```
let pcHand: HTHLDivElement = document. query Selector ("# pcHand"):
let player Hand: HTMLDIVElement = elocument query Selector ("#player Hand"):
const possible colors: string[] = ["yellow", "blue", "green", "red"];
const possible Numbers: number []=[1,2,3,4,5,6,7,8,9];
Let random color Value: string;
let random Number Value: string;
 Let current coird: HTHLS pan Element = document, query selector ("#current Card");
 Let deck: HTMLS pan Element = document . query selector ("#deck");
 Let pass: HTML Buton Element = document . query selector ("# pass");
 let pcturn: boolean = false;
 let pass Allowed: boolean = false;
 let pcNoHatch: boolean= false;
```

prompt on User: "How many cards do you want to start the game with?"

Ly Speichern in let user Prompt: string

string von prompt umwandeln in number mit parseint (string). Ly speichern in eine Variable let cards Number

> Funktion aufrugen, die die staitkarten für den User generiert: generate Player Cards (); H

Funktion aufrufeni die die Startkarten für den PC generiert: generate Pc Cards (); H

Funktion aufrufen, die die erste (zufällige) harte auf dem Ablagestapel (current Card) generiert: Create First Card (); A

```
generate Player Cards

function generate Player Cards (): void {

let i: number = 0;

while (i < Card Number) {

    create Random Values ();

let card: HTML Span Element = document. create Element ("span");

    player Hand. append Child (card);

    card. text Content = random Number Value;

    card. class Name = random Color Value;

    card. add Event Listener ("click", choose Card);

    i++;

}
```

```
function generate PcCards (): void {

let i: number = 0;

while (i < card Number) {

    create Random Values ();

let card: HTHLSpan Element = document.creale Element ("span");

    pc Hand. append Child (card);

    card.text content = random Number Value;

    card.class Name = random Color Value;

    card.style.color = "light slate grey";

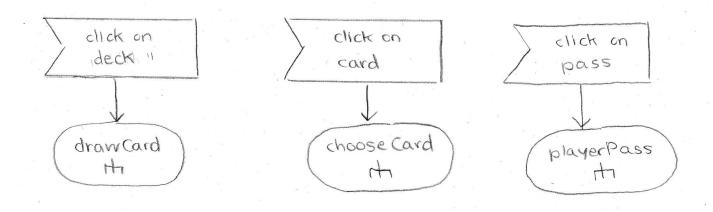
    card.style.background Color = "light slate grey";

    itt
```

```
create First Card (): void {
function create First Card (): void {
create Random Values ();
current Card · text Content = random Number Value;
current Card · class Name = random Color Value;
```

```
handle PcTurn

function handle PcTurn (): void {
    if (pcTurn) {
        pcMove (); It
    }
}
```



```
drawCard -event: MouseEvent

let card: HTHLSpanElement=document.createElement("span");
playerHand.appendChild(card);
createRandomValues(); Ita
card.textContent="random NumberValue";
card.className="random ColorValue";
passAllowed=true
```

```
playerPass - event: House Event

if (pass Allowed == true) {

pcTum = true;

handle PcTum H
```

```
Create Random Values (): void &

function create Random Values (): void &

random Color Value = possible Colors [Hath.floor (Hath.random ())

* possible Colors.length)];

let x: number = possible Numbers [Math.floor (Math.random ())

* possible Numbers.length)];

random Number Value = x. to String ();
```

```
pcMove
function perove (): void ;
    let all cards Pc: HTML Collection = pc Hand . children ;
    PCNOHOHON = true;
    for (let i: number=0; icall cards Pc. length ji++) }
        if (all cards PCE i]. Lext Content == current Card. Lext Content 11
            all Cards PC[i]. class Name == curren+Card. class Name) }
             current (ard. text (ontent = all (ards Pc[i]. text (ontent;
             current Card. class Name = all Cards Pc [i]. class Name;
             pcHand, remove Child (all Cards Pc [i]);
             pass Allowed = False; K
             pcTurn = false;
                                        'evaluate Winner(); +
             pc No Hatch = False;
             break;
      if (pc NoMatch) }
         pedraw();
```

```
pcDraw
function pedraw(): void }
   create Random Values (); it
   let card: HTHL span Element = document. create Element ("span");
   petand. appendehild (card);
    card. text content = random Number Value;
    card. class Name = random Color Valve;
    card. style : color = "lightslategrey";
    card. style. backgroundColor = Hightslategrey ";
    if (card . text Content == current Card . text Content 11
       card · class Name == current (ard · class Name) &
       current card. text Content = card. text content;
       current Card. class Name = card. class Name;
       pcHand, remove Child (card);
    pcTurn = false
     passAllowed - False
```

```
evaluate Winner

function evaluate Winner(): void {

let all Cards Pc: HTML Collection = pc Hand-children;

let all Cards Player: HTML collection = player Hand.children;

if call Cards Pc. length == 0) {

alert ("bu hast verloren: (");

location.reload();

}

if (all Cards Player.length == 0) {

alert ("bu hast gewonnen");

location.reload();

}
```