Hauptprogramm

- namespace
- document.addEventListener("DOMContentLoaded") → "init"

---

function init(): void

- Variable anlegen für Rückgabe-wert der prompt-Funktion
- let cards = promptCards();
- Funktionsaufruf createGame(cards)

---

function promptCards()

- Variablen anlegen
- für Text
- let cards: string = prompt("TEXT")
- für Eingabe
- let cardsNum: number = parseInt(cards)
- ◇ (Eingabe prüfen)
  - if (keine Zahl oder außerhalb des Rahmens) → Aufruf init()
  - else → Rückgabe cardsNum

↑ function createGame(-cards: number): void

- document.addEventListener("keydown", spaceDown) → keydown → spaceDown()
- Variablen anlegen für main + Element erstellen
- let main: HTMLElement = document.create ("main")
- Element in DOM einsetzen / Platz zuweisen
- document.body.appendChild(main)
- Funktionsaufruf createGameArea(-cards, main);
- Funktionsaufruf createFooter();
- ©→ createStack(-main) ↑
- createDiscard(-main, cardContentOpen) ↑
- createHand(-main, cardContentHand) ↑
- createSortButton(-main) ⊙

↑ function createGameArea (cards: number, main: HTMLElement): void

- großes Array mit allen 108 Karten des UNO Decks; src als string
- let contentAll: string[]
- Mischen simulieren
- ↑ Funktionsaufruf shuffleArray(cardContentAll)
- Array erstellen um Karten den 3 verschiedenen Positionen zuordnen zu können, ohne das ursprüngliche Array kaputt zu machen
- aus dem Deck angegebene Menge an Karten auf die Hand "kopieren"
- let cardContentHand: string[] = cardContentAll.slice(0, -cards)
- aus dem Deck eine Karte "kopieren" + "anfeuern"
- Ⓑ

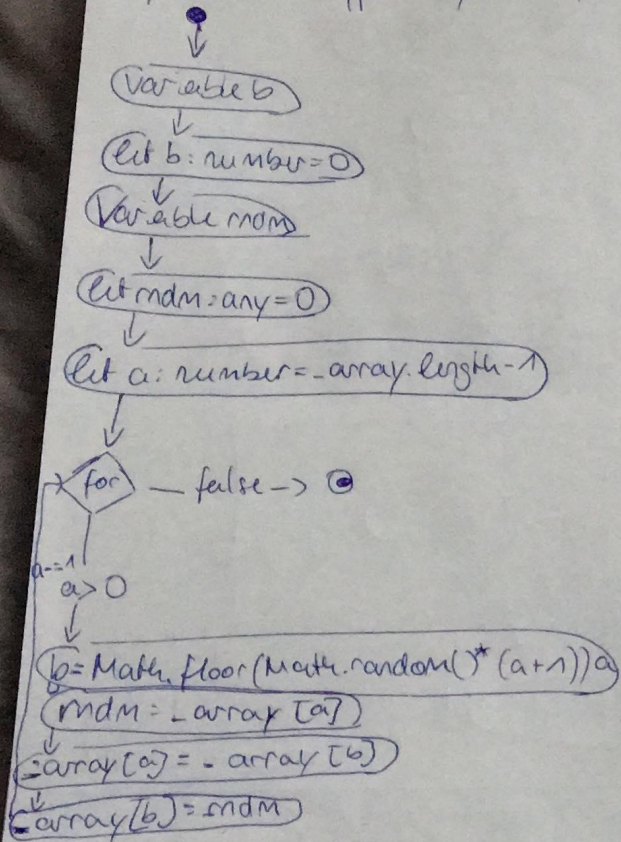---

- Ⓑ→ let cardContentOpen: string[] = cardContentAll.slice(-cards+1)
- aus dem Deck restliche Karten in den Aufnahmestapel "kopieren"
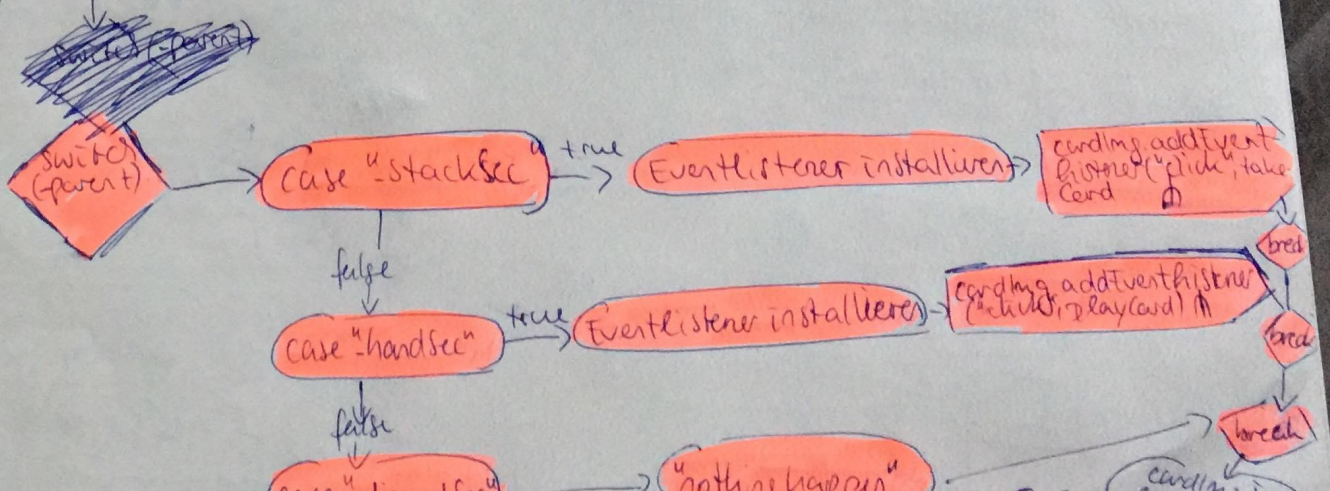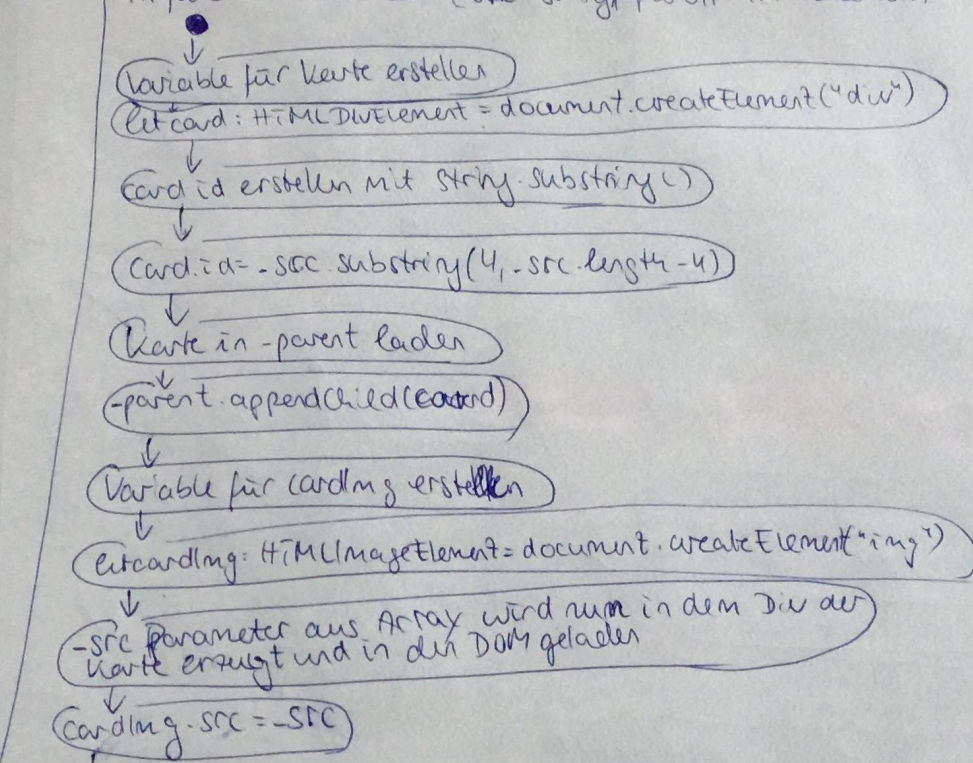- let cardContentStack: string[] = cardContent.splice(-cards+1, cardContentAll.length);
- 3 verschiedene sections in main für die 3 Positionen der Karten erstellen
- Ⓒ

function shuffleArray (-array: string[]): void

- Variable b
- let b: number = 0
- Variable mdm
- let mdm: any = 0
- let a: number = -array.length-1
- for — false -> ⊙
  - a-=1
  - a > 0
  - b = Math.floor(Math.random()*(a+1))a
  - mdm = -array[a]
  - -array[a] = -array[b]
  - -array[b] = mdm

function createCard (-src: string, -parent: HTMLElement): void

- Variable für Karte erstellen
- let card: HTMLDivElement = document.createElement("div")
- Card id erstellen mit string.substring()
- card.id = -src.substring(4, -src.length-4)
- Karte in -parent laden
- -parent.appendChild(card)
- Variable für cardImg erstellen
- let cardImg: HTMLImageElement = document.createElement("img")
- -src parameter aus Array wird nun in dem Div der Karte erzeugt und in den DOM geladen
- cardImg.src = -src
- switch (-parent)

switch (-parent) → case "-stackSec" →true→ Eventlistener installieren → cardImg.addEvent listener("click", take Card) → break

false ↓

case "-handSec" →true→ Eventlistener installieren → cardImg.addEventlistener ("click", playCard) → break

false ↓

case "-..." → "nothing happens" → cardImg...

function createFooter(): void

- Variable für Footer, diesen erzeugen und in DOM platzieren
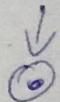- Variable für <p>, diesen erzeugen und in footer platzieren
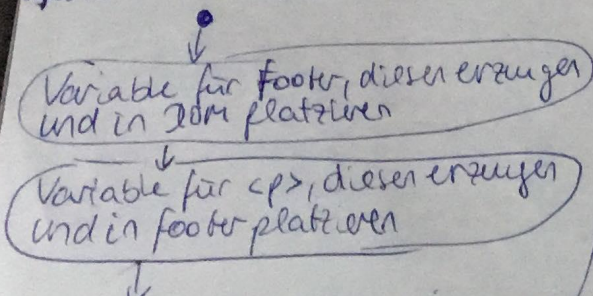
function createStack (_main : HTMLElement): void

- let stackSec : HTMLElement = document.createElement ("section")
- id zuweisen
- in DOM laden
- Funktionsaufruf createCard ("src Backside", stackSec)
- (end)

function createHand (_main : HTMLElement, _cardContentHand : string[]): void

- let handSec : HTMLElement = document.createElement ("section")
- id zuweisen
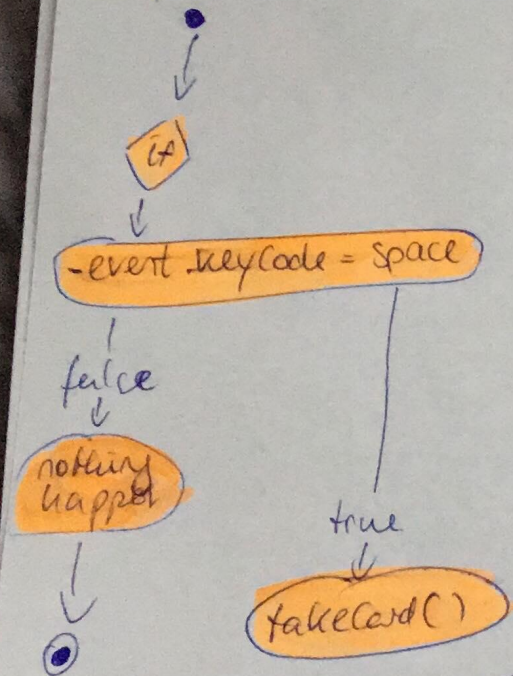- in DOM laden
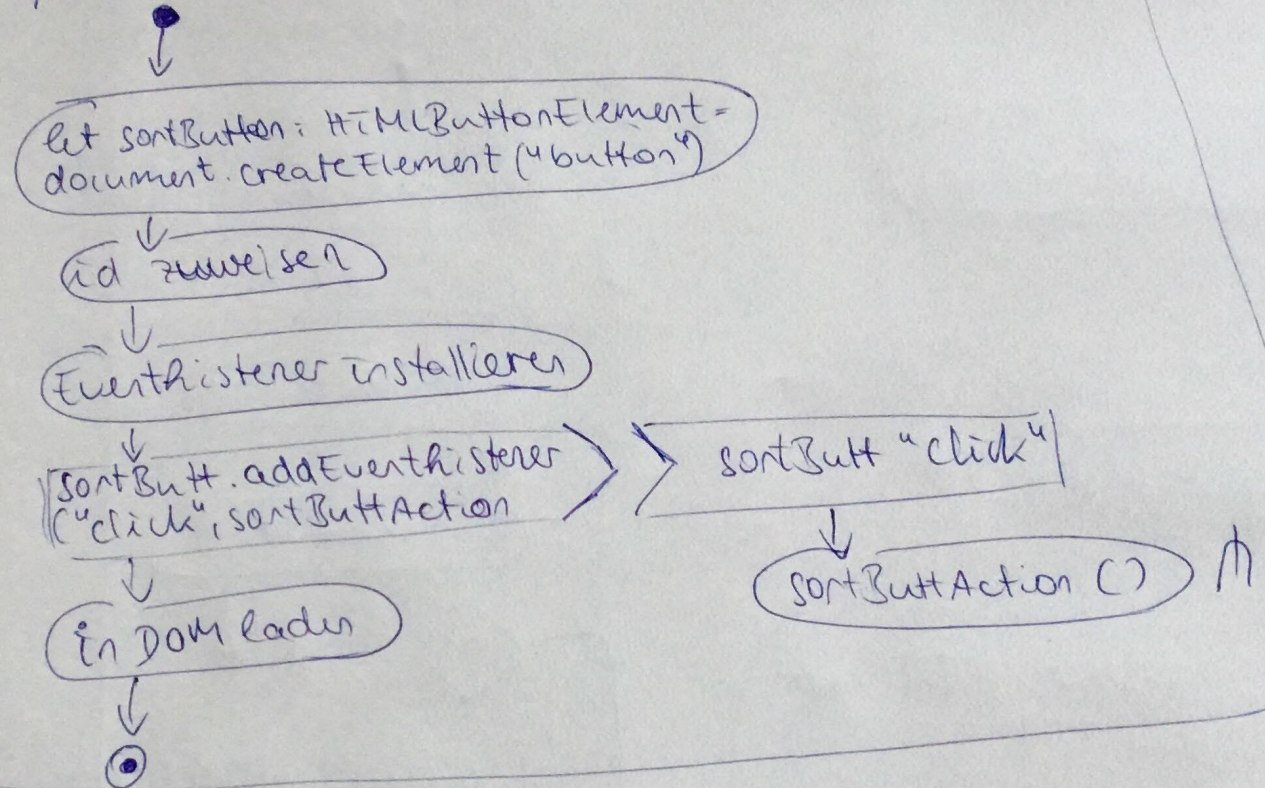- let i : number = 0
- for → false → (end)
  - i < cardContentHand.length
  - true → Funktionsaufruf createCard (cardContentHand [i], handSec)
  - i++

function createDiscard (_main: HTMLElement, _cardContentOpen: string[]): void

- let discardSec : HTMLElement = document.createElement ("section")
- id zuweisen
- in DOM laden
- Funktionsaufruf createCard ("_cardContentOpen [0], discardSec")

↑ function spaceDown ():void

↑ function createSortButt (-Main: HTMLElement):void

(f)

-event.keyCode = space

false

nothing happen

true

fakeCard()

let sortButton: HTMLButtonElement = document.createElement ("button")

id zuweisen

Eventlistener installieren

sortButt.addEventlistener ("click", sortButtAction)

sortButt "click"

in DOM laden

sortButtAction ()

↑