

# Recap 1 - Texas Hold'em

## Doel

Het doel van deze opdracht is om vijf willekeurige kaarten te selecteren en te controleren wat de hoogste combinatie is binnen de spelregels van het spel Texas Hold'em poker.



WOW, YOU HAVE A ROYAL FLUSH

## De spelregels

Bij Texas Hold'em heeft elke speler vijf kaarten. Op basis van deze kaarten zijn er 12 mogelijke "winning hand combinations". Deze zijn van laag naar hoog:

- **A pair:** een speler heeft 2 kaarten met hetzelfde nummer. Bijvoorbeeld: 2 keer een 7 of 2 keer een King
- **A double pair:** een speler heeft 2 keer 2 kaarten met hetzelfde nummer. Bijvoorbeeld: 2 keer een 7 en 2 keer een King
- **Three of a kind:** een speler heeft 3 kaarten met hetzelfde nummer
- **Straight:** een speler heeft 5 kaarten die elkaar opvolgen. Bijvoorbeeld: 4, 5, 6, 7, 8 of 9, 10, Joker, Queen, King
- **Flush:** een speler heeft 5 kaarten van dezelfde soort. Bijvoorbeeld: 5 keer harten of 5 keer ruiten
- **Full house:** Een speler heeft een keer 3 kaarten met hetzelfde nummer en een keer 2 kaarten met hetzelfde nummer. Bijvoorbeeld: 3 keer een Joker en 2 keer een 7.
- **Four of a kind:** een speler heeft 4 kaarten met hetzelfde nummer
- **Straight flush:** is een combinatie van straight en flush. Dus 5 kaarten op een rij van dezelfde soort.
- **Royal flush:** is een straight flush met als extra voorwaarde dat het om de hoogst mogelijke combinatie gaat. De kaarten moeten dus 9, 10, Joker, Queen, King, Ace zijn van dezelfde soort.

Je kan de mogelijke combinaties ook [hier](#) nalezen.

# Stappenplan

## Stap 1: 5 unieke kaarten selecteren

De code om 5 unieke kaarten te selecteren is gegeven. Zoals in oefening6-cards uit hoofdstuk 7 aan bod kwam wordt een kaart geselecteerd door een element uit de array `cardNumbers` te combineren met een element uit de array `cardSuits`.

De functie `pickRandomCards()` implementeert deze functionaliteit. Bekijk eens aandachtig de code van deze functie en bijbehorende functies.

OPMERKING: voor deze oefening staat er wel een koppelteken tussen het nummer van de kaart en de suit.

OPMERKING 2: voor deze oefening werden de kaarten gesorteerd van klein naar groot via de functie `sortCardsByNumer()`. Dit is nodig voor latere controles.

## Stap 2: de kaarten op het scherm tonen

De code om de kaarten op het scherm te tonen is ook gegeven. Deze is ook gelijkaardig aan de oplossing van oefening6-cards uit hoofdstuk 7.

De functie `showCards()` implementeert deze functionaliteit. Bekijk eens aandacht de code van deze functie en bijbehorende functies.

## Stap 3: controleren op winnende combinaties

Als je de winnende combinaties goed analyseert moet je in feite maar drie zaken controleren:

- Hoeveel kaarten met hetzelfde nummer zijn er aanwezig?
- Zijn alle kaarten van hetzelfde type (suit)?
- Volgen alle kaarten elkaar op?

Als je voor elk van deze zaken een functie schrijft is het mogelijk om alle twaalf de combinaties te controleren.

TIP: gebruik een hard-coded array om te controleren op winnende combinaties. Een combinatie als royal flush komt maar zelden voor. Je verliest veel tijd met voortdurend te refreshen.

```
const pickedCards = [`10-H`, `J-H`, `Q-H`, `K-H`, `A-H`]
```

### Stap 3.1 controleren hoeveel kaarten hetzelfde nummer hebben

Schrijf een functie om te controleren hoeveel kaarten er een identiek nummer hebben. Deze functie krijgt het gewenste aantal binnen als parameter. Op die manier kan je ze gebruiken om een pair, three of kind en four of a kind te controleren.

Aangezien je ook een double pair moet controleren zal de functie een array teruggeven. Een double pair zal dan 2 elementen in de array bevatten.

Voorbeelden:

Als er gecontroleerd wordt op 2 gelijke kaarten:

- Input: ['5-H', '6-H', '7-C', '9-D', 'J-S'] → resulteert in output []
- Input: ['5-H', '6-H', '6-C', '9-D', 'J-S'] → resulteert in output ['6']
- Input: ['5-H', '6-H', '6-C', '9-D', '9-S'] → resulteert in output ['6', '9']

Als er gecontroleerd wordt op 3 gelijke kaarten

- Input: ['5-H', '6-H', '6-C', '9-D', 'J-S'] → resulteert in output []
- Input: ['5-H', '6-H', '6-C', '6-D', 'J-S'] → resulteert in output ['6']

Als er gecontroleerd wordt op 4 gelijke kaarten

- Input: ['5-H', '6-H', '6-C', '6-D', 'J-S'] → resulteert in output []
- Input: ['6-S', '6-H', '6-C', '6-D', 'J-S'] → resulteert in output ['6']

TIP: er is een functie `getCardNumber()` aanwezig om het nummer van de kaart te filteren uit een kaart.

### Stap 3.2 controleren of alle kaarten van hetzelfde type (suit) zijn

Schrijf een functie die een boolean zal teruggeven. True indien alle kaarten van dezelfde suit zijn, anders false. Binnen de functie haal je de suit van de eerste kaart op. Alle andere kaarten moeten dezelfde suit hebben om true op te leveren.

TIP: er is een functie `getCardSuit()` aanwezig om de suit van de kaart te filteren uit een kaart.

TIP 2: het gebruik van de functie `every()` kan je veel werk besparen.

### Stap 3.3 controleren of alle kaarten elkaar opvolgen

Schrijf een functie die een boolean zal teruggeven. True indien alle kaarten elkaar opvolgen, anders false. Gebruik ook een parameter om te controleren of het om de hoogst mogelijke opeenvolging gaat (10, Joker, Queen, King, Ace). Dit is nodig om de royal flush te controleren.

De kaarten staan sowieso gesorteerd op kaartnummer (zie opmerking 2 bij stap 1). Je bent dus zeker dat de eerste kaart het kleinste getal heeft.

De beste manier om dit aan te pakken is om de positie van de kleinste kaart in de array `cardNumbers` op te zoeken en vanaf daar de eerste vijf elementen te selecteren. Als dit matcht met de nummers van jouw kaarten heb je vijf opeenvolgende kaarten, anders niet.

TIP: er is een functie `getCardNumber()` aanwezig om het nummer van de kaart te filteren uit een kaart.

### Stap 3.4 check alle combinaties

Schrijf een functie die alle combinaties zal controleren op basis van bovenstaande functies. Deze functie geeft een string terug met de melding welk een combinatie er gevonden werd. Als er geen enkele combinatie gevonden werd return je ook deze melding.

De combinaties dienen wel van hoog naar laag gecontroleerd te worden. Een royal flush is namelijk ook een flush en een straight. De functie mag in dat geval de tekst `Wow, you have a royal flush` teruggeven.

## Stap 4: toon de melding

Schrijf tot slot een functie die het `<p>` element met de class `message` opvult met de hoogst gevonden combinatie.