Asynkron vs Synkron

Synkron kod: En sak i taget

```
doSomething1();
for (let i = 0; i < 10000000; i++) {
  console.log(i);
}
doSomething2();
```

Som vi redan vet kommer koden ovan att exekveras i tur och ordning. Först funktionen doSomething1 utförs, sedan for-loopen, och när loopen är klar kommer doSomething2 att utföras.

Om koden ovan ingår i en webbsida, kommer webbläsaren, när koden körs, att bli "unresponsive". Det spelar ingen roll om användaren klickar på något, webbläsaren är upptagen med att köra koden och känner inte av användarens handlingar.

Om det tar lång tid kommer webbläsaren att meddela något i stil med "a script is making this page unresponsive...".

Detta är ganska sällan ett problem, eftersom t.o.m. vanliga bärbara datorer idag är väldigt kraftfulla och klarar av koden ovan relativt snabbt (fast just console.log kan ta tid!).

Synkron kod: Problematiskt när man ska hämta resurser

Problemet när man ska hämta resurser från webben är att det kan ta lång tid även om datorn är kraftfull, eftersom det inte handlar om datorn utan om internet.

Om webbsidan behöver ladda stora mängder data, eller om uppkopplingen är dålig, så kan det ta tid. Redan någon sekund väntetid känns av.

Med synkron exekvering av koden skulle det bli väldigt tålamodsprövande att besöka en sida som behöver hämta en del data (FB, Insta, Tiktok, Google Docs, etc).

Därför hanteras ALL kommunikation med servrar asynkront. Dvs:

1. Förfrågan skickas till servern.
    1. Med hjälp av metoden then() anger vi vilken funktion som ska utföras när responsen har kommit in.
2. Under tiden som webbläsaren väntar på responsen så utför den instruktionerna (raderna) som kommer efter fetch. Den känner också av användarens handlingar (click, etc) under tiden.
3. När svaret är på plats så utförs funktionen som angavs i steg 1.

Asynkron kod: Raderna fortsätter att utföras

```
doSomething1();
fetch(request).then(handleResponse);
doSomething2();
```

Det som kommer att hända i koden ovan är:

1. doSomething1 utförs
2. fetch utförs:
    1. förfrågan skickas till servern
    2. vi anger vilken funktion som ska anropas när svaret har kommit in (handleResponse)
3. doSomething2 utförs
4. När svaret från servern kommer in anropas (och utförs) handleResponse.

Notera:

A. Koden väntar INTE vid fetch (som den hade gjort om exekveringen hade varit synkron) utan går vidare till nästa rad och utför den. Hade det funnits fler rader efter doSomething2() så hade de också utförts, i tur och ordning.
B. Det är BARA fetch som är asynkron, resten av koden är synkron och utförs som "vanligt" (i tur och ordning)