



SKAPA EN INLOGGNINGSSIDA

Inlämningsuppgift - JavaScript introduktion



DEN 11 SEPTEMBER 2020

WIVIANNE GRAPENHOLT
grwi20ql@student.ju.se

Innehåll

Länkar	2
Funktion	2
Inloggnings-vyn:	2
Fel lösenord-vyn:	2
Välkomst-vyn:	2
Tack för besöket-vyn:	2
Reflektion	2
Planering	2
Hinder	2
Arbetsätt	3
Lösningar	3
Metoder för att manipulera DOM:	3
Villkor	3
Metoder för localStorage:	3
Metoder för kontroll:	3
Kvarstående problem	4
Sammanfattning	4

Länkar

<https://github.com/WiviWonderWoman/Inloggningssida.git>

<https://wiviwonderwoman.github.io/Inloggningssida/>

Funktion

Först kontrollerar koden om någon är inloggad genom en If-sats som kontrollerar om något värde är sparat i localStorage. Om det stämmer anropas funktionen som visar välkomst-vyn, annars anropas istället funktionen för inloggnings-vyn.

Inloggnings-vyn:

När ingen är inloggad anropas alltså funktionen för inloggnings-vyn.

Då töms först sidan på eventuellt innehåll därefter renderas en rubrik, tvåinmatningsfält med tillhörande etiketter samt en knapp för att logga in.

När besökaren klickar på knappen sparas de inmatade värdena för att jämföras med det förutbestämda användarnamnet/lösenordet.

Om värdena stämmer överens sparas de i localStorage och funktionen för välkomst-vyn anropas.

Fel lösenord-vyn:

Om däremot värdena inte stämmer så töms sidan på innehåll och en text med felmeddelande visas tillsammans med en knapp för att försöka igen. Ett klick och användaren är tillbaka på inloggningssidan.

Välkomst-vyn:

I det fall någon redan är inloggad eller precis har loggat in anropas alltså denna funktion som renderar välkomst-vyn. Sidan töms först på eventuellt innehåll för att sedan infoga text och visar knapp för utloggning.

Tack för besöket-vyn:

När användaren klickar på logga ut töms localStorage, sidan töms på eventuellt innehåll och visar en annan text samt en tillbaka-knapp. Om besökaren klickar på denna återvänder hen till inloggningssidan.

Reflektion

Som inledning vill jag påpeka att jag tolkar "...reflektion kring de **metoder** som använts..." på två sätt:

1. De JavaScript metoder jag använt
2. Mer allmänt – planering, pseudokod, flödesschema...

Därför försöker jag täcka in båda aspekterna.

Planering

Först försökte jag få grepp om funktionaliteten och flödet med hjälp av papper och penna. Dels genom att skriva pseudokod men även skissa upp ett flödesschema. Jag försökte att i förväg tänka ut vilka variabler som behövdes och hur dessa skulle namnges på ett tydligt sätt. Jag fick ju också möjlighet till handledning innan jag på riktigt började skriva koden vilket hjälpte mig att förtydliga strukturen.

Hinder

Det som kändes mest som en utmaning var att veta vilka metoder jag skulle använda för att byta och visa olika innehåll. Även hur jag skulle formulera villkoret för att kolla om någon är inloggad kändes först svårt.

Arbetssätt

Jag känner också ett behov av att verkligen skriva och testa kod för att helt greppa hur jag ska lösa svårare problem, "learning-by-doing". När jag jobbar på det viset har jag insett vikten av att dokumentera processen dels för att minnas vad som fungerade och inte samt för att försöka lära mig inför planering av kommande projekt.

När jag tog mig an själva kodningen genom att skapa och testa en liten del / funktion i taget, med intentionen att bara bli klar med just den delen. Så visade sig att det fungerade över förväntan och på köpet löstes liksom de flesta problem även för övriga delar. Det är ju ett ganska repetitivt flöde.

Lösningar

Jag har alltså valt att lösa uppgiften med två huvud-funktioner som båda har under-funktioner för de två ytterligare vyerna. Detta kändes mest naturligt eftersom "tack för besöket" och "fel lösenord"-vyerna är följder av "logga in" eller "logga ut" knapparna.

Metoder för att manipulera DOM:

Jag har valt att använda `insertAdjacentHTML` för att det kändes smidigt och bekant att skriva i html. Och för att det är möjligt att styra var innehållet hamnar i förhållande till övriga objekt på sidan. Det gjorde det även möjligt för mig att enkelt lägga in radbrytningar för att snygga till utseendet på sidan.

För mina knappar använde jag istället `createElement` och `appendChild` dels för att det inte var viktigt att styra placeringen eftersom min intention ändå var att de skulle visas efter övrigt innehåll. Och dels för att det kändes enklare att "komma åt" med `addEventListener`.

Villkor

För att kontrollera om någon är inloggad på sidan valde jag att kolla `localStorage.length`. Alltså om det fanns två värden sparade så var någon inloggad. Det kändes som den enklaste lösningen för just denna uppgift. Ett alternativ hade ju kunnat vara att tex kolla om något värde stämde överens med de förutbestämda.

Metoder för localStorage:

De metoder jag använt för att spara inloggningsuppgifterna är:

- `setItem` vilket helt enkelt stoppar in ett värde i `localStorage`, jag blev dock lite förvånad att det värdet jag stoppade in först hamnade på index 1 och det senaste på index 0, alltså fylls det på framifrån.
- `clear` som helt enkelt tömmer hela `localStorage`.

Metoder för kontroll:

Genom hela arbetet att lösa uppgiften har jag använt `console.log` flitigt! Jag testade även att stänga ner webbläsaren och att byta webbläsare, när jag var inloggad för att säkerställa att det fungerade.

Metoder för snygg kod och god praxis:

För att förhindra användande av ej deklarerade variabler inleder jag `script.js` med `'use strict'`.

Jag har försökt att inte "pollute the Global Scope" genom att skapa ett objekt med properties istället för variabler. Vilket ger en trevlig bieffekt: "passing by referens" alltså att kunna skicka in ett värde i en funktion som ändrar värdet och att detta värde sedan "lever kvar" utanför funktionens scope. De enda variabler som deklarerats enskilt är de tillfälliga som används för att lagra och jämföra inmatade värden. Jag hade en liten idé att skriva ut användarnamnet i välkomstmeddelandet men detta hade ju krävt att jag flyttade upp de som properties till objektet och jag är ärligt talat så nöjd att allt funkar så det får vara som det är.

In funktionen för inloggningssidan så blev `insertAdjacentHTML` väldigt lång därför valde jag att använda ``backticks`` eftersom det tillåter mig att byta rad mitt i en string samt hade möjliggjort ett enkelt sätt att inkludera användarnamnet i stringen med hjälp av `${användarnamn}`.

Jag vet inte om min stil för indentering och struktur följer praxis men jag tycker det ser snyggare ut och är lättare att följa strukturen om de begynnande måsvingarna är på en ny rad.

Namngivning av variabler och funktioner tycker jag är beskrivande och jag har genomgående använt "camelCase". Jag har valt en del svenska för att det känns enklast när allt är nytt. Mina kommentarer är helt på svenska av samma anledning.

Kvarstående problem

Jag har upptäckt att jag får felmeddelande i consolen när jag testat att logga in med fel användarnamn och lösenord, klickar 'försök igen' och när jag då skriver in rätt, klickar på logga in får jag följande fel: *script.js:54 Uncaught ReferenceError: namnFält is not defined at HTMLButtonElement.<anonymous> (script.js:54)*. Men eftersom allting fortfarande fungerar som det är tänkt och jag inte har någon lösning så låter jag det vara.

Sammanfattning

Jag tycker att utseendet på sidan är helt ok utan någon styling alls. Jag är positivt överraskad, stolt och väldigt nöjd med hur och att jag lyckats få allt att fungera.