Tutorial JavaScript SetBackgroundColor

**Programmering 2, 3. semester KEA, forår 2023.**

Denne tutorial er henvendt til jer som skal skrive jeres første GUI med JavaScript.

I en første GUI har man typisk 1 inputfelt, 1 trykknap og 1 funktion der er knyttet til trykknappen og som tager værdi fra inputfeltet.

I dette eksempel, taster man en farve ind i inputfeltet, og når der trykkes på trykknappen, sættes baggrundsfarven til denne farve.

Opret et nyt JavaScript project i IntelliJ. Jeg navngiver altid mine JavaScript projekter med småt.

Et billede, der indeholder tekst, skærmbillede, software, Computerikon

Automatisk genereret beskrivelse

Min IntelliJ ser sådan her ud:

Et billede, der indeholder skærmbillede, tekst, software, Computerikon

Automatisk genereret beskrivelse

Men det ændrer sig jævnligt præcis hvilke filer man får fra start.

**Det er vigtigt at slette package.json filen. Ikke for at køre eksemplet i dette dokument, men som en god vane, fordi Azure brokker sig hvis man prøver at deploye et JavaScript projekt med en package.json fil.**

Vi skal bruge en index.html, som er vores skærmbillede.

Et billede, der indeholder tekst, skærmbillede, software, nummer/tal

Automatisk genereret beskrivelse

Et billede, der indeholder tekst, software, Computerikon, Webside

Automatisk genereret beskrivelse

Og her er Html med inputfelt og button.

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Set background color</title>  
</head>  
<body>  
<h1>Ready for JavaScript</h1>  
<input type="text" class="inpColor" />  
<button class="pbSetColor">Push Me</button>  
<script src="js/setbackground.js"></script>  
  
</body>  
</html>

Så skal vi bruge en JavaScript fil, vi kan kalde den setbackground.js og lægge den i en js folder.

I ovenstående html er det det vi forventer, fordi vi har den her script linje.

<script src="js/setbackground.js"></script>

Og vi kan starte med følgende JavaScript i vores js fil:

let inp = document.querySelector(".inpColor");  
console.log(inp);  
  
let pbCol = document.querySelector(".pbSetColor");  
console.log(pbCol);  
  
pbCol.textContent = "Tryk mig for set color";  
  
let bdy = document.querySelector("body");  
console.log(bdy);

Vi skal nu prøve at køre vores program i en browser. En måde at gøre det på er at vælge index.html filen og så trykke på en af browser iconerne øverst til højre i IntelliJ.

Et billede, der indeholder tekst, skærmbillede, software, linje/række

Automatisk genereret beskrivelse

Hvis man kører ovenstående får man det her i sin browser.

Et billede, der indeholder tekst, skærmbillede, nummer/tal, Font/skrifttype

Automatisk genereret beskrivelse

Så skal vi skrive en funktion der tager værdi fra inputfelt og sætter baggrundsfarven.

function *setBackgroundColor*() {  
 let col = inp.value;  
 console.log(col);  
 bdy.style.backgroundColor = col;  
}

Lad os tilføje en css fil til projektet. Opret ny style.css i roden af projektet, f.eks. style.css

Normalt lægger vi css filer i en css folder, på samme måde som js filer.

Tilføj css fil til index.html. Ved feks blot at drag css fil til html fil.

<title>Set background color</title>  
 <link rel="stylesheet" href="style.css">  
</head>

Vi skal blot bruge denne css fil til lige at lege med backgroundcolor.

Skriv følgende i css fil.

h1 {  
 background-color: red;  
}

Et billede, der indeholder tekst, skærmbillede, software, Computerikon

Automatisk genereret beskrivelse

Før vi sætter background-color ser det sådan her ud.



Nu kan vi prøve at sætte en background color.

Det vi mangler er at tilknytte funktionen setBackgroundColor() til trykknappen som vi har kaldt pbCol.

pbCol.addEventListener('click', *setBackgroundColor*);

Vi bruger metoden addEventListener som tager 2 parametre, navnet på en event og en funktion der skal udføres når den event bliver trigget.

Prøv at køre og skriv f.eks. blue.

Et billede, der indeholder tekst, skærmbillede, Elektrisk blå, Farverigt

Automatisk genereret beskrivelse

Prøv skriv #ff0000aa

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, Rektangel

Automatisk genereret beskrivelse

Hvad foregår der her ?

Prøv skriv #ff0000, eller #00ff00, eller #0000ff

Du har sat farverne rød, grøn, blå, også kaldet RGB.

Og aa i ovenstående er et fjerde element som kaldes alphachannel, som angiver hvor gennemsigtig farven skal være.

Prøv gå til vores style.css fil.

Tryk på den lille røde firkant, til venstre for background-color: red;

Vores samlede farve består af en blanding af rød, grøn og blå. Der er ff antal værdier til hver, så altså 2\*4 bit, 8 bit, 256. Vi kan sætte 256\*256\*256 forskellige farver. Derudover alpha channel.

Et billede, der indeholder tekst, elektronik, skærmbillede, software

Automatisk genereret beskrivelse

Prøv peg med mus, og træk i vandret scrollbar.

Et billede, der indeholder tekst, elektronik, skærmbillede, software

Automatisk genereret beskrivelse

Man får både almindelige 10 tals værdier og hex numre for de 3 farver og alpha.

Vi kan prøve at tilføje en event listener til vores html document. Det betyder i princippet hele browseren. Vi sætter en lytter på når en hvilken som helst tast på tastaturet slippes efter at have været trykket ned ’keydown’. Denne event hedder ’keyup’. I kan prøve begge event keydown og keyup, og se forskellen.

document.addEventListener('keyup', changeColorBody);

Nu kan man virkelig eksperimentere med forskellige farver.

Vi kan sætte fontfarven på trykknappen med følgende linje kode.

pbCol.style.color = col;

Vi tilføjer endnu et inputfelt, og bruger det til at skrive værdien af vores valgte baggrundsfarve.

Tilføj de her 2 linjer html.

<br>  
<input type="text" class="inpColorValue" />

Prøv skriv værdien af backgroundColor i vores nye inputfelt.

inpColorValue.value = bdy.style.backgroundColor;

Prøv skriv værdien af backgroundColor i vores nye inputfelt.

Bemærk formatet af color i vores nye inputfelt.

Opret nu et nyt inputfelt, denne gang med type=color. Bemærk i øvrigt hvor mange forskellige typer inputfelter der findes i html. IntelliJ hjælper.

Et billede, der indeholder tekst, skærmbillede, display/skærm/fremvisning, software

Automatisk genereret beskrivelse

Til at identificere inputfeltet bruger vi denne gang id= i stedet for class=.

Så skal vi oprette følgende variabel i vores kode. inpColorPicker.

Et billede, der indeholder tekst, Font/skrifttype, linje/række, skærmbillede

Automatisk genereret beskrivelse

I stedet for document.querySelector, bruger vi document.getElementById.

Bemærk IntelliJ hjælper os med at finde elementet i html. Det er dumt at stave forkert når man indtaster referencer til html elementer i sin JavaScript kode.

I øvrigt kan vi godt bruge const i stedet for let. Vores inputfelt variable tildeles ikke nye værdier, så const er det rigtige at bruge.

Skriv en ny function, vi kan kalde den useColorPicker, og kopier koden fra setBackgroundColor.

Erstat den første linje i useColorPicker med at hente color fra vores nye variabel.

Tilføj en eventlistener til inpColorPicker. Læg her mærke til hvor mange forskellige Event der er at vælge imellem.

Et billede, der indeholder tekst, skærmbillede, software, display/skærm/fremvisning

Automatisk genereret beskrivelse

Her er link til en liste af de fleste html events.

<https://www.geeksforgeeks.org/html-event-attributes-complete-reference/>

Et billede, der indeholder tekst, skærmbillede, software, Operationssystem

Automatisk genereret beskrivelse

Her til sidst kan vi rode med at sætte fonten på noget tekst.

Vi kan f.eks. kopiere den ”p tag” linje nogen gange.

<p>Sæt noget tekst her</p>

Som jeg har gjort ovenfor, til at starte med for, at kunne se color blive skrevet i inputfeltet.

Opret en variabel der peger på alle vores ”p tags”.

Det gøres sådan her.

const pTags = document.getElementsByTagName('p')

I et hug får vi alle elementer med et bestemt html tag.

Prøv at skriv ud i console, hvilken klasse pTags er en instans af.

En klasse der ikke kan særlig meget ser det ud til. Det ser ud til man er nødt til at bruge gammeldags ”for i” loop for at hente, hvert enkelt element.

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, nummer/tal

Automatisk genereret beskrivelse

Hvordan ser vi nemt hvilke metoder der er i HTMLCollection ?

Kig nøje i browser – Console. Hvor vi har skrevet med console.log().

Et billede, der indeholder tekst, skærmbillede, software, Webside

Automatisk genereret beskrivelse

Umiddelbart ser vi indholdet at vores instans af HTMLCollection.

[[Prototype]]: indeholder metoderne i HTMLCollection. Så vi prøver at åbne ved at trykke på den lille pil til venstre for.

Et billede, der indeholder tekst, skærmbillede, nummer/tal, Font/skrifttype

Automatisk genereret beskrivelse

Der er kun 3 metoder, item, length og namedItem.

Vi kan eksperimentere i browseren.

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

Automatisk genereret beskrivelseDet et det gode ved at JavaScript er et fortolket sprog.

Vi kan se at HTMLCollection nedarver direkte fra Object.

Et billede, der indeholder tekst, håndskrift, Font/skrifttype, skærmbillede

Automatisk genereret beskrivelse

Ovenstående er i princippet en fin Class Browser, vi kan altid se hvilke metoder et JavaScript object implementerer.

ChatGpt foreslår bl.a. at bruge class Array til at få html elementerne fra HTMLCollection .

Et billede, der indeholder tekst, elektronik, skærmbillede, software

Automatisk genereret beskrivelse

Lad os gøre som ChatGpt foreslår.

Et billede, der indeholder tekst, Font/skrifttype, skærmbillede, linje/række

Automatisk genereret beskrivelse

IntelliJ hjælper os igen. from<T> , from en parameteroverført type og en instans der implementerer interfacet, Iterable. Hvor kommer alt det med generics og typer fra ? sagde vi ikke at JavaScript var uden erklæring af typer ?

Prøv at klik på from i Array.from. Der åbner sig følgende fil.

Et billede, der indeholder tekst, skærmbillede, software, nummer/tal

Automatisk genereret beskrivelse

Noget med core.d.ts til sidst i filnavn. ts står for TypeScript.

Så selvom vi siger vi bruger plain JavaScript, så benytter vi åbenbart alligevel noget TypeScript ind imellem, uden rigtig at være klar over det. I kan spørge ChatGpt hvad TypeScript er.

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

Automatisk genereret beskrivelse

Det er danskeren Anders Hejlsberg der står bag sproget TypeScript. Anders står også bag Turbo Pascal, .Net og C#.

https://www.aarthiandsriram.com/p/our-dream-conversation-anders-hejlsberg

function *increaseFont*(element) {  
 let fontSize = element.style.fontSize;  
 console.log(fontSize)  
}

Vi skriver en function som kan forøge vores p tags fontsize med 1, hver gang vi kalder den.

Til at starte med skriver vi font size ud.

Vi kalder vores increaseFont() function for hvert element med følgende linje:

pArray.forEach(*increaseFont*)

Et billede, der indeholder tekst, Font/skrifttype, linje/række, nummer/tal

Automatisk genereret beskrivelse

Der kommer ikke nogen font size ud. Bare 8 blanke linjer. 8 betyder 8 linjer af det samme som så er trukket sammen til 1 linje i vores Console.

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

Automatisk genereret beskrivelse

Nu har vi ovenstående kode.

Af en eller anden grund får vi ikke nogen værdi fra style.fontSize kaldet.

Vi er nødt til at sætte fontSize til en værdi. Det kan vi gøre sådan her.

element.style.fontSize = 10 + 'px'

I kan nu prøve at løse følgende opgave.

Tilføj 2 trykknapper til skærmen. Den ene knap øger vores ”p tags” font size med 1, den anden knap mindsker vores ”p tags” font size med 1.