

Inleiding Programmeren – alle javascript code

Jayden Ho – Fuchsia

Mijn Javascript code:

```
console.log("test")

var kiesmanImg = document.querySelector("#kiesman");
var kiesvrouwImg = document.querySelector("#kiesvrouw");

var restPlaceImg = document.querySelector("#restplace");
var workPlaceImg = document.querySelector("#workplace");
var karakterImg = document.querySelector("#karakter");
var aantalPuntenImg = document.querySelector("#primobar");
var energyBarImg = document.querySelector("#energybar");
var usePrimo = document.querySelector("#useprimo");
var stopSpel = document.querySelector("#stopspel")

var puzzleImg = document.querySelector("#puzzle");
var puzzles = ["images/0piece.png", "images/1piece.png", "images/2piece.png",
"images/3piece.png", "images/4piece.png", "images/5piece.png"]

var textElementEnergy = document.querySelector("#energyprocent");
var textElementPunten = document.querySelector("#punten");

var energy = 100;
var punten = 0;
var AantalPuzzleStukjes = 0;
var puzzleStatus = 0;
var endGame = false;

var feedback = document.querySelector("#feedback");
var restSound = new Audio("audio/resttake2.mp3");
var puzzleSound = new Audio("audio/puzzle.mp3");
var workSound = new Audio("audio/work.mp3");
var succesSound = new Audio("audio/succes.mp3");

// DOM manipulatie functie bed, console log weggehaald want zorgde voor "niet
meer" zo belangrijke tekst
function startSchudden01() {
    restPlaceImg.classList.add('schud')
}

function stopSchudden01() {
    restPlaceImg.classList.remove('schud')
}

restPlaceImg.addEventListener('mouseenter', startSchudden01)
```

```

restPlaceImg.addEventListener('mouseleave', stopSchudden01)

// Dom manipulatie functie werkplek, console log weggehaald want zorgde voor
// "niet meer" zo belangrijke tekst
function startSchudden2() {
    workPlaceImg.classList.add('schud')
}

function stopSchudden2() {
    workPlaceImg.classList.remove('schud')
}

workPlaceImg.addEventListener('mouseenter', startSchudden2)
workPlaceImg.addEventListener('mouseleave', stopSchudden2)

// Dom manipulatie functie use knop, console log weggehaald want zorgde voor
// "niet meer" zo belangrijke tekst
function startSchudden3() {
    usePrimo.classList.add('schud')
}

function stopSchudden3() {
    usePrimo.classList.remove('schud')
}

usePrimo.addEventListener('mouseenter', startSchudden3)
usePrimo.addEventListener('mouseleave', stopSchudden3)

// Dom manipulatie functie use knop, console log weggehaald want zorgde voor
// "niet meer" zo belangrijke tekst
function startSchudden4() {
    stopSpel.classList.add('schud')
}

function stopSchudden4() {
    stopSpel.classList.remove('schud')
}

stopSpel.addEventListener('mouseenter', startSchudden4)
stopSpel.addEventListener('mouseleave', stopSchudden4)

// Werken voor punten, ik heb hier geschreven punten = punten + 1, die punten
// aan het begin variable van gemaakt en een waarde van 0 gegeven,
// wat ik daarna zeg is dat bij de functie updatePunten er elke keer +1 komt
// bij de punten en dat er een workSound komt die afgespeeld moet worden,
// in de if statement vertel ik dat als punten meer dan 5 worden, dat het 5
// moet blijven, zodat er niet oneindig punten kunnen komen want mijn bedoeling
// is dat er maar

```

```

// max 5 punten verdient kunt worden, maar dit gebeurt allemaal achter de
schermen, om de punten ook daadwerkelijk op de scherm te veranderen, heb ik de
textContent
// van de variable textElementPunten gezet dat het de punten moet laten zien

function updatePunten() {
    console.log("update punten aantal" + "1")
    punten = punten + 1;
    workSound.play();
    if (punten > 5) {
        punten = 5
    }
    textElementPunten.textContent = punten;
}

// energy eraf bij het werken, de bedoeling is hier dat als je werkt dat er
ook 25% energie afgaat, dit doe ik door energy een waarde te geven van 100
// en elke keer bij het werken in een functie later in de code, 25% energie af
te halen, en zoals bij updatePunten is dit nu iets wat achter de schermen
gebeurt.
// textelement moet ik dus ook gelijk maken aan de waarde energie zodat dat
ook daadwerkelijk op het scherm laten zien wordt
function updateEnergy() {
    console.log("energy eraf")
    energy = energy - 25;
    textElementEnergy.textContent = energy + "%";
}

// rusten om weer energy te krijgen, hier gebruik ik een if statement waarin
staat, als energie kleiner is dan 100, dan kan energy erbij en dan het zelfde
verhaal als
// updateEnergy en updatePunten, dus achter de schermen en daadwerkelijk iets
op het scherm laten zien. Daarna zeg ik dat als energy dus kleiner is dan 100
dan kan je werken
// en komt er een animatie op het poppertje img die ik in de css heb
aangemaakt, met set time out remove ik weer die class na 3500 ms zodat het
javascript mannetje niet denkt
// van als ik nog een keer klik " de class zit er al op dus er is niks uit te
voeren", hierdoor kan je dus constant die animatie blijven zien als je kan
rusten. En heb hier dan
// ook weer een audio toegevoegd wanneer je gaat slapen. In de else statement
staat als er niet aan de if wordt voldoen, vertel dan ... zodat de gebruiker
weet
// wat die moet doen, dit heb ik met een feedback hidden gedaan, standaard is
die true maar hier vertel ik dus dat die false moet worden en dat pas komt dat
tekstje tevoorschijn
// en gaat die automatisch weer weg na 1500ms en wordt de boolean weer true
zodat dat tekstje weer verdwijnt.

```

```

function rusten() {
  if (energy < 100) {
    console.log("rusten energy erbij")
    energy = energy + 25;
    textElementEnergy.textContent = energy + "%";
    karakterImg.classList.add('gorest')
    setTimeout(() => {
      karakterImg.classList.remove('gorest')
    }, 3500)
    restSound.play();
  } else {
    feedback.textContent = "Ik heb genoeg energie hoor!"
    feedback.hidden = false;
    setTimeout(() => {
      feedback.hidden = true;
    }, 1500)
  }
}
restPlaceImg.addEventListener('click', rusten)

// werken voor punten om puzzle stukken te krijgen, hier zeg ik dus als energy
// groter is dan 0 dat de functies updatePunten en updateEnergy uitgevoerd moeten
// worden
// bij een click op workPlaceImg, je poppertje beweegt hier dan ook met behulp
// van de css naar de workPlaceImg en die animatie wordt dan automatisch
// verwijderd na 1500ms
// net zoals bij de rusten function om dezelfde reden. En als het niet aan de
// if statement voldoet, dan moet de else statement worden uitgevoerd op dezelfde
// manier als
// bij rusten maar dan met een iets andere feedback textcontent.
function werk() {
  if (energy > 0) {
    console.log("werken")
    updatePunten();
    updateEnergy();
    karakterImg.classList.add('gowork')
    setTimeout(() => {
      karakterImg.classList.remove('gowork')
    }, 1500)
  } else {
    feedback.textContent = "Ik heb wel genoeg gewerkt!"
    feedback.hidden = false;
    setTimeout(() => {
      feedback.hidden = true;
    }, 1500)
  }
}

```

```

workPlaceImg.addEventListener('click', werk)

function startSchudden05() {
  puzzleImg.classList.add('schud')
}

// Puzzle stuk erbij, als punten groter zijn dan 0 dan kan je een puzzle stuk
erbij doen, puzzleStatus houdt de waarde bij van de puzzle stukjes, in regel
185 wordt gezegd:
// puzzleImg.src (image element uit de html) vervangen voor de img's van de
array en de puzzleStatus in de array bepaalt welke img uit de array.
// En wordt er ook een zelfingesproken audio afgespeeld van dat de puzzle stuk
erbij komt. Daarna heb ik een if in een if statement gedaan, hierin staat als
puzzleStatus meer is
// dan de 4e array dus de 5e array is (puzzle compleet) dan is het endgame en
begint de puzzle te schudden en geeft het poppertje feedback van puzzle is
compleet.
// in de else statement staat dat als je niet genoeg punten hebt, dus punten
niet meer dan 0 dan komt weer een textje bij het poppertje tevoorschijn van "
je hebt niet genoeg"
// zodat gebruiker weet dat die moet gaan werken
function addPuzzle() {
  if (punten > 0) {
    console.log("puzzle stuk erbij" + puzzleStatus);
    puzzleStatus = puzzleStatus + 1
    punten = punten - 1
    textElementPunten.textContent = punten
    puzzleSound.play();

    if (puzzleStatus > 4) {
      console.log("finishtest")
      puzzleStatus = 5
      endGame = true;
      startSchudden05();
      feedback.textContent = "Topper, puzzle is compleet!"
      feedback.hidden = false;
      setTimeout(() => {
        feedback.hidden = true;
      }, 7500)
      succesSound.play();
    }

    puzzleImg.src = puzzles[puzzleStatus]
  }
}

```

```
    } else {  
      feedback.textContent = "Je hebt niet genoeg!"  
      feedback.hidden = false;  
      setTimeout(() => {  
        feedback.hidden = true;  
      }, 1500)  
    }  
  }  
}  
  
usePrimo.addEventListener("click", addPuzzle);
```