

Hendelsesobjektet

Vi har så langt sett på enkle funksjoner som håndterer hendelser på ett element. Vi kan også legge til samme lytter på flere elementer.

Vi kan for eksempel lage en side der alle `<p>`-elementer veksler mellom to bakgrunnsfarger når vi trykker på dem. Da er det praktisk å bruke den samme funksjonen til å håndtere alle hendelser, men hvordan vet vi hvilket `<p>`-element vi da har trykket på?

Alle hendelser kommer med et *hendelsesobjekt*, og i dette objektet kan vi finne nyttig informasjon om hendelsen, for eksempel hvilket element vi trykket på.

For å illustrere dette skal vi se på et klassisk sannsynlighetsproblem, kjent som «Monty Hall»-problemet. Monty Hall var programleder i det amerikanske TV-programmet «Let's Make a Deal», der et av innslagene lot deltagerne prøve å vinne en bil ved å velge den riktige av tre dører. Bak to av dørene står det geiter, men bak én av dørene står det en bil. Problemet fungerer slik:



Monty Hall i TV-programmet «Let's Make A Deal»

- 1 Deltageren velger én av tre dører.
- 2 Når deltageren har valgt en dør, åpner programlederen en av de to andre dørene, der det står en geit.
- 3 Deltageren får nå anledning til å bytte dør, til den andre uåpnede døren.
- 4 Programlederen åpner døren som deltageren bestemmer seg for.

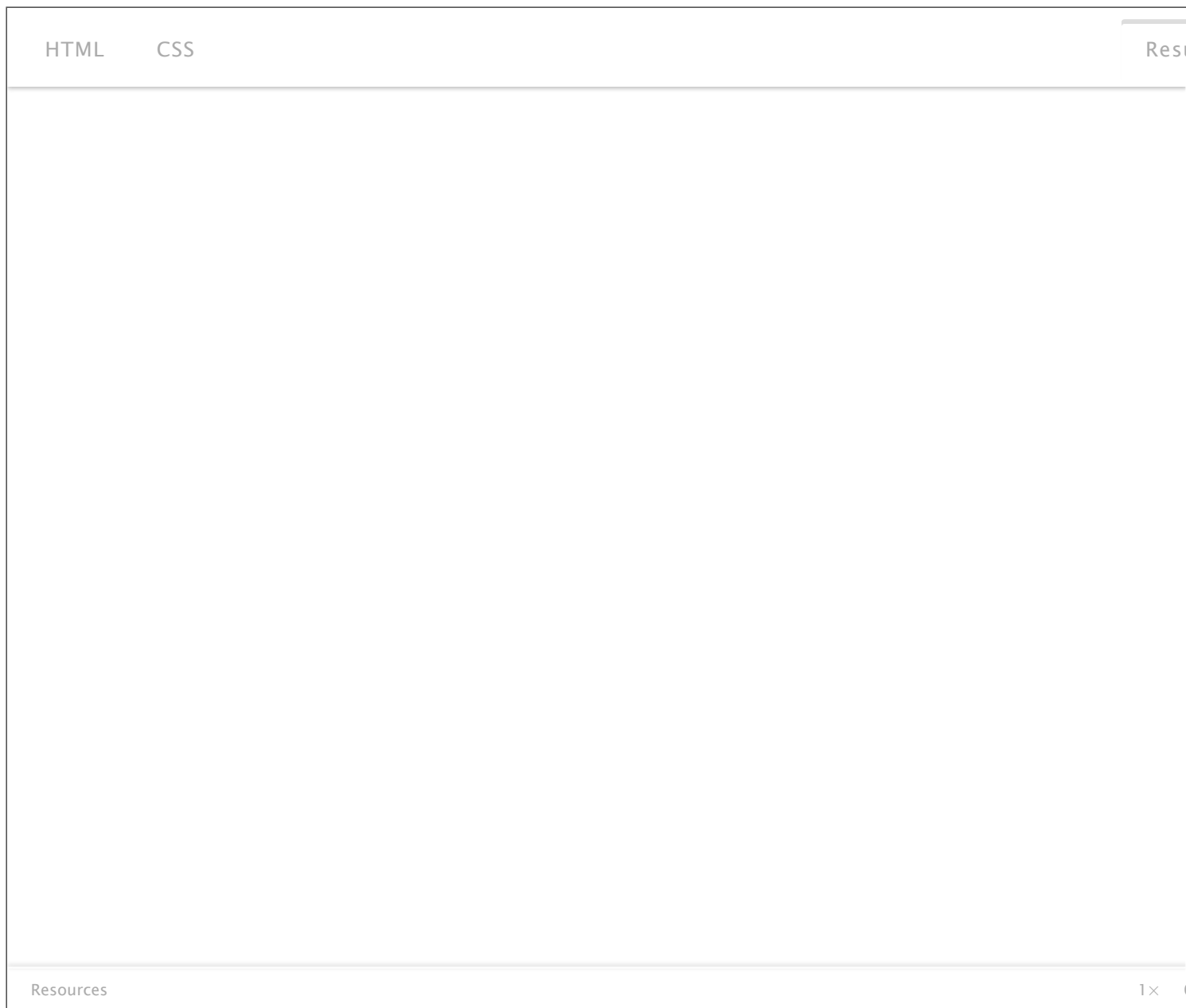
Denne oppgaven leder til et morsomt sannsynlighetsproblem: Bør deltageren beholde sin opprinnelige

dør, eller bør deltageren bytte dør?

Vi skal se på hvordan vi kan lage vår egen versjon av dette problemet, ved å lage en nettside med tre «dører» som brukeren kan klikke på. I koden nedenfor kan du se utgangspunktet før vi har skrevet noe JavaScript. Bildene i dette eksemplet ligger sammen med koden på CodePen, og har derfor en ganske lang tekst i `src` - attributtet. Husk at det holder å skrive `"dor_1.png"` , `"dor_2.png"` og `"dor_3.png"` hvis du jobber lokalt og har filene i samme mappe som HTML-dokumentet.

Bildene av dører, bil og geit som brukes i dette eksemplet, kan du laste ned her:

[Bilder_Monty_Hall.zip](#)



Nå må vi bare finne ut *hvilken* dør brukeren trykker på. Vi må starte med å hente bildeelementene og legge til lyttere. Vi har bare tre ``-elementer på siden, så vi kan bruke `querySelectorAll()` for å få tak i alle dørene.

```
let dorbilderEl = document.querySelectorAll("img");
```



```
// Går gjennom alle bildene og legger på samme lytter på hver av
for (let i = 0; i < dorbilderEl.length; i++) {
  dorbilderEl[i].addEventListener("click", sjekkDor);
}
```

```
function sjekkDor(e) {
  console.log(e.target);
  console.log(e.type);
}
```

Hendelsesobjektet

Mye av koden ovenfor er kjent fra tidligere, men den store forskjellen er argumentet `(e)` som sendes til funksjonen `sjekkDor(e)`. Bokstaven «e» står for «event» (eng. for hendelse), og i dette tilfellet er `e` et objekt som sendes som argument til funksjonen vår. Vi har valgt å bruke bokstaven «e» for dette objektet fordi det er vanlig å bruke den, men vi kunne like gjerne ha brukt «h» for «hendelse».

Hendelsesobjektet `(e)` inneholder mye informasjon om hendelsen, for eksempel hvilken hendelse som inntraff (`e.type`), og hvilket element hendelsen skjedde på (`e.target`). Ved å bruke `e.target` kan vi finne ut hvilket av de tre bildene brukeren klikket på. La oss utvide funksjonen `sjekkDor(e)` ovenfor slik at vi kan se hvilken dør brukeren klikket på:

```
function sjekkDor(e) {  
  let trykketDor = e.target;  
  console.log(trykketDor.id);  
  
  if (trykketDor.id === "dor1") {  
    console.log("Du trykket på dør 1.");  
  } else if (trykketDor.id === "dor2") {  
    console.log("Du trykket på dør 2.");  
  } else {  
    console.log("Du trykket på dør 3.");  
  }  
}
```



Her henter vi først elementet vi klikket på, og lagrer det i variabelen `trykketDor`. Deretter sjekker vi hvilken id elementet har, og skriver ut en beskjed til konsollen basert på verdien til id-en.

Med tilgang til bildeelementet har vi også muligheten til å bytte ut bildet vi har klikket på. Da kan vi gjøre om bildefilen (i `trykketDor.src` eller `e.target.src`) til for eksempel `"geit.png"` eller `"bil.png"`, avhengig av hva som befinner seg bak døren vi klikker på. I eksemplet nedenfor har vi lagt til dette, med bilen plassert bak den midterste døren.



Oppgaver

7 Du skal nå lage ditt eget Monty Hall-program.

- a Lag en forenklet versjon av spillet beskrevet ovenfor. Bestem hvilken dør bilen skjuler seg bak, og når brukeren trykker på en av dørene, vises det som skjuler seg bak den (en geit eller en bil).
- b Bruk `Math.random()` for å plassere bilen bak en tilfeldig valgt dør.
- c I originalversjonen av spillet åpnes en av de to dørene som ikke ble valgt (en dør som skjuler en geit). Deretter kan deltageren velge mellom sin opprinnelige dør og den

andre uåpnede døren. Utvid koden din slik at dette blir mulig.

- d Finn en medspiller og velg hver deres strategi. En av dere står alltid på den første valgte døren, mens den andre alltid bytter dør. Hvem vinner mest? Hvis en av dere vinner mer enn den andre, hvorfor er det slik?

- 8 Lag en ny nettside med tre `<div>`-elementer med ulike farger. Gi hvert av `<div>`-elementene attributtet `data-farge` med samme verdi som `<div>`-elementets farge. Når brukeren klikker på et av `<div>`-elementene, skal nettsidens bakgrunnsfarge endres til fargen til `<div>`-elementet brukeren klikket på.
- 9 Lag et rutenett av `<div>`-elementer på tre ganger tre ruter. Når én av rutene klikkes, skal en tekst vises i et annet element, der det står hvilken rad og hvilken kolonne ruta befinner seg i. For å løse denne oppgaven kan det være lurt å lage en en løkke inni en annen løkke.