

Sammanfattning U1

-- Sammanfattning av instruktionerna för uppgift 1 (U1) -
programmering för webben

- **Flow Chart** - göra ändringar för en "något annorlunda version av Averager-appen". Vi rekommenderar att du studerar den efter det att du har kodat klart Averager.

När du väl har förstått kopplingen mellan FlowChart och koden så ska du skapa din egen FlowChart för en modifierad version av Averager: **AveragerGroupSame**.

Denna version av appen fungerar exakt likadant som vanliga Averager med en skillnad: När användaren *markerar* en siffra så kommer alla de andra likadana siffror att automatiskt markeras också. Detsamma gäller om användaren *avmarkerar* en siffra.

Du behöver inte koda detta, bara modifiera FlowChart för att vara en lösning på detta nya problem

- Viktigt att lösa problemen i ordning (1, 2, 3 etc.)
- All kod du skriver ska hanteras via en **repository** (bara en repository för alla deluppgifter) på Github.
 - Du ska också ladda upp på Canvas in en **zip-fil med all kod** som du skapar. Använd mappstrukturen och filerna som redan finns i mappen Averager. Ingen av mina videos ska dock finnas med i det du lämnar in
- Glöm inte att inkludera en länk till din Github-repository. **Det ska göras i filen Averager/AveragerApp/index.js**. Se instruktionerna där för mer info.
- Kan lämna in samma kod som andra, men måste spela in **egna videos och lämna in i egen inlämning på Canvas**.

- Viktigt att använda korrekt terminologi i videos (strikt, kan bli komplettering). Spela dessutom in video som förklarar ändringarna du gjort i Flow Chart.

Variabler	En variabel "hämtas" inte. En variabel måste deklarerar . Den kan också få ett värde "assignat" (tilldelat) till sig (eller ett värde kan sparas i den), eller värdet som finns sparad i den används på något sätt.
For-loopar	For-loopar (och andra loopar) itereras Varje "körning" (fel ord) är en iteration (rätt ord) Variabeln som "räknar" i loopen kallas counter . Counter inkrementeras / dekrementeras i varje iteration Inför varje iteration kontrolleras om villkoret uppfylls (är sant) En for-loop avslutas när villkoret inte längre fylls.
Array	En array sparas aldrig i en variabel. Det är bara referensen till arrayen som sparas i variabeln. I övningen "More fun with Arrays" från lektion 4 står det att man inte brukar prata så mycket om referenser till arrayer och det stämmer. Men i U1 måste ni göra det. En array består av ett gäng värden (eller element). Varje element har sin plats i arrayen. Platsen identifieras med en index -siffra. I lektioner har Erik kallat platsen "cell". På engelska kallas den " slot ", detta är mer korrekt.
Funktioner	En funktion "hämtas" inte. En funktion måste deklarerar . En funktion " tar emot " parametrar . Dessa parametrar får ett värde när funktionen " anropas ". " Argument " används ofta som synonym för "parameter". Vid korrekt bruk ska "parameter" användas när man pratar om funktionsdeklaration (då vet vi inte vilket värde parameter kommer att få). "Argument" ska användas när man pratar om funktionsanrop (och vi vet vilket värde det har). Det är ok om ni, i denna övning, använder bara "parameter" eller bara "argument" när ni förklarar er kod. En funktion kan ha ett returvärde (markeras med kommandot <i>return</i>). I så fall returnerar den det värdet som skrivs efter <i>return</i> . Annars returnerar funktionen värdet <i>undefined</i> .
HTML-element	HTML-element "hämtas" inte. Ett korrekt sätt att uttrycka sig är "Referensen till elementet X sparas i (eller assignas till) variablen Y." .querySelector() returnerar referensen till ett element, inte elementet i sig.

- **Style guide**

- All kod måste vara **korrekt indenterad** (2 spaces per nivå). Detta gäller HTML-, CSS- och JS-kod. Notera att vi inte kommer att lämna feedback om var i koden ni har indenterat fel.
- Ni får inte använda några globala variabler i JS-koden. Ni **måste dock använda globala funktioner**.

- Alla namn på variabler, klasser, id:en, funktioner, etc måste vara genomtänkta och professionella.
- Koden i JS-filen ska följa denna struktur (notera att det skiljer sig från tidigare övningar):
 - A. **Globala variabler** (Ni inte får dock inte ha några globala variabler i AveragerApp)
 - B. **Funktionsdeklarationer** (tidigare har det sagts att detta ska komma sist)
 - C. **Eventlisteners** för befintliga HTML-element
 - D. **Direktkod**