

Problem 1

Fremgangsmåte: I problem 1 skal vi generere en kode som lager en tilfeldig hexadesimal fargekode. Dette var den jeg hadde størst problemer med å starte på. Det jeg startet med var å se hva slags kode vi hadde fått fra før av. Det var hvilke «karakterer» vi kunne bruke og en array.form som (etter min forståelse) går at disse «karakterene» får individuelle elementer. Jeg fant en nettside (sjekk litteraturliste) som fikk meg godt i gang med dette problemet. Den fikk meg til å få en god start, og hva slags syntaxt jeg skulle bruke. Denne nettsiden viste meg hvordan jeg skulle bygge opp koden min, og måtte deretter navngi de riktige funksjonen til riktig navn. For eksempel `function.randomHexCode` og `let random_color_code`. Jeg så at denne nettsiden brukte «while». Jeg lærte at dette lager en løkke slik at hexadesimal koden går gjentakende og følger de riktige kravene. Hvordan «mattestykket» skulle være, løste jeg ved å ta i bruk med en tidligere forelesning vi hadde om functions. Her skulle vi returnere tilfeldige tall mellom a og b. I dette problemet var prinsippet mye av det samme, bare at vi måtte bruke de verdiene vi hadde fått fra før av, som allerede var i koden.

Sammenligning med AI: Kodene har noen likheter med hverandre, men jeg syntes at løsningen jeg fant selv var mer forståelig. Det jeg syntes var enklere med å forstå var i «while» løkken hvor den forteller at det må være mer enn sju elementer i bruk. I AI koden lager den sin egen versjon hvor den forteller først at $0 = i$, og at i er < 6 . Selv om jeg syntes dette også er en fin kode, så er det litt forvirrende med å legge til at $0 = i$. AI bruker en for løkke istedenfor while, og det vil jeg si gir mer mening enn det jeg gjorde fordi for-løkke bruker man når man vet hva de ulike verdiene er.

Problem 2

Fremgangsmåte: I problem 2 skulle vi finne gitte ord som var i en array. Dette problemet var litt lettere fordi jeg hadde en viss ide om at jeg måtte bruke if og else. Jeg brukte også en nettside for å se syntaxt på dette problemet (se litteraturliste). På dette problemet brukte jeg `include()` metoden som vist på nettsiden. På nettsiden viste dem ikke til hvordan man

kunne få dette rett inn på HTML. Bare inn på konsollen. Dette var det største problemet, men etter litt tid fant jeg ut av det. Jeg måtte lage en let funksjon som er i endring. Så denne tok plassen til console.log. Deretter la jeg inn (`Ja`) på if, og (`Nei`) på else. Deretter var det å få det inn på HTML siden, og ikke på konsollen.

Sammenligning med AI: Her har AI gjort det annerledes og viser til kode som jeg ikke skjønner noe av. Koden AI har brukt er færre linjer og mindre repetativt arbeid. Istedenfor å lage en if og else på hver av ordene, har den lagd en funksjon for alle sammen som sier at svaret skal enten være Ja eller Nei om ordet eksiterer. Funksjonen den har lagd er også en include, bare at den har brukt include(ord) som er en innebygd JavaScript metode som finner spesifikke ord i en array. Det var fint at den forklarte dette, fordi jeg føler at AI lærte meg her hvordan man kan løse problemet på en mye lettere måte og med mindre linjer.

Problem 3

Fremgangsmåte: I problem 3 skulle vi finne ut av gjennomsnittet til en tallrekke i en array. Under forelesningene har vi sortert tallrekker, men jeg har aldri vært borti det å finne ut av gjennomsnitt. Jeg lette gjennom alt fra forelesningene, men jeg fant ingenting. Her brukte jeg en nettside hvor det var en som viste en eksakt kode for å vise til gjennomsnitt (se litteraturliste). Her bruker man reduce metoden for å få arrayen til en rekkefølge og til en verdi. Du må dermed lage en funksjon hvor du finner ut av summen til tallrekken. Da bruker man reduce metoden, og tar totalen sammen med numrene. Deretter for å finne ut av gjennomsnittet, så gjør man slik man har blitt lært på skolen. Man tar summen og deler det på antall siffrer. Det er også et problem som oppstod. I html koden fikk begge den samme tall-arrayen. Denne id'en var avg_array_one. Jeg syntes dette var rart, men prøvde å bruke den på begge arrayene. Når jeg testet, funknet dette ikke. Jeg ble fortalt at to ulike arrayer ikke kunne ha samme id. Dette visste jeg fra før av og ble ikke overrasket. I oppgaven står det at koden ikke trengs å forandre på, men dette var det ene unntaket. Jeg forandret id'en på den andre arrayen til avg_array_two, og dette funknet. Jeg vet ikke om dette var en feil i selve koden eller om det er jeg som har misforstått.

Sammenligning med AI: AI har også brukt reduce metoden, men har helt andre verdier. Istedenfor å bruke total og number, så bruker den acc og val. AI forteller meg at acc er en start verdi på 0, men val er hvert enkelt tall i arrayen. AI har også rettet på den id'en som jeg nevnte tidligere. Den spesifiserte dette punktet også når jeg spurte hva den hadde gjort. Jeg syntes at det er lettere å lære av den koden jeg hadde siden den bruker tekst som kan være lettere og er mer tydeligere. Selv om prinsippet er relativt likt, så syntes jeg at man lærer lettere av den første koden fordi man ser og forstår den lettere.

Bruk av AI

Jeg valgte å bruke AI på den mest lettvinnte måten. Jeg kopierte hele koden som vi hadde fått. Jeg tok med kommentarene som var i koden slik at den forstod hva den skulle løse. Ut i fra dette skjønnte den alt den skulle gjøre, og jeg fikk en komplett kode som fungerte fint. Jeg fikk ikke tatt skjermbilde av samtalen, men som sagt så limte jeg inn koden, og fikk tilbake en fullstendig kode med problemene løst.

Litteraturliste:

Microsoft Copilot. (2025, October 14). JavaScript-løsninger på tre programmeringsproblemer [AI-generert innhold]. Microsoft. <https://copilot.microsoft.com>

Pixi, E. (2020, februar 14). *What the Hex!?* (How to generate random hex color codes in javascript). DEV Community. <https://dev.to/thecodepixi/what-the-hex-how-to-generate-random-hex-color-codes-in-javascript-21n>

SitePoint. (2022, juni 9). *Understanding how average of an array works—JavaScript*. SitePoint Forums | Web Development & Design Community. <https://www.sitepoint.com/community/t/understanding-how-average-of-an-array-works/388466/3>

TutorialPoints. (2025, mars 10). *How to find specific words in an array with JavaScript?*

<https://www.tutorialspoint.com/how-to-find-specific-words-in-an-array-with-javascript>

Link til repository: <https://github.com/StormyDoom/Arbeidskrav2> Jesper Simensen