Flytkontroll, loops

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Loops_and_iteration>

Løkker, eller loops som det heter på engelsk, brukes når du har kode som skal bli gjentatt flere ganger. Det finnes flere forskjellige typer løkker, og det er opp til deg å velge hvilken løkke som er mest formålstjenlig til det du driver med. For å klare alle disse oppgavene må du lese dokumentasjonen i lenken nøye.

**Oppgave 1**

Dette er en ganske stor oppgave! Navngi de 5 forskjellige typene løkker/looper som er forklart i dokumentasjonen, og forklar hvordan de virker. Du kan ignorere labeled loop. Bruk gjerne et praktisk eksempel på hver av dem.

for (let i = 0; i < 5; i++) {

console.log(i);

}

For loop: Den gjentar en blokk med kode et bestemt antall ganger

let i = 0;

while (i < 5) {

console.log(i);

i++;

}

While loop: gjentar en blokk med kode så lenge en spesifik betinhelse er sant så f.eks while true eller while tall < 10

let i = 0;

do {

console.log(i);

i++;

} while (i < 5);

Do while loop: fungerer på samme måte som en while loop bortsett at den garanterer for at koden blir kjørt misnt en gang selv om betingelsen er ikke sann eller falsk da.

const person = { navn: Simon, alder: 16, kjønn: helikopter };

for (let key in person) {

console.log(key + “: “ + person[key]);

}

Forr in loop: den iretrerer gjennom egenskapene til et objekt

const tall = [1, 2, 3, 4, 5];

for (let num of tall) {

console.log(num);

}

For of loop: iretrer gjennom elementene i en iterable som f.eks en array

**Oppgave 2**

**const navneliste = ['Geir', 'Espen', 'Ellen', 'Erik', 'Lars', 'Gunnar', 'Oda', 'Nina', 'Tine', 'Henrik', 'Agnethe'];**

Lag en løkke som skriver ut alle navnene i denne listen og som sier hvor mange bokstaver de har. Forklar hvorfor du valgte akkurat den løkken du valgte.

for (let navn of navneliste) {

console.log(`${navn} har ${navn.length} bokstaver.`);

}

for of loop: fordi jeg kan da iretrere gjennom den iterable arryaen og få hver navn og deres lengde.

**Oppgave 3**

**const navneliste = ['Geir', 'Espen', 'Ellen', 'Erik', 'Lars', 'Gunnar', 'Oda', 'Nina', 'Tine', 'Henrik', 'Agnethe'];**

Bruk den samme lista igjen, men denne gangen skal du snu rekkefølgen på alle elementene i lista. Hvilken løkke er best å bruke nå, og hvorfor? Hvis løsningen din i kode.

const reversertListe = [];

for (let i = navneliste.length - 1; i >= 0; i--) {

  reversertListe.push(navneliste[i]);

}

console.log(reversertListe);

her er det en vanlig for loop det vi går gjennom arryaen baklengs fra siste indx også går vi nedover

**Oppgave 4**

**function reduceToOddNumber(number){**

**//Den neste linjen sjekker om argumentet som blir brukt i funksjonen**

**//Faktisk er et tall.**

**if(typeof(number) == 'number'){**

**//Skriv en løkke her**

**}**

**}**

Lag en løkke (inne i funksjonen) som gjør at funksjonen over er i henhold til følgende regler;

* Tallet skal deles på 2 minst en gang.
* Hvis tallet er et partall skal tallet deles igjen
* Hvis tallet er et oddetall skal loppen avsluttes

function oddetallpartall(number) {

    if (typeof number == 'number') {

      while (number % 2 === 0) {

        number = number / 2;

      }

    }

    return number;

  }

  console.log(oddetallpartall(16));

**Oppgave 5**

Det er nå to løkker du ikke har brukt. Hvilke er det? Dobbeltsjekk om noen av de løkkene kan brukes på noen av de oppgavene du allerede har gjort og om noen av dem passer bedre.

For in og do while

Do while er brukt i noen spesielle tilfeller. Tror ingen av dem passer til de oppgavene jeg gjorde, hvis de gjør det så er jeg veldig usikker i koden min

*Hvis du var veldig veldig smart leste du hele oppgavesettet før du løste noen oppgaver. Isåfall skal du få en freebie: det er en loop som heter each…of, og den kan du egentlig bare ignorere for nå, for den trenger vi kun når vi jobber med noe som heter ‘object literals’, og det skal du lære senere.*