan benyttes rett i JavaScript, og som også er samme format som datoene i fila *avtaler.dat* har.

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

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

**Skriv ditt svar her...**

--	--

Maks poeng: 20

## **i Vedlegg: Tidspunkt i JavaScript**

Tidspunkt i JavaScript krever litt forklaring. Et tidspunkt angis i form av en tekstlig representasjon som er bygget etter et helt fast mønster. Dette kalles ISO-format:

```
var tidspunkt = "2018-12-02T17:15:00+01:00"
```

Mønsteret er som følger:

YYYY-MM-DDTHH:II:SS+XX:XX

YYYY: År med 4 siffer:

MM: Måned

DD: Dag

T: Skilletegn for dato og selve tiden

HH: Timer

II: Minutter

SS: Sekunder

XX:XX tidssone (der f.eks. "+01:00" betyr 1 time før GMT, altså vår "norske tidssone"). I hele oppgavesettet vil "+01:00" benyttes.

En dato kan også forekomme i form av et JavaScript-objekt. Dette gjør det lettere å arbeide med datoen. Følgende kode lager et objekt som representerer tidspunktet akkurat nå:

```
var naa = new Date();
```

Følgende kode lager et objekt som representerer et bestemt tidspunkt. Legg merke til at dette settes ved hjelp av tidspunkt-strengen/formatet:

```
var bursdag = new Date("2018-12-02T17:15:00+01:00");
```

Ut i fra et dato-objekt kan man plukke en god del informasjon gjennom ulike metoder. De viktige for dette oppgavesettet vises under:

Hente ut "time-delen" av et tidspunkt:

```
var time = bursdag.getHours(); // gir 17 i dette eksempelet
```

Hente ut "minutt-delen" av et tidspunkt:

```
var time = bursdag.getMinutes(); // gir 15 i dette eksempelet
```

Hente ut antall millisekunder mellom 1. januar 1970 og tidspunktet:

```
var millisekunder = bursdag.getTime();
```

En spesielt smart funksjon er at man lett kan sammenligne to tidspunkt ved å sjekke tidspunktens "getTime()" opp mot hverandre.:

```
if (bursdag.getTime() > naa.getTime() ) {  
    alert("Ennå ikke hatt bursdag i 2018");  
}
```