Veel gemaakte fouten

Fout 1

```
    <form>
      <div class="mb-3">
        <label for="firstname" class="form-label">first name</label>
        <input type="text" class="form-control" name="firstname" id="firstname" placeholder="firstname">
      </div>
      <div class="mb-3">
        <label for="lastname" class="form-label">last name</label>
        <input type="text" class="form-control" name="lastname" id="lastname" placeholder="lastname">
      </div>

      <a>
        <button type="button" class="btn btn-primary">send</button>
      </a>
    </form>
```

Fout 1

- plaats geen buttons in a-tags of omgekeerd
- Beiden zijn elementen om klik events op te vangen.
- Je kan een a-tag als button gebruiken

Oplossing

```
    <form>
      <div class="mb-3">
        <label for="firstname" class="form-label">first name</label>
        <input type="text" class="form-control" name="firstname" id="firstname" placeholder="firstname">
      </div>
      <div class="mb-3">
        <label for="lastname" class="form-label">last name</label>
        <input type="text" class="form-control" name="lastname" id="lastname" placeholder="lastname">
      </div>


      <button type="button" class="btn btn-primary">send</button>

    </form>
```

Fout 2

```
const ul = document.getElementById('items');
const liItems = document.querySelectorAll('li');

const items = ['eerste item', 'tweede item', 'derde item'];



for (let index = 0; index < items.length; index++) {
  const element = items[index];
  const li = document.createElement('li');
  li.classList.add('li');
  li.innerText = element;
  ul.append(li);
}

liItems.forEach((element)=> {
  element.addEventListener('click', ()=> {
    element.remove();
  });
});
```

Fout 2

- LiItems is een lijst van li-items voor we deze hebben toegevoegd
- Deze lijst is dus leeg

Oplossing 1

```
const ul = document.getElementById('items');

const items = ['eerste item', 'tweede item', 'derde item'];



for (let index = 0; index < items.length; index++) {
  const element = items[index];
  const li = document.createElement('li');
  li.classList.add('li');
  li.innerText = element;
  ul.append(li);
}

const liItems = document.querySelectorAll('li');


liItems.forEach((element)=> {
  element.addEventListener('click', ()=> {
    element.remove();
  });
});
```

Oplossing 2

```
const ul = document.getElementById('items');

const items = ['eerste item', 'tweede item', 'derde item'];



for (let index = 0; index < items.length; index++) {
  const element = items[index];
  const li = document.createElement('li');
  li.classList.add('li');
  li.innerText = element;
  ul.append(li);

  li.addEventListener('click', ()=> {
    li.remove();
  });

}
```

Oplossing 3

```
const ul = document.getElementById('items');
const liItems = document.getElementsByTagName('li');

const items = ['eerste item', 'tweede item', 'derde item'];



for (let index = 0; index < items.length; index++) {
  const element = items[index];
  const li = document.createElement('li');
  li.classList.add('li');
  li.innerText = element;
  ul.append(li);
}

Array.from(liItems).forEach((element)=> {
  element.addEventListener('click', ()=> {
    element.remove();
  });
});
```

Fout 3

```javascript
const addButton = document.getElementById('add-button');
const productElement = document.getElementById('product');
const productsList = document.getElementById('products');

const product  = productElement.value;

addButton.addEventListener('click', ()=> {
  const element = document.createElement('li');
  element.innerText = product;
  productsList.append(element);
});
```

Fout 3

- Value wordt opgehaald alvorens er iets van data in het input element zit.

Oplossing

```
const addButton = document.getElementById('add-button');
const productElement = document.getElementById('product');
const productsList = document.getElementById('products');


addButton.addEventListener('click', ()=> {
  const product  = productElement.value;
  const element = document.createElement('li');
  element.innerText = product;
  productsList.append(element);
});
```

Fout 4

```
const addButton = document.getElementById('add-button');
const productElement = document.getElementById('product');
const productsList = document.getElementById('products');
const changequantityitems = document.getElementsByClassName('changequantity');
const removeItemsButtons = document.getElementsByClassName('remove');
const shoppingcartItems = [];

function removeItems(){
  for (let i = 0; i < removeItemsButtons.length; i++) {
    const element = removeItemsButtons[i];
    element.addEventListener('click', e=> {
      e.preventDefault();
      element.parentNode.remove();
      shoppingcartItems.splice(i,1);
    });
  }
}

function changeQuantitiesEvent(){

  for (let index = 0; index < changequantityitems.length; index++) {

    changequantityitems[index].addEventListener('change', ()=>{
      shoppingcartItems[index].quantity +=1;
    });
    removeItems();
  }
}
```

```
function renderItems(){
  productsList.innerHTML = '';
  shoppingcartItems.forEach(item => {
    const liElement = `<li
class="item">${item.name} <input type="number"
class="changequantity" value=${item.quantity}> <a
class="remove" href=""s>x</a> </li>`;
    productsList.innerHTML += liElement;
    changeQuantitiesEvent();
  });
}


addButton.addEventListener('click', ()=> {
  shoppingcartItems.push({name:
productElement.value, quantity:1});

  renderItems();
});
```

Fout 4

- Er worden veel te veel events toegevoegd
- Voor elk shoppingcartitem wordt er voor alle number inputs een event toegevoegd
- Voor elk number input item worden voor alle remove buttons een event toegevoegd

Oplossing - deel 1

```
function changeQuantitiesEvent(){

  for (let index = 0; index < changeQuantityItems.length; index++) {

    changeQuantityItems[index].addEventListener('change', ()=>{
      shoppingCartItems[index].quantity +=1;
    });

  }

}

function renderItems(){
  productsList.innerHTML = '';
  shoppingCartItems.forEach(item => {
    const liElement = `<li class="item">${item.name} <input type="number" class="changequantity"
value=${item.quantity}> <a class="remove" href=""s>x</a> </li>`;
    productsList.innerHTML += liElement;
  });
  changeQuantitiesEvent();
  removeItems();

}
```

Oplossing - deel 2

```
function removeItems(){
  for (let i = 0; i < removeItemsButtons.length; i++) {
    const element = removeItemsButtons[i];
    element.addEventListener('click', e=> {
      e.preventDefault();
      element.parentNode.remove();
      shoppingCartItems.splice(i,1);
      // eslint-disable-next-line no-use-before-define
      renderItems();
    });
  }
}
```

Fout 5

```
const studentsUl = document.getElementById('students');

let studentData;

fetch('students.json').then (response=> {
  return response.json();
}).then((json)=> {
  studentData = json;
});


studentData.students.forEach(element => {
  const li = document.createElement('li');
  li.innerText = element.firstname;
  studentsUl.append(li);
});
```

Fout 5

- Fetch is asynchronous, dus we weten niet wanneer studentData wordt geïnitialiseerd

Oplossing

```
const studentsUl = document.getElementById('students');

let studentData;

fetch('students.json').then (response=> {
  return response.json();
}).then((json)=> {
  studentData = json;
  loadStudents();
});


function loadStudents(){

  studentData.students.forEach(element => {
    const li = document.createElement('li');
    li.innerText = element.firstname;
    studentsUl.append(li);
  });
}
```

Fout 6

```
const shoppingCartItems = [];


function saveShoppingCart(){
  shoppingCartItems.forEach(element => {
    localStorage.setItem(element.name, element.quantity);
  });
}

function loadShoppingCart() {
  for (let i = 0; i<localStorage.length; i++) {
    const key = localStorage.key(i);
    shoppingCartItems.push({name: key, quantity: Number(localStorage[key])});
  }
  renderItems();
}

loadShoppingCart();
```

Fout 6

- Het is niet de bedoeling om per item in een array een nieuw localstorage item te voorzien

Oplossing

```
let shoppingCartItems = [];


function saveShoppingCart(){
  localStorage.setItem('cart', JSON.stringify(shoppingCartItems));
}



function loadShoppingCart() {

  if(localStorage.getItem('cart')!= null){

    shoppingCartItems = JSON.parse(localStorage.getItem('cart'));
  }

  renderItems();
}
```

Fout 7

Zie github =>
https://github.com/OdiseeDWD/demos/blob/main/Les%207/example7/index.js

Fout 7

- De code is te veel gebonden aan de ingeladen JSON files
- Probeer je code data onafhankelijk te maken.
- Wanneer je nu 1000 studenten hebt, wordt deze code gigantisch.

Oplossing

```
function updateScore(id, score){

  let allStudentScores = JSON.parse(localStorage.getItem('scores'));
  if(allStudentScores==null){
    allStudentScores= [];
  }

  let studentScores = allStudentScores.find(e=> e.id == id);

  if(studentScores == undefined){
    studentScores = { id: id, scores: []};
    allStudentScores.push(studentScores);
  }

  studentScores.scores.push(score);

  localStorage.setItem('scores',JSON.stringify(allStudentScores));
  // eslint-disable-next-line no-use-before-define
  loadStudent(students);
}
```

```
function getScores(id){
  const allStudentScores =
JSON.parse(localStorage.getItem('scores'));
  if(allStudentScores == null){
    return null;
  }

  const studentScores = allStudentScores.find(i
=> i.id == id);
  if(studentScores == undefined){
    return null;
  }

  return studentScores.scores;

}
```