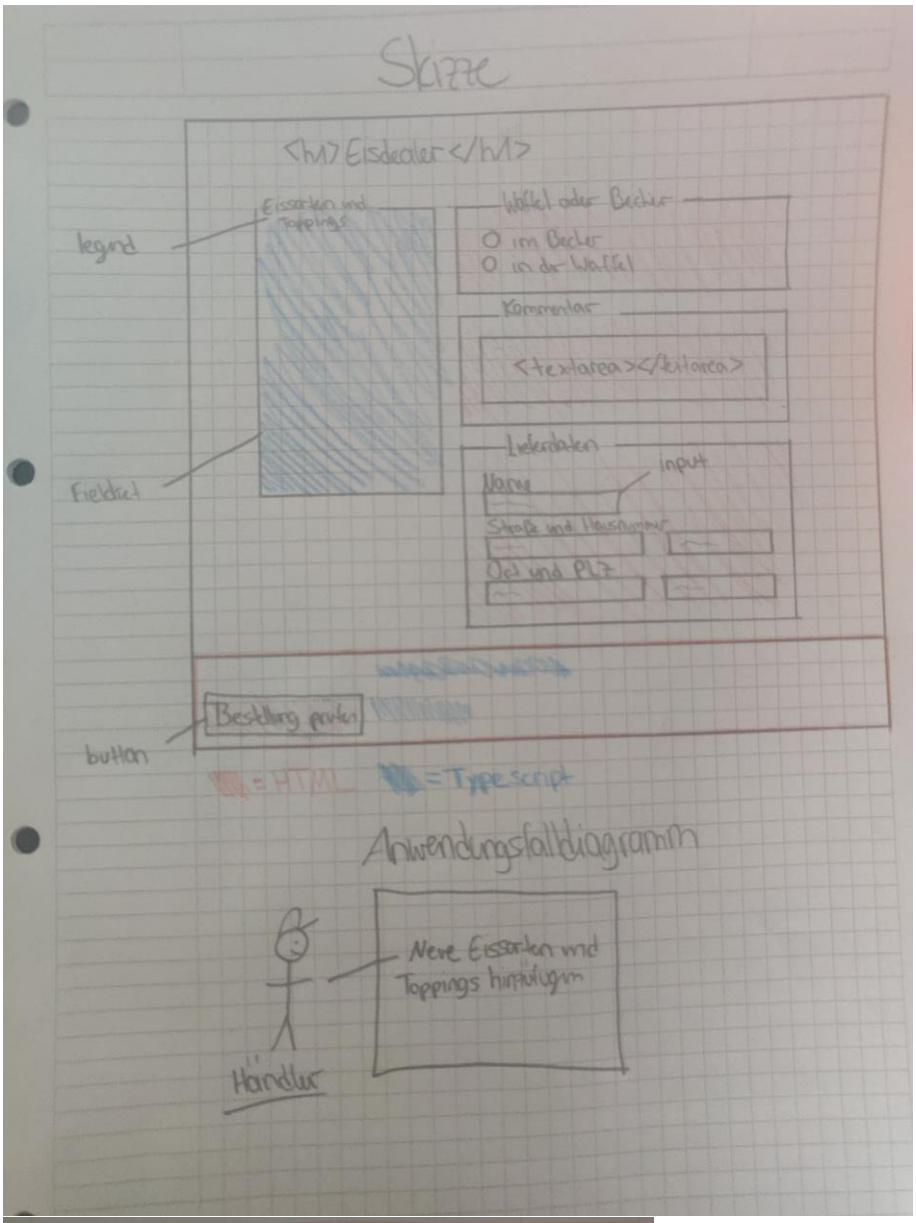
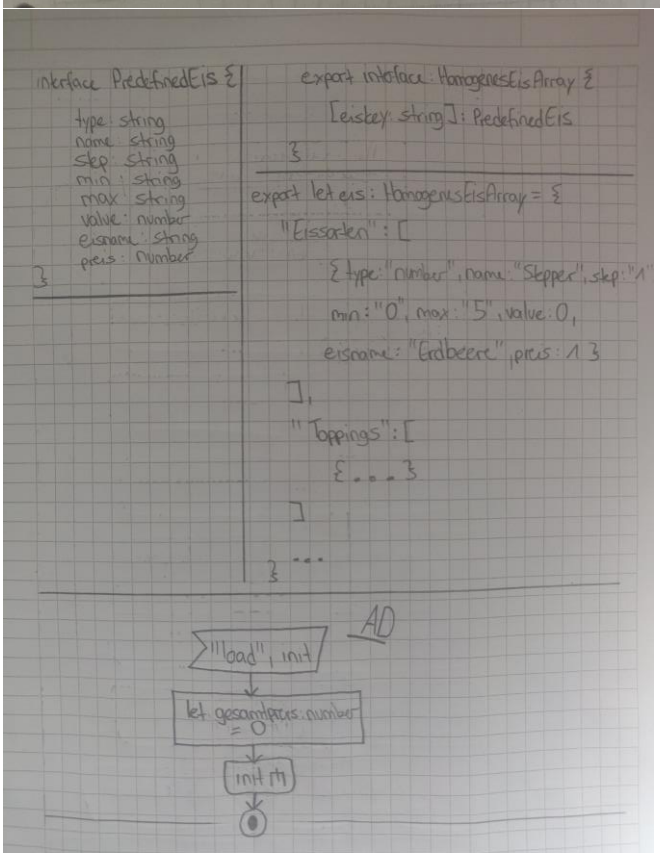
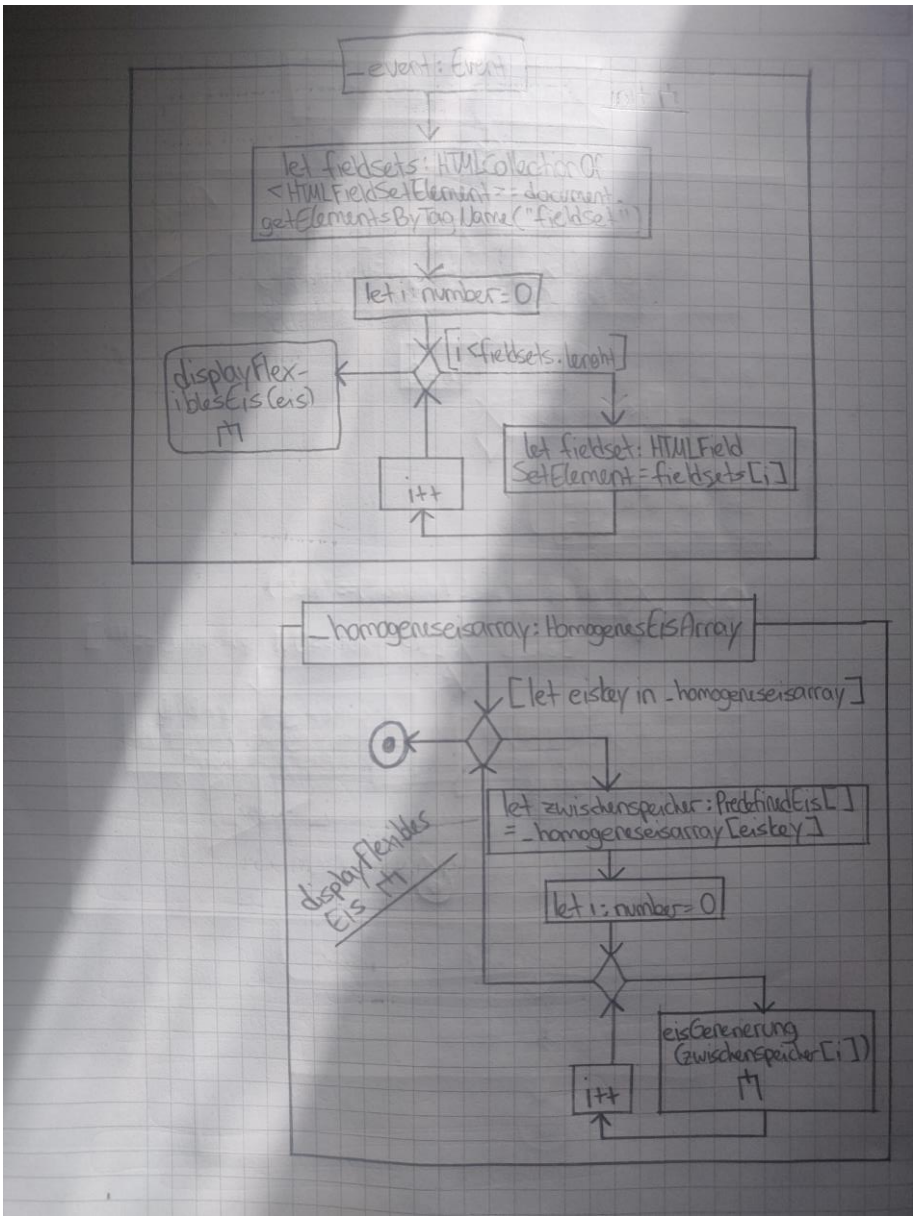
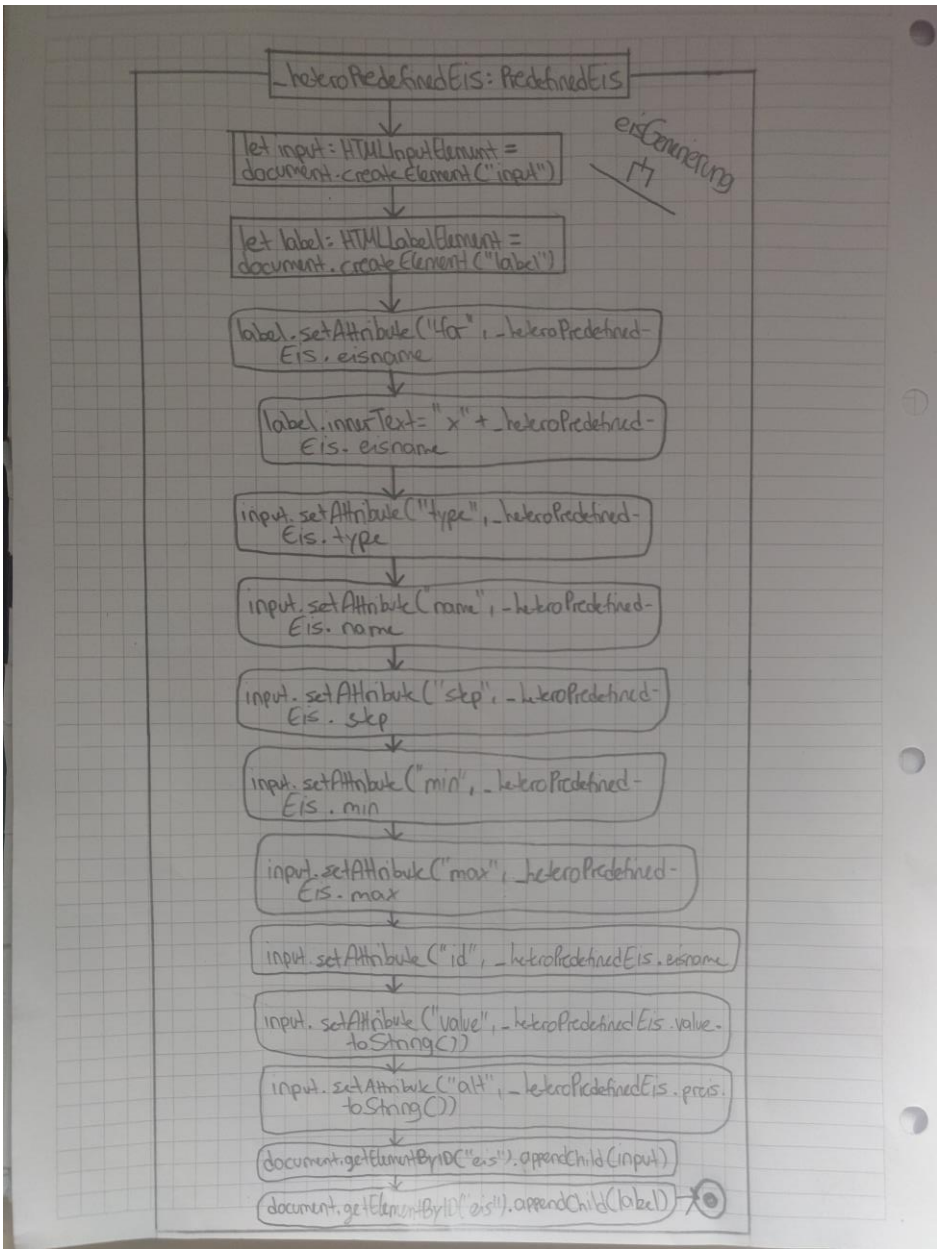
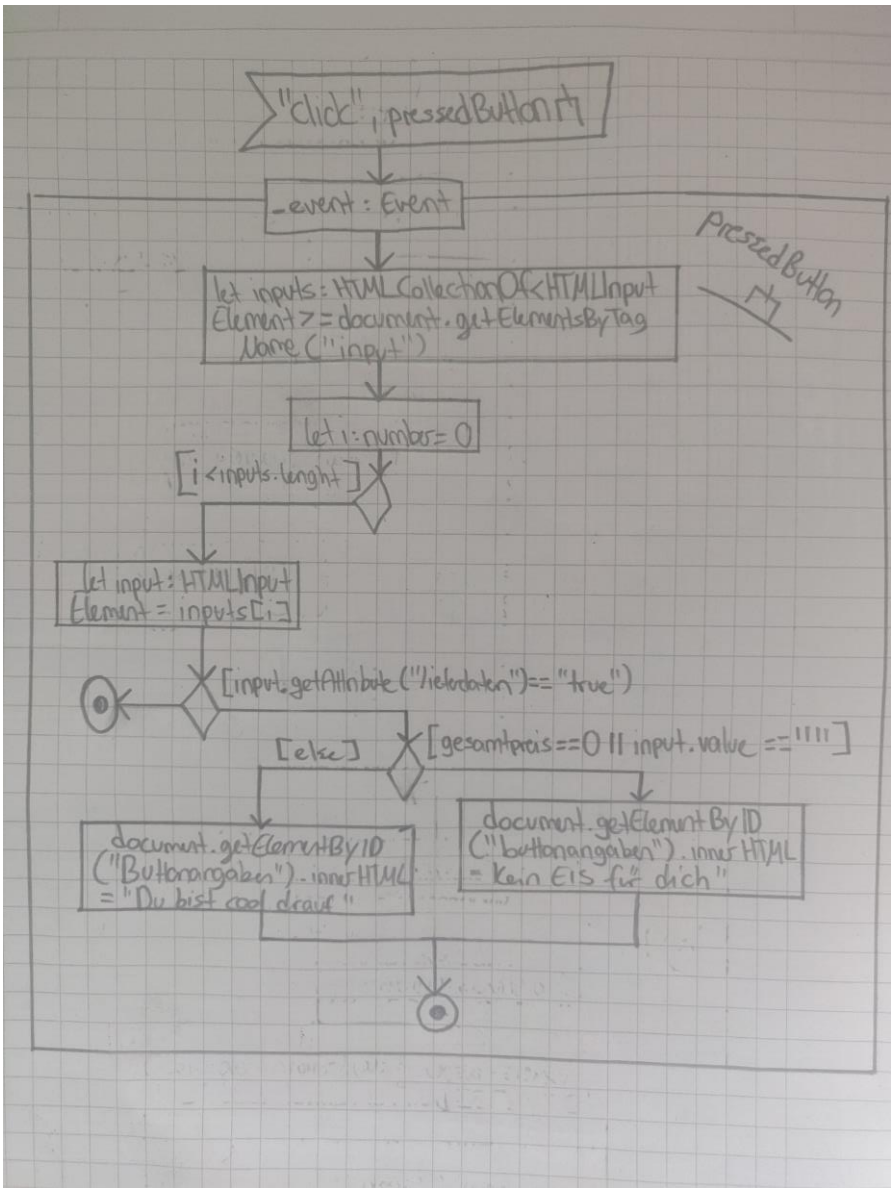
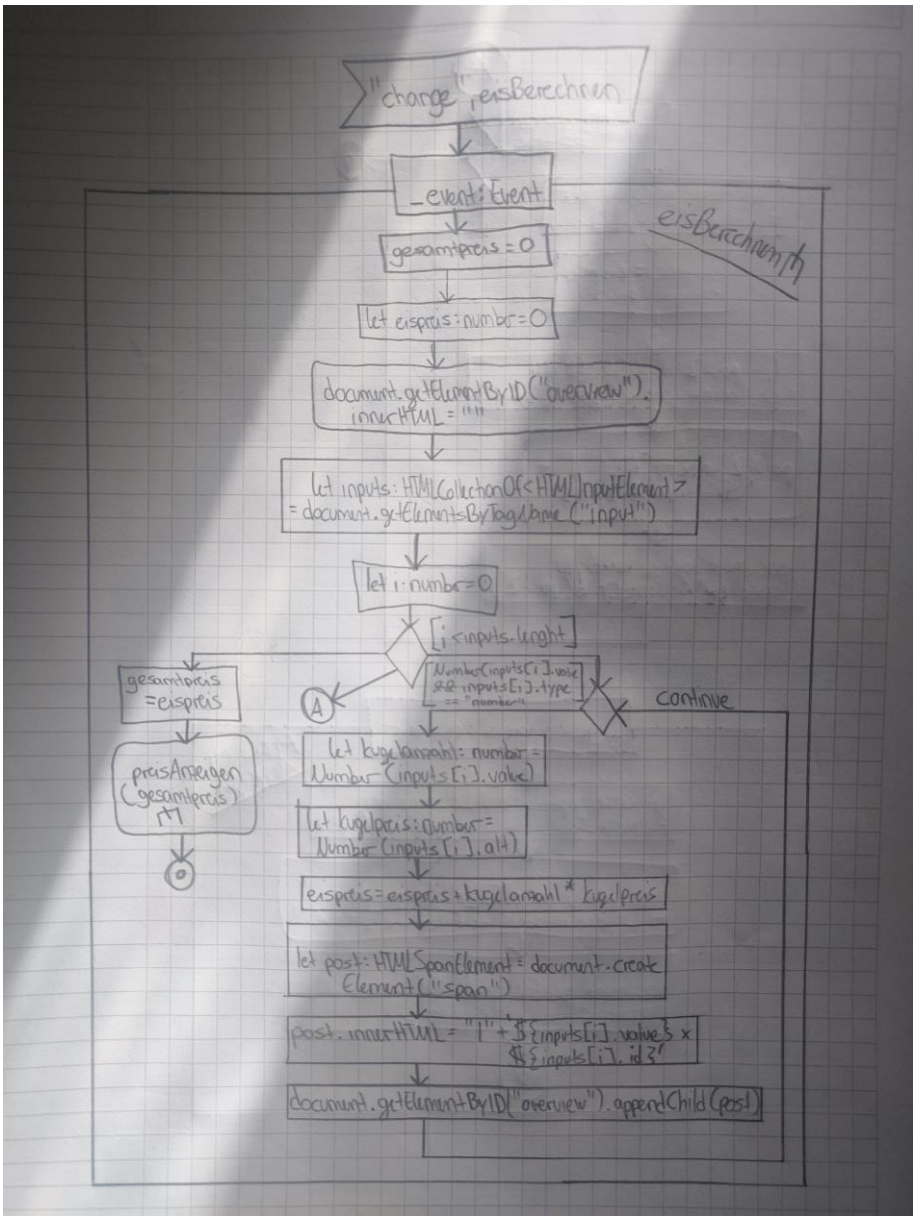

Skizze
<h1> Eisdealer </h1>
legend
Eissorten und Toppings
Waffel oder Becher
O im Becher
O in der Waffel
Kommentar
<textarea></textarea>
Fieldset
Lieferdaten
Name
input
Straße und Hausnummer
Ort und PLZ
button
Bestellung prüfen
= HTML
= Typescript
Anwendungsfalldiagramm
Neue Eissorten und Toppings hinzufügen
Händler

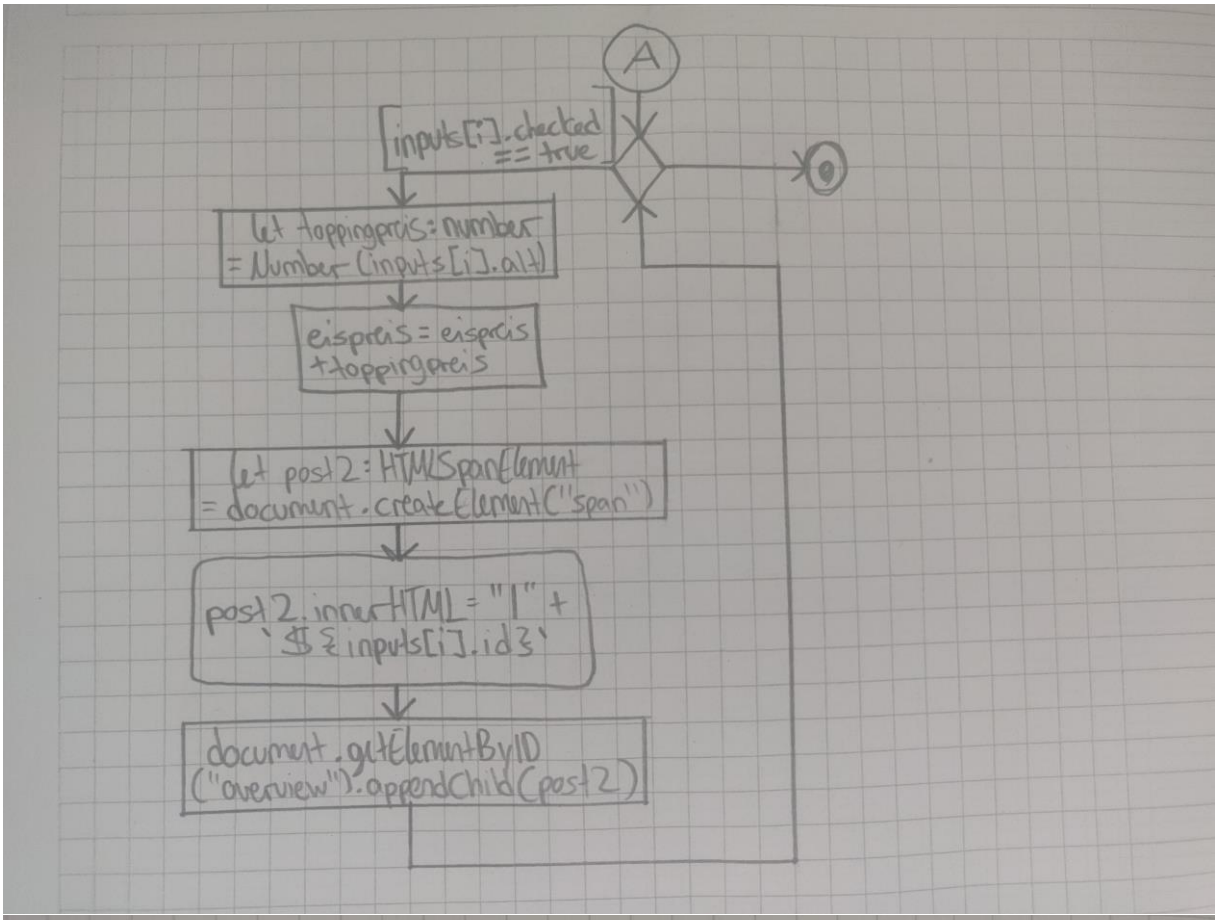

interface PredefinedEis {
type: string
name: string
step: string
min: string
max: string
value: number
eisname: string
preis: number
}
export interface HomogenesEisArray {
[eiskey: string]: PredefinedEis
}
export let eis: HomogenesEisArray = {
"Eissorten": [
{type: "number", name: "Stepper", step: "1"
min: "0", max: "5", value: 0,
eisname: "Erdbeere", preis: 1.3
],
"Toppings": [
{...}
]
}...
AD
"load", init
let gesamtpreis: number = 0
init fn

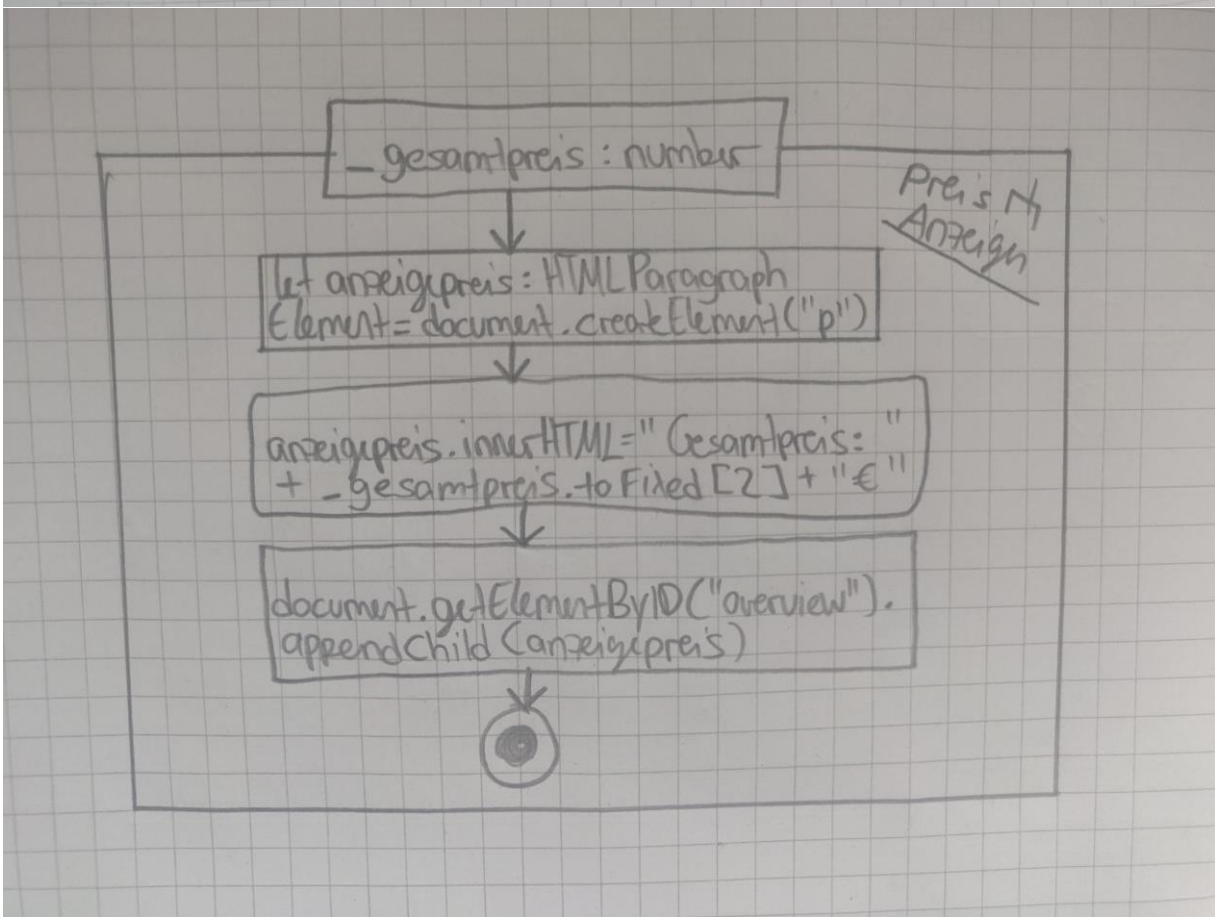

_event: Event
let fieldsets: HTMLCollectionOf
<HTMLFieldSetElement> = document.
getElementsByTagName("fieldset")
let i: number = 0
displayFlex-
ibleEis(eis)
i<fieldsets.length
let fieldset: HTMLField
SetElement = fieldsets[i]
i++
_homogeneseisarray: HomogenesEisArray
[let eistyp in _homogeneseisarray]
let zwischenspeicher: PredefinedEis[]
= _homogeneseisarray[eistyp]
displayflexibles
Eis
let i: number = 0
eisGenerierung
(zwischenspeicher[i])
i++


_heteroPredefinedEis: PredefinedEis
let input: HTMLInputElement = document.createElement("input")
let label: HTMLLabelElement = document.createElement("label")
label.setAttribute("for", _heteroPredefinedEis.eisname
label.innerText= "x" + _heteroPredefinedEis.eisname
input.setAttribute("type", _heteroPredefinedEis.type
input.setAttribute("name", _heteroPredefinedEis.name
input.setAttribute("step", _heteroPredefinedEis.step
input.setAttribute("min", _heteroPredefinedEis.min
input.setAttribute("max", _heteroPredefinedEis.max
input.setAttribute("id", _heteroPredefinedEis.eisname
input.setAttribute("value", _heteroPredefinedEis.value.toString())
input.setAttribute("alt", _heteroPredefinedEis.preis.toString())
document.getElementById("eis").appendChild(input)
document.getElementById("eis").appendChild(label)
eisGenerierung


"click", pressedButton
-event: Event
PressedButton
let inputs: HTMLCollectionOf<HTMLInput Element> = document.getElementsByTag Name("input")
let i: number = 0
[i < inputs.lenght]
let input: HTMLInput Element = inputs[i]
[input.getAttribute("ielebaten") == "true"]
[else]
[gesamtpreis == 0 || input.value == ""]
document.getElementById ("Buttonangaben").innerHTML = "Du bist cool drauf"
document.getElementById ("buttonangaben").innerHTML = "kein Eis für dich"


"change",eisBerechnen
_event: Event
eisBerechnen()
gesamtpreis = 0
let eispreis: number = 0
document.getElementById("overview").
innerHTML = ""
let inputs: HTMLCollectionOf<HTMLInputElement>
= document.getElementsByTagName("input")
let i: number = 0
[i < inputs.lenght]
gesamtpreis
=eispreis
preisAnzeigen
(gesamtpreis)
A
Number(inputs[i].value)
&& inputs[i].type
== "number"
continue
let kugelanzahl: number =
Number(inputs[i].value)
let kugelpreis: number =
Number(inputs[i].alt)
eispreis = eispreis + kugelanzahl * kugelpreis
let post: HTMLSpanElement = document.create
Element("span")
post.innerHTML = "1" + ${inputs[i].value} x
${inputs[i].id}"
document.getElementById("overview").appendChild(post)


A
inputs[i].checked == true
let toppingpreis: number = Number(inputs[i].alt)
eispreis = eispreis + toppingpreis
let post2: HTMLSpanElement = document.createElement("span")
post2.innerHTML = "|" + `${inputs[i].id}`
document.getElementById("overview").appendChild(post2)


_gesamtpreis : number
Preis Anzeigen
let anzeigepreis: HTMLParagraphElement = document.createElement("p")
anzeigepreis.innerHTML = "Gesamtpreis: " + _gesamtpreis.toFixed(2) + "€"
document.getElementById("overview").appendChild(anzeigepreis)