Oppgave 2b - Forklaring av kode oppgave 2a

GLOBALE VARIABLER

|  |
| --- |
| let randomIndex = '';  let FirstRandomWord = '';  let SecondRandomWord = ''; |

VISNINGSFUNKSJON

Hovedfunksjonen bygger et spill med 10 linjer med to ord i hver linje som mangler noen bokstaver. Inne i funksjonen gis variabler og det kjøres en while loop som begrenses av (lineCount) denne økes med 1 for hver loop og stopper på 10. I loopen kaller flere hjelpefunksjoner som lager spillet og legger dette ut til html dokumentet.

|  |
| --- |
| function generateWordPuzzles() {    let lineCount = 0;  while (lineCount < 10) {    getSecondRandomWordFromList();  createPuzzelwords();  document.getElementById('app').innerHTML += /\*HTML\*/`  <div>  ${firstPartOfWord1}\_\_\_ \_\_\_${lastPartOfOtherWord}  <button onclick="this.innerHTML = '${FirstRandomWord} ${SecondRandomWord}'">?</button>  </div>  `;  lineCount++;  }    } |

HJELPEFUNKSJONER

Denne funksjonen henter ut et tilfeldig ord fra listen inne i (words.js), dette er det første ordet.   
en variabel (randomIndex) settes til et tilfeldig tall ved hjelp av en innebygget funksjon i js.  
Math.random som gir et tilfeldig tall mellom 0 og 1, tallet ganges med lengden på listen (words.js). Deretter brukes (Math.floor) for å fjerne tall bak komma ved å runde av ned til nærmeste heltall. Så benyttes tallet som er gitt til (randomIndex) for å hente ordet som ligger på angitt index i listen (words.js) og legges til i variabelen (FirstRandomWord).

|  |
| --- |
| function getFirstRandomWordFromList() {              randomIndex = Math.floor(Math.random() \* words.length);              FirstRandomWord = words[randomIndex].trim();          } |

Denne funksjonen bruker samme kode som getFirstRandomWordFromList() men kaller funksjonen først for å finne det første ordet. Så brukes en if statement som sier at vis randomIndex ikke er samme tall som indexen til FirstRandomWord i listen words.js, så utfør koden, ellers om den er lik kall funksjonen getSecondRandomWordFromList() en gang til.

|  |
| --- |
| function getSecondRandomWordFromList() {              getFirstRandomWordFromList();              randomIndex = Math.floor(Math.random() \* words.length);              if(randomIndex !== words.indexOf(`${FirstRandomWord}`)) {                  return SecondRandomWord = words[randomIndex].trim();              } else {getSecondRandomWordFromList();}            } |

Denne funksjonen setter variabelen firstPartOfWord1 til å være lik FirstRandomWord men uten di 3 siste bokstavene. Og lastPartOfOtherWord til å være lik SecondRandomWord men uten di 3 første bokstavene.

|  |
| --- |
| function createPuzzelwords() {              firstPartOfWord1 = FirstRandomWord.substr(0, FirstRandomWord.length - 3);              lastPartOfOtherWord = SecondRandomWord.substr(3);          } |

REFACTORING

Refactoring handler om å gjøre kode mere lesbar samt fjerning av gjentagelser som skaper en mer kortere, ryddig og forståelig kode.

Fordelene med å gjøre arbeid som må refactoreres etterpå er at det er lettere å dele opp problemløsning i forhold til hva du skal lage i mindre deler.

Ulempen er den at om du er godt over kompetent og føler man kan gå rett til mål med koden, så vil det å måtte sette koden opp enkel og med mye innhold først og refactorere etterpå bruke mer tid.