

# Inlämningsuppgift 1 – Planering och förutsättningar

## Länk till källkoden på GitHub

<https://github.com/NikBjo72/evemarketing>

## Metod och teknik

Jag har valt att använda GIT och GitHub för min källkodshantering. Metoden jag använder är Gitflow. Eftersom jag nu är ensam i projektet så blir det inte alla steg, utan jag pushar direkt till main. Dock försöker jag hyfsat ofta göra commits för att det ska finnas återställningspunkter ifall något tokigt händer med koden under utvecklingen. Jag använder mig också av appen Fork, men mest för att få en övergripande bild. Kommandon skriver jag ofta i terminalen.

## Fortsatt arbete med källkodshantering

Jag har valt att använda GIT eftersom det är ett verktyg som jag hyfsat kan och känner mig bekväm med. GIT är också ett vanligt använt verktyg i branschen vilket dels gör att jag vill lära mig mer men också att det ger en indikation på att det är ett bra verktyg.

Jag använder GitHub som lagringsenhet av källkoden. Dels flyttas då koden utanför huset/datorn, vilket gör att det är en mycket bra backup, men det ger också utrymme för att arbeta från andra datorer, andra platser, jobba flera samt dela med sig av koden och arbetet. Appen Fork har jag valt att använda för att få en grafisk bild över kodens utveckling. Jag kan där tydligt se brancher, commits och kommentarer. I terminalen har jag inte samma överblick.

Min valda metod är GitFlow. När jag startar ett nytt projekt gör jag ett repository på GitHub. Klonar sedan ner det till en mapp på min dator. Mappen öppnar jag i VS Code. I VS Code skapar jag alla nya mappar, filer och koden. Jag använder mig av den inbyggda terminalen i VS Code för att lägga till filer i GIT, commita samt pusha upp till GitHub. Även om jag inte gjort det ännu, så kommer jag i framtiden skapa olika brancher i projektet beroende på vad som ska göras. Skulle jag t.ex. vilja skriva om koden på ett nytt sätt är det en fördel att göra en ny branch för just den ändringen, och då kanske utgå från en viss punkt (commit) i main-branschen. Att jobba utifrån GitFlow är bra, dels för att det är ett genomtänkt flöde, men också för att lära sig ett arbetssätt som många i branschen använder. Det gör det enklare att snabbt kunna sätta sig in i ett arbetssätt eller i ett projekt.

## **Java Script-modulers påverkan på kodstruktur**

Användandet av Javascript-moduler påverkar kodens struktur på många sätt. Dels så blir det en ytlig struktur som gör att det är lättare att få en överblick av kodstrukturen, eftersom (om man gör så) skapar modulerna med samma namn som funktionsnamnen. Jobbar man då också enligt OOP och har tydliga namn vad de olika modulerna gör (har för funktion) så ger överblicken över mappstrukturen och modulnamnen en direkt tydlig struktur av koden. Ligger allt i en main.js finns inte denna överblick.

Med moduler jobbar man tydligare med abstraktion. Dels blir det så eftersom varje modul måste importeras för att kunna användas. Det gör att man noggrannare måste fundera på varje moduls uppgift. Strukturen påverkas också av att man måste exportera funktioner, samt importera i modulerna. Koden kommer på det sättet få ett tydligare flöde, då det i en js-fil lätt kan bli ostrukturerat, men kan ändå fungera. En fördel är också att man direkt i en modul kan se vilka andra moduler som används, genom importen.

Genom modulanvändande hamnar också koden i "strict"-mode, vilket kräver bättre skriven kod. Genom att använda moduler hamnar inte heller så många variabler i "global scope", vilket ger en säkerhet i kodstrukturen. Det gör också att vi kan använda samma variabelnamn i modulerna och behöver inte "tävla" om namnen och risken att skriva över en variabel.

Genom att jobba med moduler ger vi helt enkelt koden en tydligare struktur, det gör den mer lättöfläst för andra utvecklare samt lättare att uppgradera eller skriva om. Ännu en fördel med att jobba med moduler är att fler utvecklare jobbar på ett liknande sätt, vilket minskar tröskeln att läsa/ändra i annan utvecklarens kod.