**Beskrivning**

Det här projektet består av en stor mängd uppgifter där jag isolerat små delar och gett dig ledtrådar på hur du kan lösa dem. Lite läsning kan behövas för att exempelvis förstå hur du skall göra exakt. (men sådant fuskar alla med innan man blir ordentligt van)

**Syfte**

Repetera och stärka dina javascriptkunskaper.

**Lärandemål**

Lära dig syntaxen för javascript

Lära dig hur javascript kopplas ihop med webbsidor

Lära dig hur javascript kan ”triggas” av olika händelser på en webbsida

Lära dig hur du kan påverka innehållet på en webbsida med javascript

**Lämplig litteratur inom området**

<https://www.youtube.com/watch?v=Kp3HGwlXwCk> (Miniprojekt 1)

(0:00 – 15:58) Kolla först, kolla sedan igen och skriv av och prova lite olika alternativ. Gör sedan uppgifterna.

Miniprojekt 2 - <https://youtu.be/Kp3HGwlXwCk?t=958>

Miniprojekt 3- <https://youtu.be/Kp3HGwlXwCk?t=2171>

Miniprojekt 4 - <https://youtu.be/Kp3HGwlXwCk?t=3306>

Miniprojekt 5 - <https://www.youtube.com/watch?v=Kp3HGwlXwCk>

Miniprojekt 6 – ingen länk då det är eget arbete

Miniprojekt 7 - <https://youtu.be/Kp3HGwlXwCk?t=5872>

Miniprojekt 8 - <https://youtu.be/Kp3HGwlXwCk?t=7081> (miniräknare)

Miniprojekt 9 – <https://youtu.be/Kp3HGwlXwCk?t=9213> Interaktiv meny

**Miniprojekt 1**

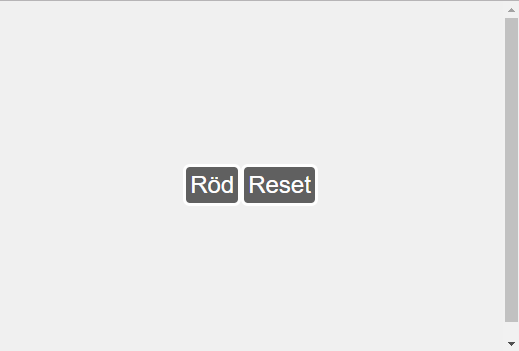
Steg 1

Titta på youtube-klippet och testa dig fram med hans första projekt. När du testat och fått det att fungera skall du göra några små ändringar i scriptet och förändra det så att det ser ut som bilden nedan. Du skall använda dig av två knappar där den första gör bakgrunden röd och den andra ändrar tillbaka till grå.

I JS-filen skall du använda dig av två funktioner som ändrar färgen på bakgrunden genom att använda sig av document.body.style.backgroundcolor.

Knapparna skall kopplas till funktionerna genom eventet onclick=

Lycka till!!!!



Steg 2

Ok, nu när du kan göra detta skall vi öka svårigheten lite.

Den första knappen skall nu ändras till Random och funktionen skall slumpa fram tre värden mellan 0-255. Dessa tre värden skall sedan användas för att sätta bakgrundsfärgen m.h.a.  
rgb(slump1, slump2, slump3);

Steg 3 (\*)

I det sista steget i den första övningen skall ni ändra eventet så att den första knappen startar en timer som triggar en funktion som slumpar färgen och som gör det varje sekund. Den andra knappen skall heta stopp och stoppa timern.

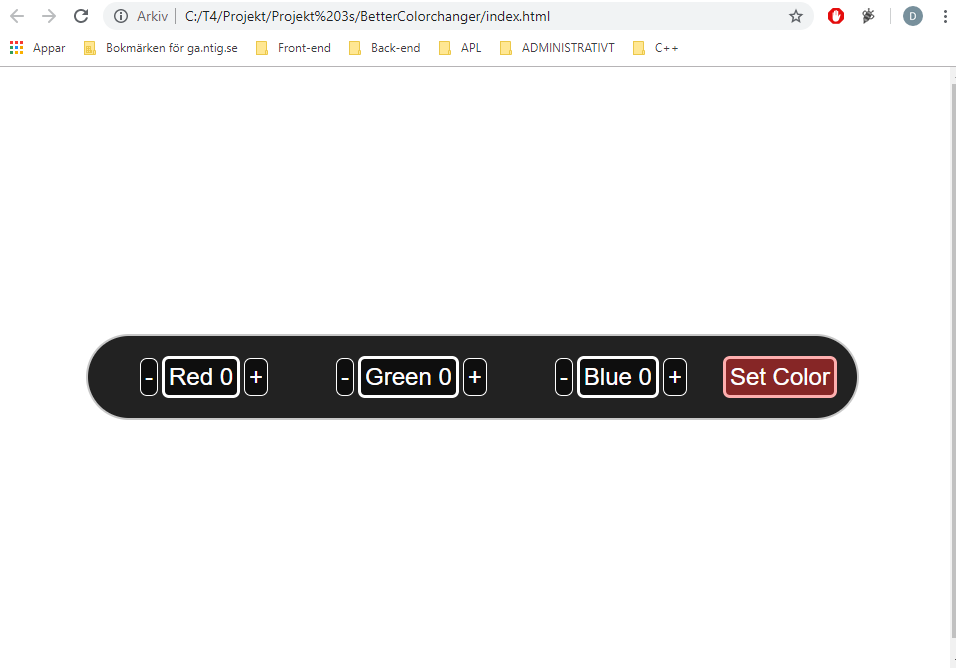
**Miniprojekt 2**

Du skall använda dig av dina kunskaper från andra delen av filmen och ändra hans program till att likna nedanstående variant. Exakt placering av knappar och liknande är inte så viktigt, men försök att göra det så gott du kan.

I exemplet nedan finns det tre knappar som innehåller texten Red 0, Green 0, Blue 0 och på varje sida finns en + och en -. När man klickar på – skall värdet minska i rutan till höger. Ifall du klickar på + skall värdet till vänster öka. (Ex du klickar 3 gånger på + till höger om Green 0. Då skall texten där ändras till Green 3.)

När du klickar på knappen Set Color skall du använda dig av de tre värdena för Red, Green och Blue och sätta bakgrundsfärgen till detta värde.

* Kontrollera att användaren inte minskar värdet till ett tal mindre än 0.
* Kontrollera att användaren inte ökar värdet till ett tal större än 255.



**Miniprojekt 3 - Movie quotes ( \* )**

Steg 1

Samla på dig 10 movie-quotes och lägg in dem i en objekt-array i ditt javascript. Du skall lagra citatet, vem som sagt det och i vilken film det förekommer. Citaten skall visas enligt nedanstående exempel.

*I've seen things you people wouldn't believe. Attack ships on fire off the shoulder of Orion. I watched C-beams glitter in the dark near the Tannhäuser Gate. All those moments will be lost in time, like tears in rain.*

***Roy Batty***in***Blade Runner***

Nedanför skall du lägga till två knappar. En som har texten ”Nytt citat” och som slumpmässigt väljer ett citat från listan av citat.

Den andra knappen skall ha texten ”Visa alla” och visa samtliga citat i formaterad utskrift.

**Miniprojekt 4 – Messenger ( \* )**

**Chatbox**

Designa en dialogruta enligt exemplet i youtube-klippet och lägg till två knappar istället för en. Den första knappen skall användas för att lägga till text i en kontainer nedanför knapparna. Ifall det redan finns text där skall den nya texten adderas till tidigare text, på ny rad.

Ifall man klickar på ”Send” med tom textruta skall man varnas.

Ifall ordet ”fuck” finns med i strängen skall man varnas – ordentligt och båda knapparna skall inaktiveras.

Den andra knappen skall användas för att rensa kontainern från tidigare text.

**Sammanfattning såhär långt.**

Du har nu lärt dig mer om CSS (förmodligen) och mer om HTML, men framförallt – du har lärt dig massor om JS.

Du kan nu göra enklare funktioner för att påverka innehållet på html-sidor. Du kan lägga till text, läsa av text-rutor, sätta intervall på olika funktioner, lägga till event-listeners och mycket mer.

Faktum är att du nu har tillräckliga kunskaper för ett flertal olika spel och i nästa projekt tänkte jag föreslå några enkla som du kan få prova dig fram med. Suveränt jobbat so far!!!

**Miniprojekt 5 ( \* \* )**



Det du skall göra i det femte lilla projektet är en box (lite snyggt formaterad) som innehåller en rubrik och en tidsangivelse i formatet hh:mm:ss. Denna tid skall uppdateras varje sekund och texten skall animeras in (fade in) så att opaciteten ökas från 0 till 1.0. Videon förklarar hur du gör detta.

Som du ser saknas knappar och allt annat som vi är vana vid. Den här gången skall klockan starta med en gång och sedan uppdateras varje sekund.

Nästa projekt kommer att bli ett litet enkelt spel som du skall få lösa 😊

**Miniprojekt 6 – Ett spel ( \*\* )**

I det här projektet är det äntligen dags för dig att få använda alla de kunskaper vi har samlat på oss och göra ett spel.

Spelet heter GuessMyNumber och går ut på att spelaren skall gissa vilket tal som datorn tänker på. Detta tal skall vara ett heltal mellan 1 och 1000.

När spelaren knappar in en gissning i ett textfält skall datorn jämföra det inmatade talet med det som datorn tänker på och tala om ifall det är rätt, för högt gissat eller för lågt gissat.

Programmet skall komma ihåg antalet gissningar och jämföra med tidigare high-score (low-score).

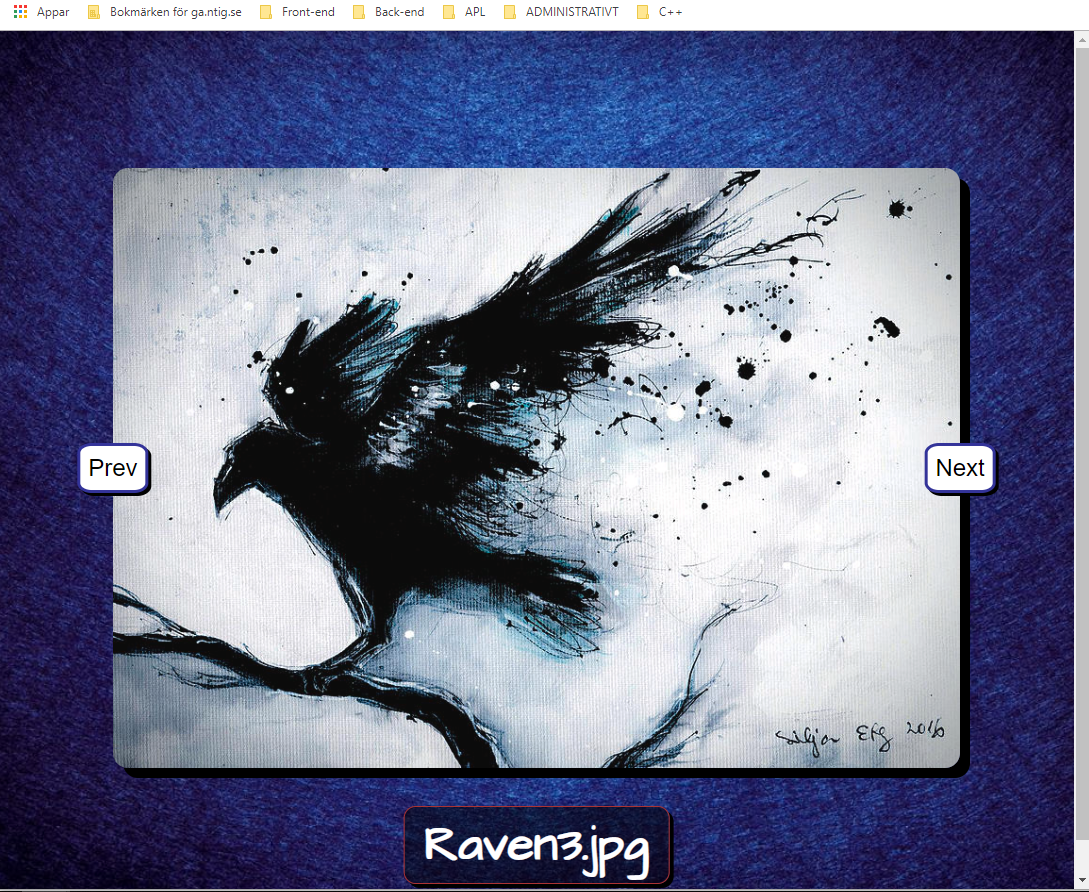
Programmet skall också komma ihåg hur lång tid som spelaren använt för att komma fram till rätt svar.

Om spelaren lyckas slå tidigare highscore skall man få knappa in sitt namn som också kommer visas tillsammans med tidigare rekord.

**Miniprojekt 7 – Bildvisare ( \* )**

Börja med att titta på videon för det här miniprojektet (länk på första sidan till rätt ställe) och häng med då han bygger upp sin sida.

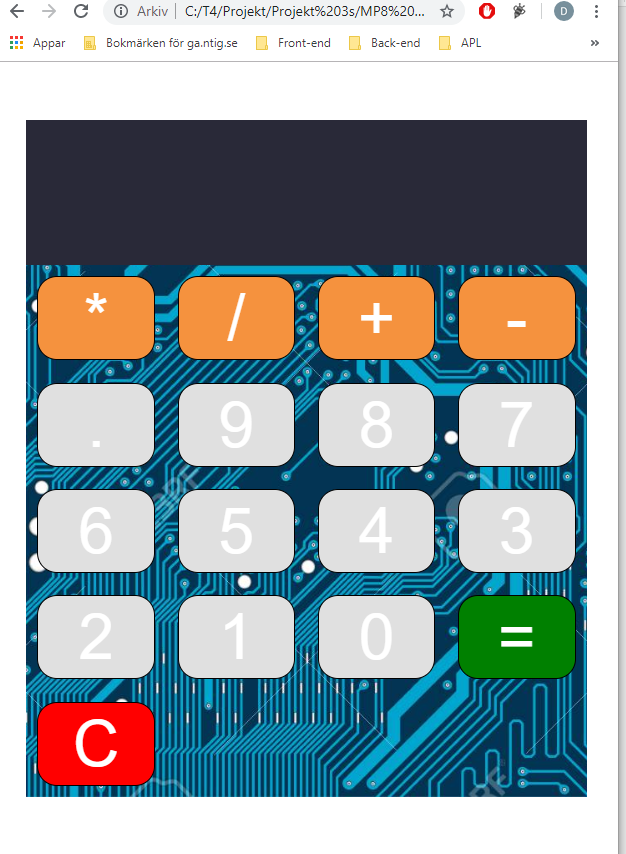
När du är klar skall du ändra lite i koden så att du längst ner visar en liten box med namnet på filen som visas just nu. Du skall också ändra så att programmet bygger på en lista med filnamn så att du närsomhelst kan lägga till en ny bild för visning utan att du behöver gå igenom koden för att ändra loopar och liknande. Du skall alltså göra som han själv säger, använda dig av bildarray.length för att hålla koll på antalet bilder. En annan fördel med filerna i en array är att de kan heta vad som helst.



**Miniprojekt 8 – Miniräknare ( \* \* )**

Det är dags för den klassiska miniräknaren. I videon beskriver författaren hur man lägger till knappar, kopplar event genom en loop med automatiska funktioner etc.

Det ni skall göra då ni är klara så långt videon visar är att modifiera räknaren något genom att lägga till en bakgrund som är något större än själva räknaren. Ni skall också minska ner knapparnas storlek något och runda av hörnen så att det ser ut mer som en räknare. Tänk på att sidan skall visas på samma sätt oavsett storlek (prova att ändra storlek på fönstret). Ifall knapparna inte får plats eller det skapas nya rader – tänk om.



**Miniprojekt 9 – Grafik ( \* \* )**

I det här projektet skall du få prova på att skapa lite grafik och skapa några algoritmer för att hantera objekt på skärmen. Du kommer att skapa och visa all grafik på en s.k. canvas som introducerades i HTML i v5. Själva elementet är sanslöst tomt när du skapar det och innehållet fixar du helt med javascript. När du skapar ett canvas-element kan du också lägga till en liten text i som används som fallback ifall klienten inte stödjer canvas (jösses). Ex

<canvas id=”canvas” width:400px height: 400px>  
 <h1>Your browser doesnt support </h1>  
</canvas>

Du behöver inte ge din canvas någon style men kan förse den med en border ifall du vill se den på skärmen innan du ritat något i den.

I ditt skript kan du sedan skapa en handle till din canvas med satsen

var canvas = document.querySelector(”#canvas”);

// sedan måste du få tillgång till ett grafikläge i din canvas (context) och det gör du medv

var gfx = canvas.getContext(”2d”); // skit i felhantering  
  
När du har 2d-kontexten lagrad i gfx kan du rita på din canvas, och ta bort 😊

Att rita en vit rektangel som är 50 pixlar bred och 150 hög gör du ex med

gfx.fillStyle = ”rgb(255,255,255)”;

gfx.fillRect(100, 100, 50, 150);

Vill du sedan ta bort den gör du det med

gfx.clearRect(0,0, canvas.width, canvas.height);

*Det var en crash-intro till canvas. Läs gärna mer på nätet (finns hur mycket som helst….)*

**Uppgiften du skall försöka göra är följande**

1. Skapa ett objekt (kvadrat) som åker omkring på skärmen i en rät linje. När den når en kant skall den studsa och ändra riktning.
2. När du lyckats få ett objekt att studsa omkring skall du lägga till ännu en kvadrat som studsar omkring och lever sitt eget liv. Du skall se två kvadrater som tuffar omkring på skärmen.
3. \*\* När de två kvadraterna stöter på varandra skall det också resultera i en studs.