

Tentamen i JavaScript för webbutveckling 5 hp (ISGB15-provkod 1500) torsdagen den 14 april 2022 08.15-13.15.

---

|                   |  |    |  |
|-------------------|--|----|--|
| Ansvariga lärare: | Peter Bellström, Pierre Sjöberg & Alexander Vestin |    |  |
| Betygsgränser:    | Max:   | 30 |  |
|                   | Godkänd:   | 15 |  |
|                   | Väl Godkänd:                                       | 23 |  |

Maxpoäng på respektive uppgift anges inom parantes.

### Inga hjälpmedel!

Förklara bilder och begrepp noggrant!

**OBS! Nytt blad för varje uppgift och glöm inte skriva anonymitetskod.**

**LYCKA TILL!**

---

### Fråga 1. (5p) (Peter)

Att ändra innehåll i Document Object Model (DOM) är något du har kommit i kontakt med under kursens gång. Din uppgift blir därför att med utgångspunkt i HTML- och JavaScript-koden nedan (Se Figur 1 och Figur 2) färdigställa JavaScript-koden för funktionen `changeTextInDivs()`. I funktionen `changeTextInDivs()` skall du ändra texten i alla div-element som finns nästlade i ett annat div-element. Följande krav finns på din lösning:

1. Din lösning skall fungera oavsett om det finns text i ett div-element eller inte.
2. Texten i respektive div-element skall hämtas från `newTextArray` vektorn som finns i `changeTextInDivs()`. Mer konkret betyder det att texten i första div:en skall vara **Ett**, text i andra div:en **Två** osv. (se Figur 3).
3. Din lösning skall vara skriven så att antalet div-element kan ändras. Du kan dock anta att antalet div-element och antal positioner i vektorn alltid är samma.
4. Du får inte ta bort något från DOM:en utan endast skapa och ändra innehåll DOM:en.
5. För att lösa uppgiften skall du använda "rätt" DOM-metoder och DOM-egenskaper därför är egenskaperna `innerHTML`, `outerHTML`, `textContent` och `innerText` *inte* tillåtna att använda för att lösa uppgiften.

```

1  <!DOCTYPE html>
2  <html lang="sv">
3      <head>
4          <title>JavaScript för webbutveckling (ISGB15) 5hp</title>
5          <meta charset="utf-8">
6          <meta name="viewport" content="width=device-width, initial-scale=1">
7          <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
8              rel="stylesheet" />
9      </head>
10     <body>
11         <div class="jumbotron text-center">
12             <h1>JavaScript</h1>
13         </div>
14         <div class="container-fluid">
15             <div>Siffror</div>
16             <div></div>
17             <div>
18                 </div>
19                 <div> </div>
20                 <div> </div>
21             </div>
22             <script src="uppgiftx.js"></script>
23         </body>
24     </html>

```

Figur 1. uppgiftx.html

```

1  'use strict';
2
3  ~ function changeTextInDivs() {
4      let newTextArray = Array('Ett', 'Två', 'Tre', 'Fyra', 'Fem');
5
6  }
7
8
9  changeTextInDivs();

```

Figur 2. uppgiftx.js

```

<!DOCTYPE html>
<html lang="sv">
  <head>
    <title>JavaScript för webbutveckling (ISGB15) 5hp</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" rel="stylesheet">
  </head>
  <body>
    <div class="jumbotron text-center">
      <h1>JavaScript</h1>
    </div>
    <div class="container-fluid">
      <div>Ett</div>
      <div>Två</div>
      <div>Tre</div>
      <div>Fyra</div>
      <div>Fem</div>
    </div>
    <script src="uppgiftx.js"></script>
  </body>
</html>

```

Figur 3. DOM:en efter att JavaScript har exekverat koden

## Fråga 2. (5p) (Peter)

Att ändra struktur i Document Object Model (DOM) och skapa egna metoder är också något du har kommit i kontakt med under kursens gång. Din uppgift blir därför att med utgångspunkt i HTML- och JavaScript-koden nedan (Se Figur 4 och Figur 5) färdigställa JavaScript-koden för metoden `createHeadings()`. I metoden skall du skapa h-element (rubriknivåer h1-h6) och det baserat på till metoden inkommande värde. Rad 21-23 i JavaScript-koden är till för att testa din lösning och resultatet från en korrekt körning visas i Figur 6. Följande krav finns på din lösning:

1. Egenskaperna (`hRef`, `hTextRef` & `divRef`) och metoden (`checkNbr`) i objektet `oHeadings` skall användas i din lösning.
2. För att lösa uppgiften skall du använda "rätt" DOM-metoder och DOM-egenskaper därför är egenskaperna `innerHTML`, `outerHTML`, `textContent` och `innerText` *inte* tillåtna att använda för att lösa uppgiften.
3. Om inkommande värde ligger mellan 1 och 6 skall rubriknivåer från inkommande värde till och med 6 skapas (se Figur 6).
4. Texten i respektive rubriknivå skall vara en siffra som motsvarar aktuell rubriknivå (se Figur 6).
5. Om inkommande värde är mindre än noll eller större än 6 skall det till konsolen skrivas ut "Felaktig indata!".

```

1 <!DOCTYPE html>
2 <html lang="sv">
3   <head>
4     <title>JavaScript för webbutveckling (ISGB15) 5hp</title>
5     <meta charset="utf-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7     <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
8         rel="stylesheet" />
9   </head>
10  <body>
11    <div class="jumbotron text-center">
12      <h1>JavaScript</h1>
13    </div>
14    <div class="container-fluid"></div>
15    <script src="uppgifty.js"></script>
16  </body>
17 </html>

```

Figur 4. uppgifty.html

```

1 'use strict';
2
3 let oHeadings = {
4   href : null,
5   hrefText : null,
6   divRef : document.querySelector('div[class=container-fluid]'),
7
8   checkNbr : function( nbr ) {
9     if( nbr >= 1 && nbr <= 6 ) {
10      return true;
11    } else {
12      return false;
13    }
14  }
15 };
16
17 oHeadings.createHeadings = function( nbr ) {
18
19 };
20
21 for(let i = 0; i < 8; i++) {
22   oHeadings.createHeadings(i);
23 }

```

Figur 5. uppgifty.js

```

<!DOCTYPE html>
<html lang="sv">
  <head>
    <title>JavaScript för webbutveckling (ISGB15) Shp</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" rel="stylesheet">
  </head>
  <body>
    <div class="jumbotron text-center">
      <h1>JavaScript</h1>
    </div>
    <div class="container-fluid">
      <h1>1</h1>
      <h1>1</h1>
      <h2>2</h2>
      <h1>1</h1>
      <h2>2</h2>
      <h3>3</h3>
      <h1>1</h1>
      <h2>2</h2>
      <h3>3</h3>
      <h4>4</h4>
      <h1>1</h1>
      <h2>2</h2>
      <h3>3</h3>
      <h4>4</h4>
      <h5>5</h5>
      <h1>1</h1>
      <h2>2</h2>
      <h3>3</h3>
      <h4>4</h4>
      <h5>5</h5>
      <h6>6</h6>
    </div>
    <script src="uppgifty.js"></script>
  </body>
</html>

```

Figur 6. DOM:en efter att JavaScript har exekverat koden

### Fråga 3. (10p) (Pierre)

HTML-formuläret nedan innehåller ett antal formulärelement som ska fyllas i av en användare.

Din uppgift är att skriva kod för att validera formuläret innan det skickas till servern.

Följande villkor skall uppfyllas för att få full poäng på uppgiften:

- a.) Lägg på en lyssnare som lyssnar efter den händelse som inträffar när användaren skickar formuläret. När händelsen inträffar skall funktionen `validate` anropas. (1p)
- b.) Skapa funktionen `validate` samt lägg in kod i funktionen som förhindrar formuläret från att skickas till servern (2p).
- c.) Skapa upp struktur för felhantering med `try-catch`. I `catch`-blocket skall ett objekt-litteral innehållande en referens till det `html`-element som orsakat valideringsfelet samt ett beskrivande felmeddelande som text tas emot. Flytta koden som förhindrar att formuläret skickas och lägg den sist i `catch`-blocket. (2p)
- d.) I `catch`-blocket ovan skall följande ske:
  - a. Det element som orsakade valideringsfelet skall markeras med en röd kantlinje.
  - b. Sätt texten i `div`-elementet med `id="error"` till innehållet i felbeskrivningen från objektet som tagits emot i `catch`. (2p)
- e.) Skriv kod för att validera följande villkor, om villkoren ej är uppfyllda skall ett felobjekt innehållande elementet som orsakade felet samt en beskrivande text kastas.
  - a. Kontrollera att textrutan med `id=email` innehåller minst 6 tecken, annars kasta fel enligt ovan.
  - b. Kontrollera att radioknappen med `value=fish` är vald, annars kasta fel enligt ovan.
  - c. Kontrollera att "klockan 8:00-8:30" är valt i `select`'en, annars kasta fel enligt ovan.

## Kontakta oss

Fyll i formuläret nedan så återkommer vi till dig!

Fyll i e-post

Välj ditt ärende

- ☒ Kontakta mig om fåglar
- ☐ Kontakta mig om båtar
- ☐ Kontakta mig om fiske
- ☐ Kontakta mig om resor

Välj en tid när du vill bli kontaktad:

Utseende i webbläsare

```
<form method="post">
  <div id="error"></div>
  <label for="email">Fyll i e-post</label><br>
  <input type="text" id="email" placeholder="epost@epost.com"><br>
  <p>Välj ditt ärende</p>
  <input type="radio" name="arende" value="bird" checked> Kontakta mig om fåglar<br>
  <input type="radio" name="arende" value="boat"> Kontakta mig om båtar<br>
  <input type="radio" name="arende" value="fish"> Kontakta mig om fiske<br>
  <input type="radio" name="arende" value="travel"> Kontakta mig om resor<br>

  <label for="klockslag">Välj en tid när du vill bli kontaktad:</label>
  <select id="klockslag">
    <option value="8">klockan 8:00-8:30</option>
    <option value="8.5">klockan 8:30-9:00</option>
    <option value="9" selected>klockan 9:00-9:30</option>
  </select><br>
  <button type="submit">Skicka förfrågan</button><br><br>
</form>
```

HTML-kod för formulär

#### Uppgift 4. (5p) (Alexander)

Lyssnare är något som används dagligen inom JavaScript och i denna uppgift ska du skapa en interaktionsbar inköpslista. I funktionen `addClickListenersToButtons()` skall du lägga till en lyssnare som lägger till en vara i div-elementet med ID:t "list". Varan som du lägger till ska ha tre lyssnare. En lyssnare för att ta bort varan med ett knapptryck. Sedan ska varan ha en lyssnare för när muspekaren är går över elementet, då ska den lägga till klassen "font-weight-bold" på varan. Den sista lyssnaren ska köras när muspekaren lämnar elementet, då ska den ta bort alla klasser för varan. Här är en lista av krav på din lösning:

- Din lösning ska fungera oavsett om det finns fler eller mindre knappar med samma struktur som i HTML-dokumentet.
- Du ska avbryta default-beteendet för klick-händelsen på button-elementet.
- Varan som läggs till ska innehålla texten som finns i attributet "data-add".
- För att lösa uppgiften skall du använda "rätt" DOM-metoder och DOM-egenskaper därför är egenskaperna `innerHTML`, `outerHTML`, `textContent` och `innerText` inte tillåtna att använda för att lösa uppgiften.

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>JavaScript för webbutveckling (ISGB15) 5hp</title>
5    </head>
6    <body>
7      <div id="foodlist">
8        <div>
9          <p>Mjölk</p>
10         <button data-add="Mjölk">+</button>
11       </div>
12       <div>
13         <p>Bröd</p>
14         <button data-add="Bröd">+</button>
15       </div>
16       <div>
17         <p>Smör</p>
18         <button data-add="Smör">+</button>
19       </div>
20       <div>
21         <p>Skinka</p>
22         <button data-add="Skinka">+</button>
23       </div>
24     </div>
25     <div id="list"></div>
26
27     <script src="uppgiftx.js"></script>
28   </body>
29 </html>
```

```
1  'use strict';
2
3  function addClickListenersToButtons(){
4
5  }
6
7  addClickListenersToButtons();
```



### Uppgift 5. (5p) (Alexander)

När användaren öppnar HTML-dokumentet nedan skapas ett antal DOM-element. Redogör för hur innehållet i div-taggen med ID:t "list" ser ut efter att sidan laddats. Du ska alltså förklara hur DOM-strukturen ser ut, dvs. det som syns i inspekteraren/granskaren i webbläsarens utvecklarverktyg när du fäller ut div-elementet och eventuella underelement.

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>JavaScript för webbutveckling (ISGB15) 5hp</title>
5    </head>
6    <body>
7
8      <div id="list"></div>
9
10     <script src="uppgiftx.js"></script>
11   </body>
12 </html>
```

```

1  'use strict';
2
3  let data = [
4      {namn: "Volvo", modell: "V70", typ: "manuell", pris: 39999},
5      {namn: "Volvo", modell: "720", typ: "manuell", pris: 59999},
6      {namn: "Volvo", modell: "XC40", typ: "automat", pris: 299999},
7      {namn: "Volvo", modell: "S90", typ: "automat", pris: 399999},
8  ];
9
10 function generateCarInformation(){
11     let list = document.querySelector("#list");
12     let ul = document.createElement("ul");
13
14     for(let i = 0; i < data.length; i++){
15         let li = createCarListItem(i + 1);
16         createCarInformation(ul, li, data[i]);
17     }
18
19     list.appendChild(ul);
20 }
21
22 function createCarListItem(id){
23     let li = document.createElement("li");
24     li.setAttribute("class", "car car-container");
25     li.setAttribute("data-id", "car-" + id);
26
27     return li;
28 }
29
30 function createCarInformation(list, listItem, data){
31     listItem.textContent = (
32         "Namn: " + data.namn + " " + data.modell + " " +
33         "Typ: " + data.typ + " " +
34         "Pris: " + data.pris
35     );
36
37     list.appendChild(listItem);
38 }
39
40 generateCarInformation();

```