

Inlämning: Inloggningssida

Githubpages: <https://sunandacarlstrom.github.io/inloggningssida/>

Repo: <https://github.com/sunandacarlstrom/inloggningssida>

Jag började med att gå igenom uppgiften, planera vad sidan ska innehålla samt vad som ska visas på skärmen före, under och efter inloggning. När jag väl satte igång med att skriva kod började jag med att skriva alla element som jag ville ha med på min sida i HTML. Sedan skapade jag ett dokument för CSS och ett för JavaScript som jag länkade från HTML-dokumentet.

Därefter designade jag mina element i CSS eftersom jag tyckte det var smidigare att göra det där än i JavaScript. Jag valde dock att skapa en 'logga ut'-knapp i Javascript istället för i HTML och återanvände style av 'logga in'-knappen som skrevs i CSS. Jag är nöjd med att jag justerade i min HTML så jag även fick lära mig hur det fungerar att skapa element i JavaScript.

När jag väl började skriva i JS-dokumentet började jag med att referera till viktiga element från HTML-dokumentet, d.v.s. till de element som jag ska vill sätta "events" på. Exempelvis kontrollera om användaren är inloggad eller inte med funktionen `checkLoginStatus`. Om inloggningen är sant d.v.s. användarnamnet och lösenordet har skrivits in korrekt så ska den utföra en annan funktion inuti den (s.k. nästlad funktion). Denna funktion döljer HTML-taggar med id som referens. Det innebär att fältet där man skriver in användarnamn och lösenord ska döljas på sidan när användaren har loggats in.

Jag skapade en 'onclick'-funktion som jag tror motsvarar `addEventListener('click')`. Funktionen är uppbyggd av "if-else statements" vilket innebär att den utför först det som står inom "if", och om det villkoret inte stämmer d.v.s. skulle vara falskt så går den vidare till nästa del "else if", och om inte det villkoret heller stämmer d.v.s skulle vara falskt så går den vidare till sista "else".

Funktionen `loginBtn` hämtar information om användaren, i detta fall Sara, när hon har tryckt på knappen 'logga in' lagras denna information i s.k. local storage. Men bara för att det är lagrat i local storage så betyder det inte nödvändigtvis att användaren fortfarande är inloggad. Jag har därför valt att hämta information från local storage för att hemsidan ska förstå att användaren fortfarande är inloggad även om hon stänger ner webbläsaren eller uppdaterar sidan. Jag sparade alltså informationen i local storage för att hämta information från local storage till en variabel som heter `loginStatus`. Denna variabel är satt till sant så länge användaren har loggat in med rätt uppgifter

`username == "Sara" && password == "qwe123"`

När Sara är inloggad och trycker på logga ut så ska hon loggas ut genom `loginStatus` som ändras från att vara sant till falskt. Därefter skrev jag att i `localStorage` ska användarnamn och lösenord tas bort för att ingen annan användare efteråt ska komma åt Saras uppgifter. Tillslut ska sidan laddas om.

Längst ner i min kod för funktionen `loginBtn` ville jag att ett varningsmeddelande ska dyka upp på skärmen ifall inloggningsuppgifterna ej stämmer.