

Innledning

Emne: PGR102 Introduksjon til Programmering

Eksamensdato: 12.12.2018

Tillatte hjelpemidler: Ingen

Velkommen til eksamen i PGR102.

Les gjennom hele oppgavesettet. Hvis du synes noe er uklart eller mener at opplysninger mangler, må du gjøre egne begrunnede antagelser/forutsetninger, og løse oppgaven ut fra disse. Husk å formatere kode med innrykk og linjeskift så den blir enklere å lese og forstå.

Teoretisk: 35%

I denne seksjonen av eksamen skal du vise til teoretisk forståelse for programmering i konteksten av JavaScript, samt vise til kunnskap om informatikkens historie.

Oppgave 1.1

Teller: 3% (gir maksimalt 3 poeng)

Vi starter med litt informatikkhistorie. I denne oppgaven skal du koble riktig person(er) til deres respektive oppfinnelse(r).

Ole Johan-Dahl og Kristen Nygaard		
Grace Hopper		
Sir Tim Berners-lee		
Joseph Marie Jacquard		
Dennis Ritchie		

⚙️ Programmeringsspråket C

⚙️ WWW og HTML

⚙️ Programmeringsspråket COBOL

⚙️ Programmeringsspråket Simula

⚙️ Hullkort

Oppgave 1.2

Teller: 11% (gir maksimalt 11 poeng)

Før JavaScript-kode kjøres i nettleseren blir koden omgjort til et annet format. Det ser ikke lenger ut som JavaScript, men det er en *representasjon* av koden som nettleseren enklere kan *tolke*. På bildet under ser du et utdrag av koden slik den er omgjort til sin alternative representasjon. Hva heter denne representasjonen, og hvorfor må JavaScript-koden omgjøres til dette?

```
- Program {
  type: "Program"
  start: 0
  end: 27
+ loc: {start, end}
  sourceType: "module"
- body: [
  - ExpressionStatement {
    type: "ExpressionStatement"
    start: 0
    end: 27
    + loc: {start, end}
    - expression: CallExpression {
      type: "CallExpression"
      start: 0
      end: 26
      + loc: {start, end}
      + callee: MemberExpression {type, start, end, loc, object, ... +3}
      - arguments: [
        - Literal = Node {
          type: "Literal"
          start: 12
          end: 25
          + loc: {start, end}
          value: "Hello World"
          rawValue: "Hello World"
          raw: "\"Hello World\""
          + range: [2 elements]
        }
      ]
      + range: [2 elements]
    }
    + range: [2 elements]
  ]
+ range: [2 elements]
}
```

B *I* U | ☰ ☷ á

Oppgave 1.3

Teller: 11% (gir maksimalt 11 poeng)

Ta utgangspunkt i kodesnutten under.

```
var result = 90 + (true * 2);
```

Besvar følgende tre spørsmål:

1. Hva vil variabelen *result* inneholde hvis denne koden ble kjørt?
2. Hvis vi hadde sjekket datatypen til *result* med *typeof()* etter koden ble kjørt, hvilken datatype ville *typeof()* fortalt oss at *result* er?
3. Til slutt, forklar hva som foregår -- hvorfor blir resultatet slik det blir?

B *I* U | :≡ ≡≡ á

0 / 200 Word Limit

Oppgave 1.4

Teller: 7% (gir maksimalt 7 poeng)

Vis med eget eksempel hva leksikografisk rekkefølge (*lexicographical order*) er, og forklar kort hvordan det fungerer.

B *I* U | :≡ ≡≡ á

0 / 200 Word Limit

Oppgave 1.5

Teller: 3% (gir maksimalt 3 poeng)

Kodesnuttene under bruker enten *eksplisitt* (casting) eller *implisitt* (coercion) konvertering av datatyper. Fyll ut med riktig svar.

		Eksplisitt (explicit casting)	Implisitt (implicit coercion)
A	<code>var result = +"1";</code>	<input type="radio"/>	<input type="radio"/>
B	<code>var result = parseInt("100");</code>	<input type="radio"/>	<input type="radio"/>
C	<code>var result = 1 * "100";</code>	<input type="radio"/>	<input type="radio"/>
D	<code>var result = (100).toString();</code>	<input type="radio"/>	<input type="radio"/>
E	<code>var result = "" + ["Oslo", "Bergen", "Trondheim"];</code>	<input type="radio"/>	<input type="radio"/>

Anvendt teknisk: 65%

I denne seksjonen av eksamen skal du vise til teknisk forståelse og anvendelse av JavaScript for problemløsning. Det er kun to deloppgaver, men for hver av dem følger det en rekke forslag til utvidelse av funksjonalitet. For å oppnå en høy totalpoengsum, er det viktig å implementere slike utvidelser. Hvis du velger å utvide kompleksiteten i besvarelsen med annen funksjonalitet enn forslagene som listes opp, bør denne funksjonaliteten være av relativt lik kompleksitet som forslagene, i tillegg til at du må forklare hva du har utvidet med.

Oppgave 2.1

Teller: 30% (gir maksimalt 30 poeng)

Oppgaven går ut på å lage et program som gir brukeren muligheten til å skrive inn tallet på en måned, f.eks. **4**, og få returnert navnet på måneden, altså "April" i dette tilfellet. Du står fritt til å velge om du vil skrive ut navnet på måneden i console eller alert. Ta utgangspunkt i tabellen under.

Assosiert tall:	1	2	3	4	5	6	7	8	9	10	11	12
Måned:	Januar	Februar	Mars	April	Mai	Juni	Juli	August	September	Oktober	November	Desember

Eksempler på utvidelser som kan øke poengsum - **husk** å nevne i besvarelsen hvilke utvidelser du har implementert:

- a) I tillegg til å tillate at bruker kan skrive inn tallet på måneden og få returnert månedens respektive navn, tillat også brukeren å kunne skrive inn navnet på måneden for å få returnert månedens respektive assosierte tall. Her belønnes løsninger med minst mulig repeterende kode.
- b) Ta i bruk DOM-manipulasjon med `<input>`-felt hvor bruker skriver inn tallet på måneden, en knapp (inkl. event-lytter), og et HTML-element av typen paragraf (`<p> </p>`), sistnevnte for å vise frem resultatet (navnet på måneden) når bruker trykker på knappen.
- c) *NB: dette er en videreutvikling av utvidelse C over.* Stilsett paragraf-elementet (`<p> </p>`) sin tekstfarge basert på årstiden til måneden som skrives til elementet. Hvis bruker i `<input>`-feltet skriver inn assosiert tall for månedene september, oktober, november, desember, januar eller februar, for så å trykke på knappen, skal paragraf-elementet sin tekstfarge settes til fargekoden #0000FF (blå). For de resterende månedene, mars, april, mai, juni juli eller august, skal paragraf-elementet sin tekstfarge settes til fargekode #00ff00 (grønn).

B

I

U

☰

☷

á

0 / 5000 Word Limit

Oppgave 2.2

Teller: 35% (gir maksimalt 35 poeng)

Ta utgangspunkt i følgende objekt-baserte array:

```
var cities = [  
  { name: "Trondheim", population: 182000 },  
  { name: "Bergen", population: 272000 },  
  { name: "Stavanger", population: 130000 },  
  { name: "Tromsø", population: 71000 }  
];
```

Oppgaven din er å skrive en funksjon som kan erstatte et spesifikt objekt i dette arrayet, med et annet spesifikt objekt. Funksjonskallet som vist under har to argumenter -- dette funksjonskallet fungerer som et eksempel for hvordan du skal implementere funksjonen. Det første argumentet er navnet på byen vi ønsker å erstatte (Bergen). Det andre argumentet er objektet vi ønsker å erstatte Bergen med, i dette tilfellet objektet til Oslo med medfølgende innbyggertall. Her brukes Bergen og Oslo bare som eksempler, hvilke som helst av byene skal kunne byttes ut med hvilke som helst andre objekter.

```
replaceCityInArray("Bergen", { name: "Oslo", population: 630000 });
```

Etter at funksjonskallet over er kjørt, skal innholdet i *cities*-arrayet se slik ut -- legg merke til at Oslo-objektet nå er på plasseringen hvor "Bergen"-objektet var tidligere:

```
[  
  { name: "Trondheim", population: 182000 },  
  { name: "Oslo", population: 630000 },  
  { name: "Stavanger", population: 130000 },  
  { name: "Tromsø", population: 71000 }  
];
```

Eksempler på utvidelser som kan øke poengsum - **husk** å nevne i besvarelsen hvilke utvidelser du har implementert:

- Sørg for at implementasjonen din håndterer at samme by (objekt) eksisterer flere ganger i *cities*-arrayet. Med utgangspunkt i funksjonskallet over, hvis objektet med property *name*: "Bergen" eksisterer to ganger i *cities*-arrayet, skal det altså byttes ut med "Oslo"-objektet på begge plasseringene i *cities*-arrayet.
- Gjør koden din mer modulær og gjenbrukbar. Implementer funksjonen *replaceCityInArray* på en slik måte at et vilkårlig array med samme struktur som *cities*-arrayet kan modifiseres ved bruk av funksjonen, istedenfor kun *cities*-arrayet.
- Håndter/valider at byen som skal byttes ut (f.eks. "Bergen") eksisterer i *cities*-arrayet (eller det vilkårlige arrayet, hvis du også implementerer utvidelse b)). Eksisterer det ikke, gi tilbakemelding til bruker om at byen ikke eksisterer.
- Sørg for at funksjonen *replaceCityInArray* kun utfører operasjonen den skal gjøre hvis det første argumentet som sendes inn er av datatypen *string*, og det andre argumentet er av datatypen *object*.

Implementer nå funksjonen *replaceCityInArray* slik at den besvarer oppgaveteksten, og implementer samtidig de utvidelsene du kan.

B *I* U | ::≡ ≡≡ á

