

Deleksamen 2

Vi har laget et Stein-Saks-Papir-spill, men mangler noen ting for å få det komplett. Bruk gjerne penn og papir for å tegne/skissere sammenhengen i spillet og koden.

Åpne `rock_paper_scissors.html` fra mappen dere har lastet ned i en teksteditor. Les koden, jobb med å forstå hva den gjør. Når dere er kjent med koden, åpne filen i en nettleser, og løs følgende oppgaver som er presentert i kronologisk rekkefølge, enklest til vanskeligst. Kommenter koden du endrer eller lager!

Oppgave 1

Åpne konsollen i nettleseren du har åpnet `rock_paper_scissors.html` i. Bruk denne for å løse opp feilene i koden (4 feil). Det kan være at du må klikke på ting og spille spillet for å finne alle feilene. Når feilene er rettet, skal spillet være spillbart.

Oppgave 2

I koden er det tre markerte, åpne kommentarblokker. I disse, forklar hva koden i linjene etterspurt gjør der det står *_ditt svar her_*.

Oppgave 3

Gjør om bakgrunnsfargen i `<div>` med id `winner` til gullfarget (`#FFD700`) hvis brukeren vinner spillet ved hjelp av JavaScript.

Hint: https://www.w3schools.com/jsref/prop_style_backgroundcolor.asp [Lenker til en ekstern side.](#)

PS: Det betyr at fargen bør endres tilbake igjen dersom utfallet blir at computeren vinner, eller det blir uavgjort.

Oppgave 4

Etter telleren for antall spill har blitt oppdatert (første gang, altså til 1), oppdater fargen på tallet til rød (se fargekode i CSS-delen av kode) ved hjelp av JavaScript.

Oppgave 5

Nå skriver spillet ut en fast streng når en spiller vinner: "The winner is " og vinneren. Lag betinget logikk som heller skriver ut

- "You win" hvis spilleren er vinner
- "You lost" hvis datamaskinen er vinner
- "It's a tie" hvis spiller og datamaskin har valgt samme spillertegn

Oppgave 6

I koden finnes en teller for å telle antall ganger man har spilt. Nå skal vi også telle hvor mange ganger spilleren vinner, datamaskinen vinner, eller det blir uavgjort.

Du trenger altså tre variabler som skal telle. Disse kan eksempelvis hete `playerWins`, `computerWins` og `draws`. Deklarer disse under counter-variabelen i koden. Analyser kodeflyten (hva skjer når) for å oppdatere disse tellerne på riktig sted.

I HTML-koden for utskrift av telleren, legg til nødvendig tekst og HTML-elementer med id for å oppdatere denne linjen til

Number of games played: 0. Player wins: 0. Computer Wins: 0. Draws: 0.

Tallet 0 i utskriften skal naturligvis oppdateres når tellerne oppdateres i koden.

Oppgave 7

Les dokumentasjonen om den innebygde funksjonen [setTimeout\(\) fra W3Schools](#). Legg til `setTimeout(function() { //kode som kjøres... }, 1000);` slik at koden bruker 1 sekund før den viser vinneren i et spill.

Oppgave 8

If-testene som sjekker hvem som vinner er lite effektive. Vi har informasjon i koden som bør gjøre det enkelt å sjekke for tilfeller av uavgjort. Skriv om denne if-testen til å være en if-else-test, hvor betingelsen i if-testen sjekker om spiller og datamaskin har valgt samme tegn og setter `winner`-variabelen til `"undecided"`. De resterende nødvendige if-sjekkene som setter `winner` til å være `"player"` eller `"computer"` nøstes inn i else-delen av logikken.

Oppgave 9

Skill ut if-testen fra oppgave 8 til en egen funksjon kalt `checkWinner()`. Funksjonen må ta imot to parametere, en for spillerens valgte tegn og en for datamaskinens valgte tegn. Funksjonen må returnere hvem som vinner (`"player"`, `"computer"`, `"undecided"`). Når funksjonen er klar, må resten av koden oppdateres til å bruke denne funksjonen, og ikke variabelen `"winner"`.

Oppgave 10

I mappen ligger en bildefil kalt `gun.png`. Gjør de nødvendige oppdateringene/utvidelsene av koden for å kunne spille med Pistol (gun) i tillegg til Stein, Saks og Papir. Dette endrer reglene noe, og må tas hensyn til i koden; Stein slår Pistol, Pistol slår Saks, Saks slår Papir, Papir slår Stein.