

Rapport- Ak3

Oppgave 1

1.3 AI løsningen og refleksjon

```
/**
 * numberSelector() er en rekursiv funksjon som velger et tilfeldig tall
 * mellom 1 og 9 og legger det til i en HTML-liste.
 * Den rekursive kallet stopper (Basistilfellet) når tallet 4 eller 7 er valgt.
 */
function numberSelector() {
    // 1. Generer et tilfeldig tall mellom 1 og 9 (inkludert)
    const randomNumber = Math.floor(Math.random() * 9) + 1;

    // 2. Finn listen i HTML-dokumentet
    const numberList = document.getElementById("numberList");

    // Fjern eventuell placeholder-tekst fra starten
    if (numberList.children.length === 1 &&
numberList.children[0].textContent.includes('skrives ut her')) {
        numberList.innerHTML = "";
    }

    // 3. Opprett et nytt listeelement (<li>) og legg til tallet
    const listItem = document.createElement("li");
    listItem.textContent = `Valgt tall: ${randomNumber}`;
    numberList.appendChild(listItem);

    // 4. Sjekk Basistilfellet (Stopping Condition)
    if (randomNumber === 4 || randomNumber === 7) {
        // Legg til en markering for å vise at rekursjonen er stoppet
        listItem.textContent += " – Stopp: Basistilfellet nådd!";
    }
}
```

```
listItem.style.fontWeight = "bold";

// Stopp videre rekursivt kall ved å returnere
return;
} else {
    // 5. Rekursivt Kall: Fortsett prosessen
    numberSelector();
}
}

// Start funksjonen
numberSelector();
```

Refleksjon: Jeg ser at det ikke er så stor forskjell på min kode fra AI sin kode. Det er ganske minimalt. Det er bare små variasjoner i skrivemåten for teksten og hvordan *if-blokken* er satt opp.

Oppgave 2

2.1 Disse feltene må brukeren arbeide med:

1. passordlengde: brukeren må velge hvor langt passordet skal være
2. Små bokstaver: inkludere a-z
3. Store bokstaver: inkludere A-Z
4. Tall: inkludere 0-9
5. Symboler: inkludere spesialtegn
6. Generer passord: knapp for å starte funksjonen

Grensesnittelement	Type HTML-felt	Formål/Funksjon
Passordlengde	Input (Number)	Lar brukeren velge lengden på passordet (f.eks., 4 til 30).
Små bokstaver (a-z)	Checkbox	Inkluderer lowercase strengen i det mulige tegnsettet.
Store bokstaver (A-Z)	Checkbox	Inkluderer uppercase strengen i det mulige tegnsettet.
Tall (0-9)	Checkbox	Inkluderer numbers strengen i det mulige tegnsettet.
Symboler (!@#\$...)	Checkbox	Inkluderer symbols strengen i det mulige tegnsettet.
Generer passord	Button	Kjører passwordgenerator() funksjonen og viser resultatet.

2.4 AI løsningen og refleksjon

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Passordgenerator (KI-løsning)</title>
  <style>
    body { font-family: Arial, sans-serif; }
    label { display: block; margin-bottom: 5px; }
    input[type="number"] { width: 60px; }
    #generatedPassword { font-weight: bold; color: darkgreen; font-size: 1.1em; }
    .controls { margin-bottom: 20px; border: 1px solid #ccc; padding: 15px; border-radius:
5px; }
  </style>
</head>
<body>
  <h1>Passordgenerator</h1>

```

```

<div class="controls">
  <h3>Velg passordkriterier</h3>
  <label for="length">Lengde på passordet (min 4, maks 30):</label>
  <input type="number" id="length" min="4" max="30" value="12"><br><br>

  <label>
    <input type="checkbox" id="includeLowercase" checked> Små bokstaver (a-z)
  </label>
  <label>
    <input type="checkbox" id="includeUppercase" checked> Store bokstaver (A-Z)
  </label>
  <label>
    <input type="checkbox" id="includeNumbers" checked> Tall (0-9)
  </label>
  <label>
    <input type="checkbox" id="includeSymbols"> Spesialtegn (!@#$%...)
  </label>
</div>

<button id="generateBtn">Generer passord</button>

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

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

  /**
   * Hovedfunksjonen for å generere et tilfeldig passord basert på gitte parametere.
   * @param {number} length - Ønsket lengde på passordet.

```

```

* @param {boolean} includeLowercase - Inkluder små bokstaver.
* @param {boolean} includeUppercase - Inkluder store bokstaver.
* @param {boolean} includeNumbers - Inkluder tall.
* @param {boolean} includeSymbols - Inkluder spesialtegn.
* @returns {string} Det genererte passordet eller en feilmelding.
*/

function passwordgenerator(length, includeLowercase, includeUppercase,
includeNumbers, includeSymbols) {

    let availableCharacters = "";
    let generatedPassword = "";

    // Bygger opp strengen med tilgjengelige tegn basert på brukerens valg
    if (includeLowercase) availableCharacters += lowercase;
    if (includeUppercase) availableCharacters += uppercase;
    if (includeNumbers) availableCharacters += numbers;
    if (includeSymbols) availableCharacters += symbols;

    // Feilhåndtering: Sørger for at minst én type tegn er valgt
    if (availableCharacters.length === 0) {
        return "Feil: Vennligst velg minst én tegnkategori for å generere passord.";
    }

    // Juster lengden hvis den er utenfor rimelig grense (for sikkerhets skyld)
    const passwordLength = Math.min(Math.max(length, 4), 30);

    // Genererer passordet ved å velge tilfeldige tegn fra den tilgjengelige strengen
    for (let i = 0; i < passwordLength; i++) {
        const randomIndex = Math.floor(Math.random() * availableCharacters.length);
        generatedPassword += availableCharacters[randomIndex];
    }

    return generatedPassword;
}

```

```
// --- Event Listener og Henting av Brukerinput ---

document.getElementById("generateBtn").addEventListener("click", () => {
  // Henter verdier fra input-feltene
  const lengthInput = document.getElementById("length");
  const length = parseInt(lengthInput.value, 10);

  const includeLowercase = document.getElementById("includeLowercase").checked;
  const includeUppercase = document.getElementById("includeUppercase").checked;
  const includeNumbers = document.getElementById("includeNumbers").checked;
  const includeSymbols = document.getElementById("includeSymbols").checked;

  // Kaller passordgeneratoren
  const password = passwordgenerator(length, includeLowercase, includeUppercase,
includeNumbers, includeSymbols);

  // Skriver ut resultatet til brukeren
  document.getElementById("generatedPassword").textContent = password;
});

</script>
</body>
</html>
```

Refleksjon: Begge kodene er funksjonelt korrekte og bruker samme grunnleggende logikk. De samler alle tillatte tegn i stor streng og velger deretter tilfeldig fra denne strengen. Den viktigste forskjellen mellom de to kodene er at min kode genererer et passord med den nøyaktige lengden brukeren oppgir noe som da kan føre til treghet eller feil. Mens AI koden sikrer at den aldri genererer et passord kortere enn 4 eller lengre enn 30 tegn selv om brukeren skriver inn en ekstrem verdi.

Læring: Jeg lærer at det er viktig å legge til en validering av input i koden som AI gjorde. Dette forhindrer feil dersom brukeren omgår HTML-validering.

Konklusjon: Jeg anser AI løsningen som den beste koden siden den har bedre struktur. Den var mer profesjonell og mer pålitelig.

Referanseliste:

Google. (2025). *Svar på spørsmål om hjelp til å besvare AI oppgavene.*

<https://gemini.google.com/share/35a47f423864>

W3Schools. (u.å.). *JavaScripts functions.*

https://www.w3schools.com/js/js_functions.asp

Webtricks. (u.å.). *Bruker input.*

<https://lms.webtricks.blog/kurs/innforing-i-programmering/8-brukerinput/bruke-verdier-fra-skjemafelter#h17dd1e933814>

Nätt, Tom Heine: rekursjon - programmering i Store norske leksikon på snl.no.

Hentet 11. november 2025 fra https://snl.no/rekursjon_-_programmering