## Opdracht

Je levert de Memory game waarmee je eerder geoefend hebt op. Als je dit nog niet gedaan hebt, moet je dat alsnog doen om verder te kunnen. Voor de volledigheid hier de beide opdrachten:

Opdracht 1

* + Je gaat een begin maken met een compleet spel in JavaScript, een memory game! Om te beginnen, ga je de kaartjes die in een array geplaatst zijn zichtbaar maken door ze met de geleerde DOM methods toe te voegen aan de DOM. Je gaat onder andere gebruikmaken van:

Afbeelding met tekst, ontvangst, Lettertype, wit

Door AI gegenereerde inhoud is mogelijk onjuist.

* + Open de folder memory\_start uit de map met oefenbestanden in VSC (zie de Mediatheek onder Bestanden oefeningen SA).
  + Bestudeer de HTML en de CSS zodat je een beeld krijgt bij wat je aangeleverd hebt gekregen.
  + Open memory.js en bekijk de commentaarcode. Probeer eerst een goed beeld te krijgen van wat de bedoeling is. Voer de hierin beschreven stappen vervolgens nauwkeurig uit. Lees ze goed, en herlees ze desnoods nog een keer zodat je precies weet wat er van je verwacht wordt. De programmaregel van het array myCardSet krijg je alvast.
  + Bestudeer eventueel de desbetreffende materie nogmaals vóór je een regel code programmeert.
  + Als het goed is uitgevoerd, zie je in de browser de kaartjes allemaal verschijnen. Als je op een kaartje klikt, verschijnt de naam van het dier in de console.
  + Gebruik de technieken die je eerder geleerd hebt om te debuggen. In de map memory-Result vind je het resultaat, maar gebruik deze alleen om het resultaat te controleren of als je er helemaal niet uit komt.
  + Als extra uitdaging kun je de loop met de volgende statements uitvoeren:

Afbeelding met tekst, Lettertype, ontvangst, wit

Door AI gegenereerde inhoud is mogelijk onjuist.

Opdracht 2

1. Maak in uw script van de memory game (van de vorige oefening) een nieuwe class 'Card' aan met als properties card1, card2, set en sound. Op deze manier zou je dit object ook kunnen gebruiken voor een spel waarin je twee verschillende kaartjes van één set moet matchen en zou je een geluid kunnen afspelen, bijvoorbeeld een dierengeluid bij het spel met dieren. Alleen argument card1 is verplicht, de rest is optioneel. Je hebt nog geen kennisgemaakt met optionele (default-)argumenten, meer informatie hierover vind je [hier](https://link.loi.nl/r/loi-101273).
2. Zorg ervoor dat card2, set en sound als defaultwaarde dezelfde waarde krijgen als image1. Dit kun je gewoon tussen de haakjes van de class doen! Dit heeft als resultaat dat als je alleen de naam van card1 invoert, voor zowel card2 als sound naar dezelfde benaming wordt gezocht als voor card1. Mochten de geluidsbestanden echter anders heten, dan kunnen de afwijkende benamingen ingevoerd worden.
3. Hernoem het array myCardSet naar myCardArray en maak een nieuw array myCardSet met deze Card-objecten. Dit kun je op twee manieren doen, handmatig of met een loop. Slaag je er in om met de *map()* method een nieuw array met Card-objecten te bouwen vanuit myCardArray? Je kunt met jouw eigen oefening verder gaan, of de map memory\_2.5/start als uitgangspunt nemen. Het eindresultaat vind je in de map memory\_2.5/result.

Als het goed is, heb je in ieder geval een complete set met kaartjes op het scherm staan die je aan kunt klikken waarna een melding in de console verschijnt.

1. Om te beginnen, de kaarten moeten verdubbeld worden, anders is er geen sprake van een memory game, de speler moet immers twee gelijke kaarten vinden. De eerste stap is het verdubbelen van de array met één van de array methods. Welke je moet gebruiken? Dat moet je zelf uitzoeken.
2. Vervolgens moet de resulterende set kaarten random over het array verdeeld worden. De kaarten van een memory game zijn immers eerst geschud en worden dan pas neergelegd. Dit schudden kan eenvoudig met JavaScript. Er is zelfs een optimale 'shuffle' bedacht, de [Fisher-Yates shuffle](https://link.loi.nl/r/loi-101274).
3. Deze vind je [hier](https://link.loi.nl/r/loi-101275" \t "_blank) uitgewerkt in JavaScript, op een pagina geschreven door niemand minder dan Mike Bostock, het brein achter het D3JS framework.
4. Implementeer deze shuffle als functie in jouw script. Gebruik deze vervolgens om myCardArray te 'schudden' vóór je dit omzet naar een array met Card Objecten.
5. Het resultaat is een pagina met een complete set (dubbele) kaartjes die na een page refresh compleet anders neergelegd is.

**De beoordelingscriteria:**

* + In de code is een class 'Card' gemaakt.
  + Deze class heeft de properties card1, card2, set en sound.
  + Alleen de property card1 moet tijdens instantiëren ingevoerd worden, card2, set en sound krijgen card1 als defaultwaarde.
  + De array myCardSet wordt verdubbeld m.b.v. de array method concat() of een alternatieve oplossing.
  + De functie shuffle() (de Fisher-Yates shuffle) is geïmplementeerd en werkt.
  + Met de map() method wordt de myDblCardArray naar een myCardSet getransformeerd, bestaande uit Card-objecten.