

## Oppgave 1

### Hva er en rekursiv funksjon?

Jeg startet med å lese gjennom disse nettsidene for å lære om rekursive funksjoner.

<https://www.freecodecamp.org/news/recursion-in-javascript/>

<https://developer.mozilla.org/en-US/docs/Glossary/Recursion>

En rekursiv funksjon er kort forklart en funksjon som kaller seg selv for å løse et problem. Den består av en “Base Case”, altså en betingelse som stopper funksjonen når den blir oppfylt, som jeg har skjønt det så er det vanligvis (om ikke alltid) en if test.

Betingelsen for å stoppe funksjonen er i denne oppgaven en if test som sjekker om tallet valgt er 4 eller 7.

Den består også av kode som skal kjøre flere ganger (I tilfellet i oppgaven er det en kode som finner et tilfeldig tall mellom 1 og 9 og deretter kaller funksjonen på seg selv i koden etter.

### Planlegging:

Jeg vet nå etter å ha lest meg opp på hva en rekursiv funksjon er at jeg må ha en betingelse som skal stoppe funksjonen og at jeg må få funksjonen til å kalle på seg selv etter koden om betingelsen ikke blir oppfylt.

Jeg tenker også at det er sikkert greit å ha en knapp for å kjøre funksjonen så du slipper å kalle den gjennom konsollen eller i selve koden.

Planen min er å bruke `math.random` for å generere et tilfeldig tall og justere denne til å bli et tall mellom 1-9. Deretter skal jeg bruke dette tilfeldige tallet i en IF test der betingelsen for å stoppe funksjonen er at tallet generert er 4 eller 7. Om denne betingelsen ikke blir møtt vil jeg at funksjonen skal kalle på seg selv så den fortsetter å generere tall helt til enten tallet 4 eller 7 blir generert.

## Gjennomføring:

Jeg setter først opp en variabel som jeg kaller rng, denne variablen bruker kodelinen

```
const rng = Math.floor((Math.random() * 9)+1)
```

Til å generere et tilfeldig tall mellom 1-9. Jeg lager deretter en variabel jeg kaller utskrift som kobler seg til numberList med hjelp av document.getElementById.

Når variablene jeg trenger er klare så lager jeg en if test som sjekker om det tilfeldige tallet er 4 eller 7, om denne betingelsen blir møtt så skrives tallet ut til numberList med tekst som forklarer at dette er tallet vi leter etter og stopper funksjonen.

Etter dette bruker jeg else som skal kjøre om tallet ikke er 4 eller 7. I denne delen skriver jeg også ut det tilfeldige tallet til numberList men jeg vil at koden skal fortsette å prøve å finne 4 eller 7 og derfor ender jeg koden med

```
return numberSelector()
```

Dette kaller funksjonen igjen og derfor vil funksjonen fortsette å generere tilfeldige tall mellom 1-9 helt frem til tallet er 4 eller 7.

## AI:

Jeg valgte å bruke GPT UIO. (<https://gpt.uio.no>)

Ai løser oppgaven ganske likt meg. Største forskjellen mellom min løsning og AI er måten den gjør utskrift til listen i html på. Den bruker document.createElement("li") for å lage listeelement også bruker den .textContent for å endre hva som skal stå på listeelementet. Den bruker også .appendChild som jeg tipper plasserer listeelementet i html listen men jeg er litt usikker.

Bortsett fra måten den fikser utskrift til listen så fungerer kodene veldig likt men det er litt forskjell i hvordan de er strukturert. Jeg bruker if test på betingelsen for å stoppe koden og else før resten av koden og AI bruker bare en if test for betingelsen og plasserer resten av koden utenfor if testen.

Screenshots fra GPT UIO under:



```
<!DOCTYPE html> <html lang="nb"> <head> <meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Rekursive funksjoner i JavaScript</title> </head> <body>
<h1>Rekursiv utskrift av tall</h1> <ul id="numberList"> <li>Tall vil skrives ut
her.</li> </ul>
```

```
<script>
/*
    Hva er en rekursiv funksjon?
    --Ditt svar her. Husk kilder--
*/

function numberSelector() {
    // Din kode her
}
</script>
```

</body> </html> Uten å bruke hjelpemdirler, planlegg og lag en rekursiv funksjon som velger et tilfeldig tall mellom 1 og 9, og skriver tallene ut i listen i eksempekkoden. Funksjonen skal slutte å kjøre rekursivt når tallet 4 eller 7 blir valgt. Javascript

 Edit

En rekursiv funksjon er en funksjon som kaller seg selv for å løse et problem som kan deles opp i mindre, enklere problemer av samme type. Dette tilnærmingen brukes ofte i programmering for å håndtere problemer som har naturlige substrukturer, som for eksempel traversering av trestrukturer eller utføre repetisjoner frem til en basistilstand er nådd.

Kilde:

- <https://developer.mozilla.org/en-US/docs/Glossary/Recursion>

### Rekursiv løsning

Her er en plan for hvordan du kan skrive en rekursiv funksjon som oppfyller kravene:

1. **Base Case:** Sjekk om det tilfeldige tallet som er valgt er enten 4 eller 7. Hvis det er en av disse, avslutt funksjonskallet.
2. **Recursive Case:** Hvis det tilfeldige tallet er mellom 1 og 9, men ikke 4 eller 7, skriver ut tallet til HTML-listen og kaller funksjonen på nytt.

Her er koden som implementerer denne logikken:

```
<!DOCTYPE html>
<html lang="nb">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Rekursive funksjoner i JavaScript</title>
</head>
<body>
    <h1>Rekursiv utskrift av tall</h1>
    <ul id="numberList">
        <!-- Tall vil skrives ut her -->
    </ul>

    <script>
        function numberSelector() {
            // Velger et tilfeldig tall mellom 1 og 9
            const randomNumber = Math.floor(Math.random() * 9) + 1;

            // Skriver ut tallet i listen
            const list = document.getElementById("numberList");
            const listItem = document.createElement("li");
            listItem.textContent = randomNumber;
            list.appendChild(listItem);

            // Sjekker basistilstand om tallet er 4 eller 7
            if (randomNumber === 4 || randomNumber === 7) {
                return; // Stopper hvis tallet er 4 eller 7
            }

            // Kaller funksjonen igjen for å fortsette prosessen
            numberSelector();
        }

        // Kjør funksjonen
        numberSelector();
    </script>
</body>
</html>
```

## Oppgave 2

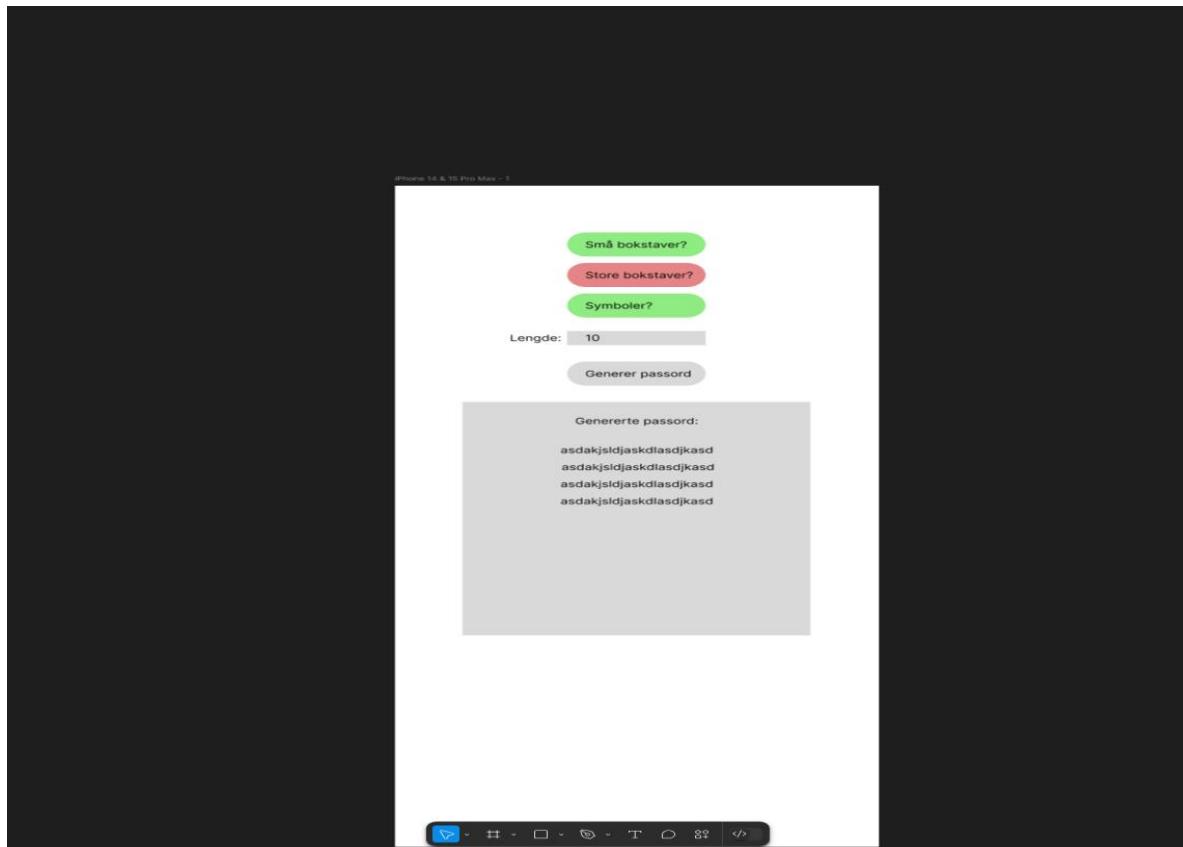
### Planlegging:

For å lage en passordgenerator der brukeren skal kunne velge lengde på passordet og samtidig gi brukeren mulighet til å skru av/på bruk av store bokstaver, små bokstaver og symboler (spesial tegn), tenker jeg å bruke knapper og et input felt til passord lengde.

Jeg vil ha en knapp for å toggle store bokstaver, en for små og en for symboler, jeg skal også ha en knapp for å generere passordet. Jeg vil at også at toggle-knappene skal gi en visuell feedback basert på om de er på eller av.

Jeg skal også bruke et input felt der brukeren kan skrive inn hvor langt passordet skal være. Til slutt vil jeg skrive ut de genererte passordene til en liste på bunnen av html dokumentet.

Jeg valgte å bruke figma til å lage en rask illustrasjon over hvordan jeg vil at passord-generatoren circa skal se ut.



Uten bruk av CSS vil det sikkert ikke være mulig at passord-generatoren ser sånn ut men illustrasjonen hjelper meg fortsatt siden nå har jeg en layout å begynne å jobbe mot.

### Gjennomføring:

Jeg begynte med å lage knappene og input feltet til passordlengden som jeg hadde planlagt på skissen jeg lagde i figma. Jeg ga dem en egen ID så jeg kan koble dem til koden min og legge til funksjonalitet senere.

Jeg valgte å bruke en uordnet liste som jeg kan skrive de ferdig genererte passordene inn i senere som listelementer.

Når jeg var fornøyd med strukturen på UI'en så lagde jeg en del variabler. Jeg lagde tre variabler som bestod av en tekst-streng med små bokstaver, store bokstaver og symboler som skal brukes i passord generatoren om brukeren velger det.

For å gjøre dette mulig lagde jeg også tre variabler med boolean verdier som skal fungere som "switcher" for brukervalgene. Jeg vil at om brukeren trykker på f eks små bokstaver knappen i html dokumentet så skal det toggle små bokstaver booleanen av og på.

Etter dette begynte jeg på funksjonen for passord-generering. Først lagde jeg en variabel som skal holde på det tilfeldig genererte passordet i en tekst streng som jeg vil skrive ut når funksjonen har kjørt ferdig.

Jeg lagde en tom array som jeg vil fylle med symbolene som brukeren har valgt å generere passordet med og en variabel som kobler seg til UL utskriften i html dokumentet.

Jeg lagde en variabel som koblet seg til passord-lengde inputtet i html dokumentet som senere viste seg å ikke fungere. Jeg valgte å google om hvordan man skal ta i mot tall verdi fra input felt i html og bruke det i javascript, jeg fant en post på stackoverflow av en bruker som lurte på det samme som meg og kom over en kommentar i kommentarfeltet som lærte meg om .valueAsNumber. Input feltet i html koden, uansett om jeg brukte type="number" ville fortsatt leses som en tekst streng i javascript og med å legge til .valueAsNumber bak koden under, løste dette problemet mitt.

```
let passordLengdeInput = document.getElementById("passLengde")
```

(Kilde <https://stackoverflow.com/questions/28695617/how-to-get-a-number-value-from-an-input-field> -Kommentar skrevet av bruker "Stack Overflow, Aug 13, 2021"

– user3989103 Feb 24, 2015 at 12:42

- 1 You can also use `.valueAsNumber` to get a number instead of a string. – Stack Overflow Aug 13, 2021 at 21:12

[Add a comment](#)

Etter dette problemet var løst valgte jeg å lage en del if-tester for å sjekke om brukeren har valgt å bruke ulike tegn (små, store, symbol) og om brukeren har skrudd på disse vil tegnene fra tegn-variablene jeg lagde tidligere bli pushet inn i arrayen passordet skal genereres fra.

Etter if testene for arrayen er gjennomført så valgte jeg å lage tre if tester til for å tvinge brukeren til å ha minst et valg av tegn (en sjekk for å se om arrayen er tom), at lengden på passordet er 1 eller over og at det faktisk står et tall i input feltet for passord lengde og ikke en bokstav.

Grunnen til at jeg ville gjøre dette var fordi jeg testet litt og fant ut at jeg ikke klarte å låse input feltet til kun å ta i mot tall på nettleseren min. Jeg googlet litt om hvordan sjekke om inputten er et tall eller ikke

(Her brukte jeg isNaN – is Not a Number, jeg hadde ikke kjennskap til isNaN fra før, jeg googlet litt og fant en god forklaring på hvordan denne fungerte (Kilde:

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/isNaN](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/isNaN)

Om disse testene finner en feil som f eks at lengden på passordet så har brukeren valgt bokstaven A så vil funksjonen returnere en alert som informerer brukeren om feilen og stoppe funksjonen der.

Om alt er i orden går det videre til neste steg der jeg valgte å bruke en for-løkke for å hente ut tilfeldige tegn fra arrayen som er fylt med tegn som brukeren har valgt å bruke til passord generasjonen. For å gjøre dette lagde jeg en variabel som er et tilfeldig generert tall mellom 0 og lengden på arrayen med tegnene med hjelp av

```
Math.floor(Math.random() * array.length)
```

Jeg bruker tallet til å hente ut tegnet med indeks nummeret generert av math random fra arrayen og skrive det i variablen som skal bli det ferdige genererte passordet:

```
password += array[rng]
```

På slutten av funksjonen skriver jeg ut det ferdige passordet til listen av passord i html dokumentet og ender funksjonen med å returnere passordet:

```
passListe.innerHTML += `<li>${password}</li>`  
return password
```

Jeg lagde tre mindre funksjoner som skal gjøre at toggle knappene jeg lagde helt i begynnelsen av oppgaven fungerer.

De tre mindre funksjonene inneholder alle først og fremst en kodelinje som bytter boolean verdien mellom true og false når du kjører funksjonene og deretter endre teksten på knappene ut i fra verdien på booleanen med hjelp av en if test (Er f eks booleanen på små bokstaver true vil knappen i html dokumentet endres til "Små bokstaver? JA" og motsatt blir "Små bokstaver? NEI").

Med å gjøre dette vil brukeren aldri måtte gjette om bruk små bokstaver er skrudd på eller av.

Helt til slutt la jeg til event listeners for å sjekke om knappene blir trykket på av brukeren og gjøre at knappene kjører hver sin funksjon.

## Oppgave 2 med AI

Ai har gjort oppgaven ganske forskjellig fra måten jeg har gjort det. Jeg ser med en gang at jeg kanskje burde ha gitt brukeren et valg om å bruke bare tall i passordet sitt men jeg har tall og spesialtegn som et valg.

Jeg har brukt knapper for valg av å inkludere ulike tegn, AI har brukt checkbox.

Jeg ser at AI ikke har brukt .valueAsNumber for å lese passordlengde inputten sin som et tall men

AI har ikke brukt noen if test for å sjekke at passordlengden som brukeren har valgt er et tall. I firefox kan jeg skrive inn for eksempel passordlengde: a og det gir ingen feilmelding. AI har derimot en mindre irriterende måte å gi "feilmelding" på til brukeren. Om du ikke har valgt å inkludere noen tegn vil den bare skrive ut "velg minst en type tegn!" der det genererte passordet blir skrevet ut mens måten jeg gjorde det på var å

bruke en alert. Alert er en veldig irriterende popup har jeg kommet fram til nå etter å ha sammenlignet med AI sin løsning.

Selv passordgenereringen er veldig lik, både jeg og AI har brukt en for løkke som kjører flere ganger basert på lengde på passord valgt av bruker. Jeg samler tegnene som brukeren vil ha i et array men AI samler dem i en tekst-streng, vi begge bruker et tilfeldig generert nummer mellom 0 og lengden på arrayen/tekststrengen til å hente ut et tilfeldig tegn og legger disse tegnene sammen i en tom tekst streng som blir det ferdig-genererte passordet.

Noe jeg ikke visste at det var mulig her var når AI brukte kodelinjen

```
password += characterPool[randomIndex];
```

For å hente et tilfeldig tegn fra characterPool. Character pool er en tekststreng i AI sin løsning og jeg trodde du bare kunne bruke Index på arrayer på den måten. Derfor er måten jeg og AI brukte array og tekststreng veldig lik siden det gjør egentlig det samme.

AI har løst oppgaven med å bruke en god del mindre kode enn meg.

Det jeg kunne spart meg mest kodelinjer på er funksjonene jeg lagde for å “toggle” brukervalgene for ulike tegn brukt i passordet.

Dette er en spennende kodelinje brukt av AI som jeg tror henter tallverdien til passordlengde inputten. Dette er sikkert grunnen til at AI ikke trengte å bruke .valueAsNumber for å hente tallverdi fra html inputten men jeg måtte bruke det.

```
const length = parseInt(document.getElementById('length').value);
```

Screenshots fra GPT UIO under:

⚡ Model: GPT-4 Omni

```
<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Passordgenerator</title> </head> <body> <h1>Passordgenerator</h1> <!-- Felter for brukervalg her -->

<p>Ditt passord: <span id="generatedPassword"></span></p>

<script>
const lowercase = "abcdefghijklmnopqrstuvwxyz";
const uppercase = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
const numbers = "0123456789";
const symbols = "!@#$%^&*()_+-=[{}];:,.<>?/~";
```

```
function passwordgenerator(length, includeLowercase, includeUppercase, includeNumbers, includeSymbols) {
    /* Din kode her */
}
```

```
</body> </html> I denne oppgaven skal du lage en passordgenerator. Brukeren skal kunne velge lengde på passordet som skal generes, om det skal inneholde bare store eller små bokstaver (eller begge deler), og om det skal kunne inneholde spesialtegn eller ikke. Javascript
```

 Edit

To create a password generator with the specified options, you'll need to implement the logic in the passwordgenerator function within your JavaScript code. This function should take parameters that determine the password length, inclusion of lowercase letters, uppercase letters, numbers, and symbols. Based on these conditions, the function will generate a random password.

Here's how you can implement the password generator:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Passordgenerator</title>
</head>
<body>
```

```
<h1>Passordgenerator</h1>

<label for="length">Passordlengde:</label>
<input type="number" id="length" value="12" min="1" max="50"><br><br>

<input type="checkbox" id="lowercase" checked>
<label for="lowercase">Inkluder små bokstaver</label><br>

<input type="checkbox" id="uppercase" checked>
<label for="uppercase">Inkluder store bokstaver</label><br>

<input type="checkbox" id="numbers" checked>
<label for="numbers">Inkluder tall</label><br>

<input type="checkbox" id="symbols" checked>
<label for="symbols">Inkluder spesialtegn</label><br><br>

<button onclick="generatePassword()">Generer Passord</button>

<p>Ditt passord: <span id="generatedPassword"></span></p>

<script>
const lowercase = "abcdefghijklmnopqrstuvwxyz";
const uppercase = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
const numbers = "0123456789";
const symbols = "!@#$%^&*()_+-=[{}];:,.<>?/~";
```

```
function passwordgenerator(length, includeLowercase, includeUppercase, includeNumbers, includeSymbols) {
    let characterPool = "";
    if (includeLowercase) {
        characterPool += lowercase;
    }
    if (includeUppercase) {
        characterPool += uppercase;
    }
    if (includeNumbers) {
        characterPool += numbers;
    }
    if (includeSymbols) {
        characterPool += symbols;
    }

    if (characterPool.length === 0) {
        return "Vælg minst en type tegn!";
    }

    let password = "";
    for (let i = 0; i < length; i++) {
        const randomIndex = Math.floor(Math.random() * characterPool.length);
        password += characterPool[randomIndex];
    }
    return password;
}
```

```
function generatePassword() {
    const length = parseInt(document.getElementById('length').value);
    const includeLowercase = document.getElementById('lowercase').checked;
    const includeUppercase = document.getElementById('uppercase').checked;
    const includeNumbers = document.getElementById('numbers').checked;
    const includeSymbols = document.getElementById('symbols').checked;

    const password = passwordgenerator(length, includeLowercase, includeUppercase, i
    document.getElementById('generatedPassword').innerText = password;
}
</script>
</body>
</html>
```

### Explanation:

- **HTML Elements:** Added input fields for password length and checkboxes to select which types of characters to include in the password.
- **JavaScript Logic:** The passwordgenerator function checks which types of characters are to be included and constructs a characterPool string from which it randomly selects characters to build the password.
- **generatePassword Function:** This is triggered by clicking the "Generer Passord" button, and it gets the user's selection from the input fields, calls passwordgenerator with these options, and displays the generated password.

 Regenerate  Copy

Kilder:

Value as number post på Stack Overflow:

<https://stackoverflow.com/questions/28695617/how-to-get-a-number-value-from-an-input-field>

Is not a number:

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/isNaN](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/isNaN)

Rekursive funksjoner:

<https://www.freecodecamp.org/news/recursion-in-javascript/>

<https://developer.mozilla.org/en-US/docs/Glossary/Recursion/>

AI: GPT UIO

<https://gpt.uio.no>