

1. Gå till jQuery.com och ladda ner senaste stabila versionen av jQuery
2. Skapa en html-sida och länka in en ny js-fil. Länka också in jQuery innan den vanliga js-filen länkas in. Varför spelar ordningen roll?
3. För att få ordning på grunderna i jQuery behöver du lära dig hur du kan "starta" jquery för din sida. I din javascript-fil, lägg till en document.ready-funktion där du skriver ut en alert med någon text i. Kontrollera att denna alert kommer upp när du laddar om sidan.

När du har kommit så här långt har du en fungerande koppling till jquery. Bra jobbat!

I kommande övningar kommer du att få använda de vanligaste funktionerna i jquery. För mer information, titta på <https://api.jquery.com/>

4. I din html, skapa upp en div.
I din css, skapa upp en klass där regler för vidd, höjd och en ram läggs till.
I din js, lägg till css-klassen till din div från html-filen. Detta görs i document.ready. Funktionen som du bör använda är addClass från jquery.
5. När din div syns från förra uppgiften gör följande:
HTML: Skapa en knapp med ett id.
JS: Skapa en händelse för din knapp som skall köras när du klickar på knappen. Händelsen skall göra att din div långsamt försvinner från skärmen.
6. När din div försvinner från skärmen från förra uppgiften:
HTML: Inga ändringar
JS: Ändra din kod så att div:en kommer tillbaka varannan gång som du klickar på knappen och göms varannan gång du klickar på knappen. Det finns flera möjliga lösningar till denna uppgift.
7. I denna övning bör du skapa en ny html-fil, js-fil och css-fil.
HTML: Skapa en textruta tillsammans med en knapp.
CSS: Do your magic :)
JS: Använd jQuery för att skapa en händelse för knapptryckningen. I denna händelse skall texten från textrutan skrivas ut i consolen.
8. Från föregående uppgift, skapa en <p>-tagg som skall innehålla texten från textrutan och placera denna <p>-tagg efter knappen i HTML-strukturen. Observera att det inte skall ske någon förändring i HTML-dokumentet, bara i JS-filen.
9. HTML: Skapa en ul/li-lista med några punkter (kan vara vad som helst)
CSS: Skapa en klass som skall sätta färgen på en text.
JS: Skapa en funktion som byter färg på alla li-elementen på en gång.
10. HTML: Skapa upp två stycken textrutor (självklart med unika id:n). Skapa upp en knapp som kör en ny funktion i js.
JS: Skapa upp funktionen från den nya knappen. Nu skall du beräkna summan av talen som skrivs in i de två textrutorna och skriva ut detta på skärmen.
11. HTML: Skapa upp en textruta för inmatning av tal. Skapa upp en knapp för din js-funktion.
JS: När användaren trycker på knappen skall du kontrollera om talet som skrevs in var större än 100. Om så är fallet skriv ut: "Det var ett stort tal" på skärmen. Annars skriver du ut "Det var ett litet tal".
12. JS: Skapa en lista (Array) med texter. T.ex. var myArray = ["Katt", "Hund", "Fisk", "Häst"];
Skapa nu en loop som går igenom listan och skriver ut alla texter i listan på skärmen. Kom ihåg, använd jQuery för detta :)
Fråga: Hur skall texten skrivas ut? I en span, i en lista eller i en tabell?
13. Skapa upp en klass som heter Person. Egenskaperna i denna klass kan t.ex. vara FirstName, LastName, Adress, Phone, Email. Komplettera gärna med flera egenskaper som du tycker passar.
14. Skapa nu upp flera stycken person-objekt och spara dem i en lista.

15. Loopa igenom listan för förra uppgiften och skriv ut förnamn och efternamn på varje person på skärmen. Använd gärna samma teknik som i uppgiften med listan med texter ovan.
16. Ibland kan behovet av att skicka med parametrar till exempelvis click-händelsen uppstå. Läs på hur detta kan göras genom att titta på api.jquery.com, funktionen `.on()` och avsnittet *Passing data to the handler*. När du har läst detta avsnitt gör följande uppgift:
HTML: Skapa upp tre knappar med olika id:n och texter men med samma klass.
JS: Skapa upp en klick-händelse för klassen som du använde på knapparna ovan. Skapa nu logik (kod) för att skriva ut en godtycklig parameter som du skickar till din handler.
17. Anpassa handler-funktionen så att den skriver ut id:t på den klickade knappen istället för en parameter som du skickar in. Det finns flera sätt att lösa detta på.
18. Världigt ofta i den riktiga världen vet vi inte hur många saker som kommer att finnas på skärmen när vi börjar göra en applikation. Skapa upp en lista med texter i din JS. Loopa igenom denna lista och skapa upp element (valfri typ) som innehåller texten från din lista. Skapa nu upp en klickhändelse för dina element och hantera klickningen genom att skriva ut texten från det klickade elementet i konsolen.
19. Ibland kan det vara intressant att ta bort en händelse från element. Detta görs med funktionen `off()`. Testa att i handlern från förra uppgiften lägga till `off()` när ett klick inträffar och testa att det fungerar.
20. Istället för att hålla på med `on` och `off` när vi vet att en knapptryckning bara skall inträffa en gång kan du använda funktionen `one()`.
HTML: Skapa en knapp med en text.
JS: Bind en klickhändelse till knappen genom att använda `one()`.
Kontrollera att handlern bara körs en gång, även om du klickar flera gånger på knappen.
21. Använd jQuery för att implementera ett entertryck i en textbox. Detta är väldigt vanligt, t.ex. i er förra inlämningsuppgift. När en användare har skrivit någonting i en textruta och trycker enter, vill ni att samma sak som knapptryckningen på knappen bredvid, skall hända. Hur skall ni tänka här? Hur kan ni fånga en knapptryckning? Och hur vet ni att det är enter?
22. Skapa en `ul/li`-lista med ca tio rader. Använd nu jQuery för att lägga till klassen `odd` på hälften av dessa och även på andra halvan. Detta kan ni säkert göra enkelt med CSS, men hur fungerar det med JS?
23. Använd jQuery's `map`-funktion på en lista med objekt för att göra om objekten till en array av en klass som ni styr över. Tänk er att ni hämtar data från en extern datakälla. Dessa objekt kommer att vara, förhoppningsvis, strukturerade på samma sätt. Övningen är att ni skall göra om dessa objekt till dina egna objekt där du styr vad egenskaperna heter. Denna operation kallas för mappning, vilket metodens namn antyder.
Skapa alltså upp en array med godtyckliga, likadana, objekt. Kör sedan `map`-funktionen på denna array och se till att resultatet blir en array som innehåller objekt av en klass som du har definierat.
24. SVÅR: Använd `deferred` för att göra AJAX-anropet nedan.

AJAX

1. I dessa övningar kommer vi att använda oss av information från filmer. Det finns ett öppet API som heter OMDb som hämtar information från IMDB. För att vi skall kunna hämta information från denna datakälla behöver ni gå till följande url och registrera er för en gratis API-nyckel:
<http://www.omdbapi.com/apikey.aspx>
2. Skapa en ny HTML-sida och en ny JS-fil.
I din html: Skapa upp behållare för följande information:
 - År
 - Titel
 - Handling

- Bild

3. I din JS-fil: I document-ready, gör ett anrop till url:n som står beskriven på <http://www.omdbapi.com/>. Titta på hur informationen som kommer tillbaka från anropet ser ut. Vilket format har det? Finns det något sätt för oss att komma åt denna information från js? Finns det någon färdig funktion i jQuery som vi kan använda? Skriv ut informationen från anropet i fälten i din html.