

# Lab 3: Sänka skepp (Battleship) i javascript

## Överblick

Det här är en labb i att få dig bekväm med att använda javascript. Du är dessutom uppmanad att använda dig av CSS3. Alla javascript-bibliotek får användas, JQuery är till och med rekommenderat för att göra uppgiften enklare.

I den här labben så ska du göra en partiell (eller komplett) implementation av spelet Sänka skepp. Det finns många versioner av spelet och du bör välja en av de vanligare versionerna.

## Specifika krav:

1. Spelplanen måste vara genererad, du får inte hårdkoda en 9x9 grid.
2. Det ska vara enkelt att ändra storleken på spelplanen (det ska räcka att ändra värdet av en variabel på ett ställe i programkoden)
3. HTML-taggen `<table>` får inte missbrukas för layout av spelplanens olika delar. Tables ska endast användas för tabulering av data, [inte för layout](http://stackoverflow.com/questions/83073/why-not-use-tables-for-layout-in-html). (<http://stackoverflow.com/questions/83073/why-not-use-tables-for-layout-in-html>).
4. Programmet ska fungera i både Firefox och Chrome ( det här kravet borde vara trivialt)
5. Ni måste använda "use strict" i hela koden.
6. Du måste använda dig av ett linting-tool (du får konfigurera det hur ni vill, men ni måste använda ett.)
7. Använd ÅTMINSTONE 5 olika typer av CSS attribut. Koden som innehåller CSS deklARATIONER ska vara i en egen fil och ska inte blandas med HTML och JavaScript koden. Fokus är på att det ska vara lärorikt och roligt [snarare än vackert](http://www.themostamazingwebsiteontheinternet.com/). (<http://www.themostamazingwebsiteontheinternet.com/>).

## Definitioner

"ruta": Termen ruta i denna text är ett godtyckligt klickbart HTML-objekt av ditt val.

"Skepp": En ruta som är visuellt skilt från de andra klickbara HTML-objekt när det klickas på.

## Uppgift.

Du ska implementera spelet Sänka Skepp. Spelet antar att det är två spelare vid samma dator. Spelet består av två delar. Första fasen är placering av skepp och då är spelarna ärliga och tittar bort när den andra spelare placerar sina skepp. Under andra fasen så ska spelarna skjuta ned varandras skepp. Det är dock inte ett krav att båda gör sin fas ett innan del två utan uppgiften kan förenklas till att någon skjuter klart innan hen placerar ut sina skepp. Detta får du implementera hur du vill vara det går att spela .

Spelplanen ska bestå av 9x9 grid men det ska vara lätta att ändra spelplanens storlek.

Utplacering av skepp på spelplanen ska ske genom att man klickar på en ruta, om man vill ångra sin placering så kan man klicka på det redan utplacerad skeppet för att ta bort den.

Det ska finnas en start-knapp som startar spelet/går vidare till nästa fas.

Bas-versionen av spelet kommer inte att kolla att skeppens form är rätt, detta lämnas till bonusuppgiften.

Det ska finnas en mekanism för att spelarna ska kunna hålla koll på antalet placerade skepp, hur många man har sänkt samt totala antalet skott som har avfyrats. Det ska också finnas en knapp som rensar brädet som man kan använda när man har spelat klar.

Den här uppgiften är ganska fri i sin form, men spelet måste vara spelbart och ni får inte hitta på regler godtyckligt utan måste utgå från wikipedia eller annan källa för regler.

### **Tips:**

När du debuggar använd dig av konsolen för att skriva spårutskrifter.

För att öppna konsolen på Firefox, tryck Ctrl + shift + k.

[https://developer.mozilla.org/en/docs/Tools/Web\\_Console](https://developer.mozilla.org/en/docs/Tools/Web_Console).

För att öppna den i Chrome, tryck, Ctrl + Shift + j.

<https://developer.chrome.com/devtools/docs/console#using-the-console-api>

Du kan inspektera element i de flesta webbläsare för att få mer förståelse för vad som faktiskt händer.

### **References:**

Tips and trix:

<http://modernweb.com/2013/12/23/45-useful-javascript-tips-tricks-and-best-practices/>

A commonly used tutorial:

[W3schools tutorial](#)

Strict mode

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Strict\\_mode](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Strict_mode)

### **Frågor:**

Vad är skillnaden mellan == och ===? Vad är "best practice" gällande vilken man ska använda?

Vad gör "use strict"?

### **Bonus**

Modifiera din kod så att det bara går att placera ut skepp som är på rätt form. Med andra ord, de är raka och av längd 2, 3, 3, 4 och 5 (eller vad för regler du har använt).

Skeppen ska antingen placeras slumpmässigt (dator vs människa) eller manuellt (människa mot människa). Placerings-reglerna ska följas oavsett.

### **Frågor:**

Har javascript klasser? Är prototypes klasser? **Motivera ditt svar.**

Vad är callbacks i Javascript?

Hur används callbacks asynkront?