



# Inlämningsuppgift 3

Betyg: U/G

## Syfte

Syftet med den tredje inlämningen är att göra sig bekant med hur vi kan kombinera JavaScript och HTML - detta genom något som kallas för Document Object Model (DOM). Denna modell tillgängliggör ett gränssnitt för oss att kunna arbeta med JavaScript och HTML tillsammans. Detta innebär saker som att skapa nya, radera eller ändra på olika HTML-element i vårt HTML-dokument. Utöver denna interaktionen kommer vi även att använda oss av några inbyggda funktioner för interaktion med en användare (i detta fallet i formen av popup-rutor).

Inför denna uppgiften rekommenderas det starkt att läsa kapitel 12, 13, 14 och 18 i boken Eloquent JavaScript.

Det rekommenderas även starkt att ta en titt på följande källor (dessa kommer användas i uppgifterna):

- `alert()`
- `prompt()`
- `confirm()`
- `.addEventListener()`
- `.createElement()`
- `.children`
- `.firstElementChild`
- `.lastElementChild`

Längst ner finner ni kodexempel för att komma igång - ta en titt på dessa innan ni börjar.

## Uppgiften

Denna uppgift är indelad i mindre deluppgifter, det rekommenderas att försöka hålla sig till den ordning som presenteras (då svårighetsgraden ökar med varje).

Uppgifterna kan lösas på många olika vis, det finns alltså inte alltid ett korrekt svar



utan ibland kan det finnas ett flertal. När ni söker information via webben efter lösningar eller hjälp på vägen så rekommenderas det starkt att oavsett vad ni hittar - försök att förstå hur det fungerar - experimentera och utforska koden med hjälp av webbkonsollen.

Tips 1: testa gärna att klistra in er kod på **JSHint** för att se om den kan ge förslag på förbättringar. Det är en sida som följer vissa standarder för hur en bör skriva kod på ett korrekt vis.

Tips 2: testa gärna att använda er av tjänsten **Dirty Markup** som hjälper er formatera er kod.

**Glöm inte att läsa igenom kravspecifikation innan ni börjar, så ni inte missar någonting eller får komplettera på grund av slarvfel.**

## Uppgift 1

I uppgift 1 kommer ni bli tilldelad färdig HTML- och CSS-kod som ni kan placera i era egna filer. Det innebär att ni endast ska arbeta med JavaScript, dock är det valfritt om ni vill anpassa HTML- eller CSS-koden på någotvis - så länge ni kan färdigställa uppgiften.

Uppgiften går ut på att ni ska göra så att om en användare klickar på en av de tre knappar som finns så kommer en ny CSS-klass att läggas till på den `<div>` vi har med id "message-box". För att ytterligare förtydliga kan vi ta ett exempel: om en användare klickar på knappen med id "success" så ska er div få klassen "success", dvs. `<div id="message-box" class="success"` och på så vis kommer utseendet ändras på denna `<div>` i och med att vi har en CSS-klass med namnet "success".

Detta innebär att ni måste lägga till ett `click` event på respektive knapp och beroende på vilken knapp som blir klickad så ska knappens id läggas till som en klass på div:en.

### HTML



```
<div id="message-box">
  <p>This is a very important message box!</p>
</div>

<button type="button" id="success">Success</button>
<button type="button" id="error">Error</button>
<button type="button" id="info">Info</button>
```

## CSS

```
#message-box {
  border: 1px solid black;
  padding: 15px;
  font-size: 20px;
}
.success {
  background-color: #dff0d8;
  border-color: #98B98B;
}
.error {
  background-color: #f2dede;
  border-color: #BE9090;
}
.info {
  background-color: #d9edf7;
  border-color: #7294A5;
}
```

## Uppgift 2

I den andra uppgiften ska ni arbeta med att lägga till och skapa nya HTML-element. Mer specifikt ska ni göra så att en användare, när den klickar på en knapp, ska få fylla i en text som sedan läggs till som ett list-element i en lista. För att få en popup-ruta där användaren kan fylla i en text så ska ni använda er av den inbyggda funktionen `prompt()`. Nedan blir ni tilldelad färdig HTML för både listan (där list-element ska läggas till) och för knappen (den som använder sig av `prompt`-funktionen).

## HTML



```
<ul id="items">
  <li>The first item is free!</li>
</ul>

<button type="button" id="add-item">Add item</button>
```

För att förtydliga: när en användare klickar på knappen “Add item” så ska den få fylla i en text genom prompt-funktionen (t.ex “Hello World”) och sedan ska denna texten läggas till som ett nytt list-element i listan (t.ex `<li>Hello World</li>` ).

## Uppgift 3

I den tredje uppgiften ska ni komplettera *Uppgift 2* med en extra knapp. Ni väljer själv `id` och text. Denna knapp ska, när en användare klickar på den, radera det sista elementet i listan `<ul id="items">` .

Tips: använd er av `.lastElementChild` i kombination med `.removeChild` .

## Uppgift 4

I uppgift 4 kommer ni utgå från en lista innehållande list-element vars innehåll är en bit text samt en knapp med texten “X”. Tanken är att när en användare klickar på ett list-elements knapp så ska list-elementet raderas, dvs. att en användare ska kunna radera element från listan genom respektive knapp. **Dock** ska användaren godkänna detta utförande innan ett list-element raderas. Detta görs genom att ni använder er av den inbyggda funktionen `confirm()` .

### HTML

```
<ul>
  <li>
    This is the first item
    <button type="button" class="remove-list-item">X</button>
  </li>
  <li>
    This is the second item
    <button type="button" class="remove-list-item">X</button>
  </li>
  <li>
    This is the third item
```



```
        <button type="button" class="remove-list-item">X</button>
    </li>
    <li>
        This is the fourth item
        <button type="button" class="remove-list-item">X</button>
    </li>
</ul>
```

Tips 1: använd er av nyckelordet `this` i funktionen som hanterar ert `click` -event för respektive knapp, detta för att komma åt knappens förälder (dvs. list-elementet)

Tips 2: eftersom alla knappar har samma klass rekommenderas det att hämta alla dessa knappar och sedan lägga till ett klick-event för dessa - ta en titt på kodexemplet nedan för hur ni kan åstadkomma detta:

```
// Hämta alla knappar
var buttons = document.getElementsByClassName("remove-list-item")
// Gå igenom alla knappar och lägg till ett klick-event
for (var i = 0; i < buttons.length; i++) {
    buttons[i].addEventListener("click", function() {
        // ...
    });
}
```

## Uppgift 5

I uppgift 5 kommer vi att arbeta med formulär, denna uppgift är indelad i två delar:

1. Hämta innehållet från det ifyllda formuläret och skriva ut detta i webbkonsollen med `console.log()`
2. Validera formuläret innan det skickas. Här kan vi använda oss av `alert()` för att ge användaren ett meddelande om vad som var fel

Nedan finner ni den färdiga HTML-koden för formuläret. Det är givetvis tillåtet att skapa ett eget, så länge ni är noggranna med alla attribut, id:n och klasser. Lägg gärna också till lite egen CSS för att få det hela lite snyggare!

Börja med del 1 och gå sedan vidare till del 2.

### HTML



```
<form action="" method="get" id="apply-for-pet">
  <h2>Ansökan om att skaffa husdjur</h2>
  <input type="text" name="firstname"> Förnamn<br>
  <input type="text" name="lastname"> Efternamn<br>
  <input type="text" name="age"> Ålder<br>
  <input type="text" name="email"> Epost<br>

  <div id="pets">
    <p>Typ av husdjur?</p>
    <input type="radio" name="pet" value="dog"> Hund<br>
    <input type="radio" name="pet" value="cat"> Katt<br>
    <input type="radio" name="pet" value="fish"> Fisk<br>
    <input type="radio" name="pet" value="bird"> Fågel<br>
  </div>

  <button type="submit">Skicka ansökan</button>
</form>
```

## Del 1

Det första steget ni måste göra är att hämta formuläret ( `.getElementById()` ) och sedan lägga till ett event för `submit` (observera att vi inte lyssnar på ett klick-event utan vi lyssnar på när formuläret skickas iväg). I och med vi lyssnar på `submit` kan vi även göra så att avälja om formuläret ska skickas iväg eller inte, dvs. vi kan själva styra om formuläret ska få skickas iväg eller inte. Grunden för detta presenteras nedan:

```
var form = /* hämta formuläret */;

form.addEventListener("submit", function(event) {
  // Genom denna rad så avbryter vi att formuläret skickar oss
  event.preventDefault();

  // Om vi dock vill skicka formuläret vidare (vilket vi ska g
  // så hade vi skrivet följande, denna rad placeras lämpligen
  // en if-sats (t.ex. när vi validerat alla fältens innehåll)
  event.target.submit();
});
```



Observera att raden `event.target.submit()`; bör inte användas i del 1 eftersom det skickar iväg formuläret

Tanken med del 1 är att ni ska inuti ovanstående kod använda `console.log()` för att skriva ut alla värden från formuläret (förnamn, efternamn, osv.).

För att komma åt värdena från fälten i ett formulär ska ni nytta `this.elements` - denna innehåller alla våra fält. Hur vet vi vilket fält vi kan hämta då? Saken är det att om vi tar en titt på vår HTML-kod så ser ni att alla fält har attributet `name` med tillhörande värde (t.ex. "firstname"). Det innebär att om vi vill skriva ut förnamnet hade vi skrivit följande `console.log(this.elements.firstname.value);`.

Utifrån ovanstående ska ni nu kunna färdigställa del 1.

## Del 2

I del 2 så ska ni använda er av ett antal `if` -satser för att kontrollera värdena för respektive fält i formuläret. Nedan presenteras kraven för varje fält:

- Förnamn - får endast innehålla 0 till 50 bokstäver
- Efternamn - får endast innehålla 0 till 50 bokstäver
- Ålder - måste vara en siffra ( `number` ) och vara mer än 0
- Epost - får endast innehålla 0 till 50 bokstäver
- Husdjur - ett husdjur måste vara valt

Om alla dessa kraven stämmer för samtliga fält kan ni anropa `event.target.submit()` , om något inte stämmer ska ni använda er av `alert()` med ett relevant meddelande för vad som var fel.

Tips: använd er av en stor `if` -sats med `else if ()` och slutligen en `else { }` del.

## Kravspecifikation

- På första raden i er fil ska ni använda raden `"use strict";` för att göra webbläsaren mer strikt i sin tolkning av er kod
- Samtliga uppgifter ska markeras med en kommentar innan uppgiften innehållande uppgiftsnummret, dvs. `// Uppgift 1.` för första uppgiften osv.
- Det får inte visas några felmeddelande i webbkonsollen



- All JavaScript-kod ska placeras i en egen fil och länkas in till ert HTML-dokument
- All JavaScript-kod måste ha godtycklig indentering, dvs. allting får inte vara vänsterställt eller allt för svårt att läsa

## Inlämning

**Glöm inte kontrollera att ni skickat med svar på alla uppgifter och att ni följt kravspecifikationen.**

När du är färdig med din uppgift ska du ladda upp denna som en `.zip`-fil innehållande alla dina filer på Canvas (på samma sätt som inlämningsuppgift 1). Döp denna enligt formatet `inl3_Förnamn_Efternamn.zip`. Du ska även ladda upp dessa filer på dvwebb och därefter ska du även inkludera länken dit.

Lycka till!

## Kodexempel

### Att skapa ett element

```
// Skapa ett stycke
var p = document.createElement("p");
// Skapa en textnod
var textNode = document.createTextNode("Leo finally got an oscar");
// Lägg till textnoden på vårt stycke
p.appendChild(textNode);

// Alternativ till att ändra text på ett element
// p.textContent = "Leo finally got an oscar!";

// Hämta vår <body>
var body = document.querySelector("body");
// Lägg till vårt stycke sist
body.appendChild(p);
```

### Att radera ett element

```
// Vi utgår från följande HTML
// <p id="info">
//     Lite exempel innehåll <a href="google.com" id="link">här.
```





```
// </p>
```

```
// Hämta de element vi arbetar med
```

```
var p = document.getElementById("info");
```

```
var a = document.getElementById("link");
```

```
// Vi måste utgå från förälder-elementet när vi ska radera  
p.removeChild(a);
```

### Att lägga till ett event på ett element

```
// Hämta knappen vi vill ska ha ett klick-event på sig
```

```
var button = document.getElementById("my-button");
```

```
// Alternativ till att hämta ett element:
```

```
// var button = document.querySelector("#my-button");
```

```
// Lägg till ett klick-event
```

```
button.addEventListener("click", function(event) {
```

```
    // Den kod du vill ska utföras vid ett klick
```

```
    // ...
```

```
    // "this" representerar knappen som blev klickad
```

```
    console.log(this);
```

```
    // "event" innehåller information om eventet (klicket)
```

```
    console.log(event);
```

```
});
```

Detta projekt är publicerat som **öppen källkod på GitHub**.

Användning av allt material är fritt att tillgå genom **The MIT License (MIT)**.