

## 3.1.3 DOM-objekt

Uppgiften introducerar till DOM-objekt som ofta används som stöd i JavaScript-program. För att förenkla interaktionen med DOM-objektet från JavaScript kan man använda exempelvis jQuery.

### Uppgift

Gör ett eller flera JavaScript-program som använder DOM-objektet för att:

- Välja ut ett HTML-element
- Göra något med ett HTML-element:
  - Ändra en egenskap för ett HTML-elementet
  - Lägg till nytt innehåll till ett HTML-elementet
  - Ta bort ett HTML-element
  - Utvinna information från ett HTML-element
  - Lägg till och ta bort ett klass-attribut från ett HTML-element

Gör även (eller ihop med ovanstående) ett eller flera JavaScript-program som använder DOM-objektet för att reagera på följande händelser:

- Muehändelser (någon av `click`, `dblclick`, `mousedown`, `mouseup`, `mouseover`, `mouseout` eller `mousemove`)
- Dokument och fönsterhändelser (någon av `load`, `resize`, `scroll` eller `unload`)
- Formulärhändelser (någon av `submit`, `reset`, `change`, `focus` eller `blur`)
- Tangentbordshändelser (någon av `keypress`, `keydown` eller `keyup`)

Gör även ett program som använder den kombinerade händelsen `hover` (= `mouseover` + `mouseout`).

Använd även händelse-objektet `event` och någon av dess egenskaper (`pageX`, `pageY`, `screenX`, `screenY`, `shiftKey`, `which`, `target` eller `data`) för att exempelvis få fram koordinaterna för var en användare är med muspekaren.

Gör till sist ett program som:

- Stoppa en händelses normala beteende (`evt.preventDefault` eller `return false`)
- Tar bort en händelse (`removeEventListener`)

Alla JavaScript-program ska följa kraven under sidan [Krav](#).

### Exempel

Inget exempel ges.

### Tips

HTML-dokumentet måste (ofta) vara färdigladdat i webbläsaren innan JavaScript-programmet körs igång (eftersom programmet ofta använder element i HTML-sidan)

och det görs med jQuerys `ready`-funktion:

```
$(document).ready(function() {  
    // Egen kod här körs först när HTML-dokumentet är färdigladdat  
});
```

Ett snyggt sätt att knyta en händelse till en tagg är att använda jQuery:s `on`-funktion. Man kan ta bort en händelse med jQuery:s `off`-funktion.

Om man inte vill använda jQuery för detta så kan man istället göra enligt följande exempel:

Filen `Test.html`:

```
<!DOCTYPE html>  
<html lang="sv">  
  <head>  
    <meta charset="utf-8">  
    <title>Test</title>  
  </head>  
  <body>  
    <input type="button" value="Tryck mig" id="b"></input>  
    <script type="text/javascript" src="Test.js"></script>  
  </body>  
</html>
```

Filen `Test.js`:

```
function x() {  
    window.alert('Du klarade det!');  
}  
  
document.getElementById("b").addEventListener("click", x);
```

## Hjälp

Om man fastnar ordentligt så kan man få hjälp via epost:

[Klicka här för hjälp](#)

Om man ska sända med kod:

- Sänd endast med de delar som är relevanta för frågan
- Koden ska vara välkommenterad och välstrukturerad

Det är **ingen** handledning på kunskap som ingår i förkunskapskraven (kompileringsfel med mera). Epost sänds beroende på ämne till en specifik handledare som försöker svara snabbt men eftersom det ibland är väldigt många frågor så kan svaren dröja. Handledarresurserna fördelas jämt över studenterna så om man ställer många frågor så får man vänta längre tid på svar. Notera att handledarna enbart arbetar inom angivna handledningstider.

Den här sidan fungerar på både stationära och mobila klienter, har hög tillgänglighet för funktionshindrade och använder webbsäkra fonter. Den följer W3C:s strikta standard HTML5 och CSS 3. Den är genererad från textfiler som använder ett egendefinerat minimalt WIKI-baserat språk och en databas. Systemet och allt material är gjort av [Pierre A. I. Wijkman](#) och är skyddat av lagen om upphovsrätt.