

Development website Oumy

Max-width

```
.maxWidth {  
  margin: 0 auto;  
  max-width: 1280px;  
  padding: 0 10px;  
}
```

De docent heeft mij aangeraden om gebruik te maken van 'max-width'. Hierbij heeft hij mij ook geholpen. Met 'max-width: 1280px' wordt de maximale breedte van het element beperkt tot 1280 pixels, zodat het niet breder wordt dan deze waarde. Door de eigenschap 'margin: 0 auto' toe te passen, worden gelijkmatige marges aan beide zijden van het element gecreëerd, wat resulteert in een horizontale centrering. Daarnaast definieert 'padding' de ruimte binnen het element.

One page

```
<section class="about-section" id="about">  
  ...  
<section class="contact-section" id="contact">
```

Om een website te maken met een one page heb ik een video bekeken, zie onderstaande bron. Om dit voor elkaar te krijgen heb ik gewerkt met <section>. Elke section heeft een andere class gekregen, zodat de lengte aangepast kan worden naar de hoeveelheid tekst. Elke section heeft ook een id gekregen, omdat deze verwijst naar de knopjes in de navigatiebalk. Hierdoor kan er geklikt worden op de navigatiebalk en wordt je naar de goede section gestuurd.

Bron: <https://www.youtube.com/watch?v=CxJUnPEb9bM>

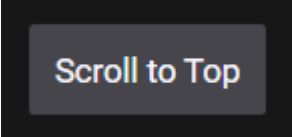
```
<nav class="navbar">  
  <ul>  
    <li><a href="index.html">Home</a></li>  
    <li><a href="music.html">Music</a></li>  
    <li><a href="photos.html">Photo's</a></li>  
    <li><a href="events.html">Events</a></li>  
    <li><a href="#about">About</a></li>  
    <li><a href="#contact">Contact</a></li>  
    <li><a href="shop.html">Shop</a></li>  
  </ul>  
</nav>
```

Op de homepagina heb ik in de navigatiebalk gebruikgemaakt van een href met '#sectienaam', waarbij 'sectienaam' verwijst naar de id van de section.

```
<nav class="navbar">  
  <ul>  
    <li><a href="index.html">Home</a></li>  
    <li><a href="music.html">Music</a></li>  
    <li><a href="photos.html">Photo's</a></li>  
    <li><a href="events.html">Events</a></li>  
    <li><a href="index.html#about">About</a></li>  
    <li><a href="index.html#contact">Contact</a></li>  
    <li><a href="shop.html">Shop</a></li>  
  </ul>  
</nav>
```

Op de andere pagina's heb ik in de navigatiebalk gebruikgemaakt van een href met 'index.html#sectienaam', waarbij 'sectienaam' verwijst naar de id van de section. Hierdoor wordt er op de homepagina direct naar de betreffende sectie gescrold.

Scroll to top



```
function scrollToTop() {  
    window.scrollTo(0, 0);  
}
```

De functie `scrollToTop` wordt gebruikt om naar de bovenkant van de pagina te scrollen. In JavaScript maakt deze functie gebruik van de `window.scrollTo`-methode, die het mogelijk maakt om naar een specifieke positie op de pagina te scrollen. Door de parameters van `scrollTo` op 0 horizontaal en 0 verticaal in te stellen, wordt er helemaal naar de bovenkant van de pagina gescrold.

```
<button id="scrollToTopBtn" class="scroll-to-top">Scroll to Top</button>  
  
var scrollToTopBtn = document.getElementById("scrollToTopBtn");
```

Deze regel code zoekt naar een HTML-element met de id `"scrollToTopBtn"` en slaat dit element op in de variabele `"scrollToTopBtn"`. Door het element aan de variabele `scrollToTopBtn` toe te wijzen, wordt het mogelijk om vervolgens met dit element te werken en er acties op toe te passen.

```
scrollToTopBtn.addEventListener("click", scrollToTop);
```

Deze regel code voegt een `"click"`-EventListener toe aan het element dat is opgeslagen in de variabele `"scrollToTopBtn"`. Dit betekent dat wanneer er op dat element wordt geklikt, een bepaalde functie genaamd `"scrollToTop"` wordt uitgevoerd.

Vertaalfunctie



Ik heb een vertaalfunctie ontwikkeld voor dit gedeelte van de homepage. De Nederlandse versie werkt goed. Bij de Engelse versie was er een probleem, waarbij de tekst versprong naar rechts.

```
<div class="languages">
  <li><a href="" onclick="setLanguage('nl')">  </a></li>
  <li><a href="" onclick="setLanguage('en')">  </a></li>
</div>
```

Om de vertaalfunctie te implementeren, heb ik onclick-events toegevoegd aan twee afbeeldingen. Als er wordt geklikt op de Nederlandse vlag, wordt de setLanguage-functie uitgevoerd met de waarde 'nl'. Als er wordt geklikt op de Engelse vlag, wordt de functie uitgevoerd met de waarde 'en'.

```
<div class="languages">
  <li><a id="nl-link" href=""></a></li>
  <li><a id="en-link" href=""></a></li>
</div>
```

Om de HTML en JavaScript beter van elkaar te scheiden, heb ik de onclick verwijderd en de afbeeldingen een id gegeven.

```
<section class="about-section" id="about">
  <div class="homepageText">
    <div class="column">
      <h1 class="nl">Wie ben ik?</h1>
      <p class="nl">OUMY is een Nederlandse singer-songwriter.
        Met haar intense, mysterieuze en vervreemdende stijl vertelt ze haar verhaal.
        Samen met haar zevenkoppige band neemt ze je mee op een mentale reis door haar verleden en verder.
        Haar muziek is geïnspireerd door filmmuziek, pop en wereldmuziek en brengt daarmee een nieuwe,
        filmische sound naar de muziekscene.</p>
    </div>

    <div class="column">
      <h1 class="en">Who am I?</h1>
      <p class="en" style="display: none;">OUMY is a Dutch singer-songwriter. Deep, dark and
        intense. With abstract but catchy melodies and deep diving instrumentals, OUMY takes you
        on a journey, back to her past and even further. A mental escape from reality. With wide
        influences ranging from film scores, pop and world music, OUMY and her seven headed
        band bring a new cinematic sounding experience to the music scene.</p>
    </div>

    
  </div>
</section>
```

Voor het vertaal gedeelte op de homepage heb ik twee verschillende classes gebruikt. Als de functie met de waarde 'nl' wordt uitgevoerd, wordt de Nederlandse tekst weergegeven en de Engelse tekst verborgen. Als de waarde 'en' wordt gebruikt, is het juist omgekeerd: de Engelse tekst wordt weergegeven en de Nederlandse tekst wordt verborgen.

Wie ben ik?

OUMY is een Nederlandse singer-songwriter. Met haar intense, mysterieuze en vervreemdende stijl vertelt ze haar verhaal. Samen met haar zevenkoppige band neemt ze je mee op een mentale reis door haar verleden en verder. Haar muziek is geïnspireerd door filmmuziek, pop en wereldmuziek en brengt daarmee een nieuwe, filmische sound naar de muzikscene.



Who am I?

OUMY is a Dutch singer-songwriter. Deep, dark and intense. With abstract but catchy melodies and deep diving instrumentals, OUMY takes you on a journey, back to her past and even further. A mental escape from reality. With wide influences ranging from film scores, pop and world music, OUMY and her seven headed band bring a new cinematic sounding experience to the music scene.



div.column 0 x 304.15

Who am I?

OUMY is a Dutch singer-songwriter. Deep, dark and intense. With abstract but catchy melodies and deep diving instrumentals, OUMY takes you on a journey, back to her past and even further. A mental escape from reality. With wide influences ranging from film scores, pop and world music, OUMY and her seven headed band bring a new cinematic sounding experience to the music scene.



Uit de bovenstaande afbeeldingen blijkt dat de Nederlandse en Engelse tekst niet op dezelfde positie worden weergegeven. Dit wordt veroorzaakt doordat de column-gap is ingesteld op 10%. Hierdoor verspringt de kolom bij de Engelse versie naar links. Als gevolg van de column-gap wordt de Engelse tekst meer naar rechts geplaatst.

```

.homepageText {
  display: flex;
  justify-content: center;
  flex-direction: row;
  margin: 7%;
  column-gap: 10%;
}

.column {
  display: flex;
  flex-direction: column;
}

```

→

```

.homepageText {
  display: flex;
  justify-content: center;
  flex-direction: row;
  margin: 0 auto;
  max-width: 1280px;
  padding: 100px 10px;
}

.photoOumyHome {
  margin-left: 160px;
}

.column {
  display: flex;
  flex-direction: column;
  padding-right: 20px;
}

```

Om dit probleem op te lossen, heb ik de column-gap verwijderd en vervolgens de 'margin' en 'padding' aangepast om de juiste uitlijning te verkrijgen.

```

<div class="photoOumyHome">
  
</div>

```

```

.photoOumyHome {
  margin-left: 160px;
}

```

Bovendien heb ik een marge toegevoegd aan de foto van Oumy, waardoor de tekst naar links is verschoven.

```

// Functie om de taalvoorkeur van de gebruiker op te slaan in een cookie
function setLanguage(language) {
  document.cookie = "language=" + language + "; path=/";
  updateContent();
}

document.getElementById("nl-link").addEventListener("click", function(event) {
  event.preventDefault();
  setLanguage("nl");
});

document.getElementById("en-link").addEventListener("click", function(event) {
  event.preventDefault();
  setLanguage("en");
});

```

De functie "setLanguage" is zorgt voor het aanmaken van een cookie waarin de geselecteerde taal wordt opgeslagen. De cookie wordt opgeslagen in de browser van de gebruiker met het path=/, waardoor deze beschikbaar is op de gehele website.

De functie "updateContent" zorgt ervoor dat de webpagina wordt bijgewerkt op basis van de geselecteerde taal. De code maakt gebruik van "document.getElementById" om een element op te halen op basis van de opgegeven id. Vervolgens wordt er een click-event aan dat element toegevoegd. Wanneer erop wordt geklikt, wordt het standaardgedrag gestopt om te voorkomen dat een link wordt gevolgd en een nieuwe pagina wordt geladen. Vervolgens wordt de functie "setLanguage" opgeroepen met de opgegeven waarden.

```
// Functie om de juiste taalversie van de inhoud te tonen op basis van de opgeslagen taal in het cookie
function updateContent() {
    var language = getLanguageFromCookie();
    var elements = document.getElementsByClassName(language);
    for (var i = 0; i < elements.length; i++) {
        elements[i].style.display = "block";
    }
    var elements = document.getElementsByClassName(language === "nl" ? "en" : "nl");
    for (var i = 0; i < elements.length; i++) {
        elements[i].style.display = "none";
    }
}
```

Bij het uitvoeren van de functie `updateContent()` wordt eerst de huidige taal opgehaald uit de cookie met behulp van `getLanguageFromCookie()`. Vervolgens worden alle elementen in het document geselecteerd die overeenkomen met deze taal-class, door gebruik te maken van `document.getElementsByClassName(language)`. Deze elementen worden opgeslagen in de variabele `elements`. Daarna wordt er een loop doorlopen om elk van de geselecteerde elementen te doorlopen. Voor elk element wordt de CSS-eigenschap `display` ingesteld op `'block'`, waardoor de elementen zichtbaar worden.

Daarna wordt een nieuwe set elementen geselecteerd met behulp van `document.getElementsByClassName(language === 'nl' ? 'en' : 'nl')`. Hier wordt gecontroleerd of de huidige taal `'nl'` is. Als dat het geval is, worden elementen met de class `'en'` geselecteerd. Als dat niet het geval is, worden de elementen met de class `'nl'` geselecteerd. Deze elementen worden opnieuw opgeslagen in de variabele `elements`. Vervolgens wordt er een nieuwe loop gestart om door deze elementen te lopen. Voor elk element wordt de CSS-eigenschap `display` ingesteld op `'none'`, waardoor de elementen verborgen worden.

```
// Functie om de opgeslagen taal uit het cookie op te halen
function getLanguageFromCookie() {
    var cookies = document.cookie.split("; ");
    for (var i = 0; i < cookies.length; i++) {
        var cookie = cookies[i].split("=");
        if (cookie[0] === "language") {
            return cookie[1];
        }
    }
    return "nl"; // Standaardtaal is Nederlands
}
```

De functie `getLanguageFromCookie` haalt de cookie-informatie op via `document.cookie`. Deze informatie wordt vervolgens gesplitst op basis van het scheidingsteken `' '` om een array van cookies te verkrijgen. Daarna wordt er een loop gestart om elk cookie in de array te doorlopen. Voor elke cookie wordt de cookie-informatie verder gesplitst met behulp van het `'='` teken om de naam en de waarde van de cookie te verkrijgen. Binnen de loop wordt gecontroleerd of de naam van de cookie overeenkomt met `'language'`. Wanneer de cookie met de naam `'language'` wordt gevonden, wordt de waarde geretourneerd als resultaat van de functie. Als er geen cookie wordt gevonden met de naam `'language'` of als er geen cookies aanwezig zijn, wordt standaard `'nl'` geretourneerd.

```
// Roep de updateContent functie aan bij het laden van de pagina om de juiste taalversie van de inhoud te tonen
window.onload = function() {
    updateContent();
}
```

Wanneer de pagina geladen wordt, wordt automatisch de functie `updateContent()` geactiveerd. Hierdoor is de taal automatisch in de geselecteerde taal te zien als er wordt doorgedrukt of opnieuw wordt bezocht.

Event



Ik vond het lastig om een eventblok te maken met deze opmaak.

```
<div class="center">
  
  <div class="info">
    <div class="date">Zaterdag 20 mei 2023</div>
    <div class="event">OUMY</div>
    <div class="location">Rotterdam, Rotown</div>
  </div>
  <div class="buttonEventsPosition">
    <button class="buttonEvents">Event info</button>
    <button class="buttonEvents">Koop Tickets</button>
  </div>
</div>
```

Dit stukje code is bedoeld om een gestructureerd blok weer te geven met een afbeelding, informatie en twee knoppen in een specifieke lay-out.

```
.center {
  margin: 0 auto;
  width: 900px;
  height: 186px;
  position: relative;
  padding: 20px;
  border: 1px solid #ghostwhite;
  margin-top: 20px;
  border-radius: 2px;
}
```

Ik heb een div gemaakt met de class 'center' om alle evenementen op dezelfde manier weer te geven. Deze div heeft een witte rand eromheen, zodat het duidelijk is waar de informatie van elk evenement begint en eindigt.

```
.imageEvent {
  position: absolute;
  top: 0;
  left: 0;
  height: 100%;
  width: auto;
}
```

Om de afbeelding aan de linkerkant van de div te plaatsen, gebruikte ik de CSS-regel "position: absolute". Met "top: 0" en "left: 0" wordt de afbeelding helemaal linksboven in de div geplaatst. "Height: 100%" zorgt ervoor dat de afbeelding de volledige hoogte van de div inneemt zonder witruimte. "Width: auto" behoudt de oorspronkelijke verhouding van de afbeelding, waardoor deze niet wordt vervormd.

```
.info {  
  margin-left: 350px;  
  position: absolute;  
  top: 50%;  
  transform: translate(0, -50%);  
  color: ■ ghostwhite;  
  font-family: 'Roboto', 'Arial', 'Verdana', 'Times New Roman';  
}
```

Om ervoor te zorgen dat de informatie in het midden van de div wordt weergegeven, is er een div met class "info" toegevoegd. Hierbij is de "top" eigenschap ingesteld op 50%, zodat de informatie verticaal in het midden wordt geplaatst. Vervolgens wordt de informatie met behulp van "transform" op 50% van de hoogte gepositioneerd, waardoor er zowel boven als onder evenveel ruimte ontstaat.

```
.buttonEventsPosition {  
  position: absolute;  
  bottom: 0;  
  right: 0;  
}
```

Tot slot is er een div met de class 'buttonEventsPosition' toegevoegd. Om ervoor te zorgen dat deze rechtsonder wordt geplaatst, zijn de bottom- en right-waarden ingesteld op 0. In deze div zitten de buttons met class buttonEvents.

Contactformulier

Contact

Blijf op de hoogte!

Wil je op de hoogte blijven van mijn muziek en events? Meld je dan nu aan!

127.0.0.1:5501 meldt het volgende

Bedankt voor je aanmelding!
Thank you for your registration!

Ik heb ervoor gekozen om het contactformulier met JavaScript op te stellen in plaats van PHP. Mijn voornaamste reden hiervoor was dat ik mijn vaardigheid in JavaScript wil ontwikkelen voordat ik me ga verdiepen in PHP. Ik heb gemerkt dat ik het werken met PHP uitdagend vind. Tijdens de presentatie wilde ik laten zien dat het contactformulier functioneel leek te werken met behulp van JavaScript.

```
<div class="contactpage">
  <form id="contact-form" method="post">
    <div class="textContactForm">
      <h1>Blijf op de hoogte!</h1>
      <p>Wil je op de hoogte blijven van mijn muziek en events? Meld je dan nu aan!</p>
    </div>

    <input type="text" id="name" name="name" placeholder="Naam" required>
    <input type="email" id="email" name="email" placeholder="E-mail" required>
    <button type="submit" id="submit-btn">Aanmelden</button>
  </form>
</div>
```

Het contactformulier heeft een id 'contact-form' gekregen, zodat de Javascript hierop kan worden toegepast. Het form heeft als methode post, zodat het formulier als HTTP POST -verzoek verstuurd kan worden. Wanneer je een POST-verzoek uitvoert, stuurt de client (bijvoorbeeld een webbrowser) gegevens naar de server, die vervolgens deze gegevens verwerkt. Het verschil met de "GET"-methode is dat bij een POST-verzoek de gegevens in de aanvraag worden opgenomen, in plaats van in de URL. Door bij input required te zetten, zorg ik ervoor dat het formulier alleen verzonden kan worden als alle gegevens zijn ingevuld.

Het "text"-type is een invoertype in HTML dat wordt gebruikt voor het invoeren van tekstuele gegevens. De browser zal een enkele regel tekstinvoer weergeven waar gebruikers tekst kunnen typen. Dit is het standaardtype voor input-elementen als er geen specifiekere invoertypes worden gespecificeerd.

Het type "email" is een invoertype in HTML dat wordt gebruikt voor het invoeren van e-mailadressen. Wanneer je het attribuut "type" van een input-element instelt op "email", zal de browser automatisch een aantal validatieregels toepassen om te controleren of het ingevoerde waardepatroon overeenkomt met een geldig e-mailadres. Dit omvat onder andere het controleren op de aanwezigheid van een "@"-symbool en een domeinnaam.

Het "submit"-type is een invoertype in HTML dat wordt gebruikt om een formulier te verzenden naar de server voor verwerking. De browser zal de bijbehorende knop weergeven en, wanneer erop wordt geklikt, het bijbehorende formulier verzenden. Dit stuurt de ingevulde gegevens naar de server voor verdere verwerking of opslag.

```
// Selecteer het formulier
const form = document.querySelector('#contact-form');

// Voeg een event listener toe voor het indienen van het formulier
form.addEventListener('submit', (e) => {
  e.preventDefault();

  // Valideer het formulier
  if (validateForm()) {
    // Verwerk het formulier
    processForm();
  }
});
```

De eerste regel selecteert het formulier op basis van het id-attriboot "contact-form" en slaat dit op in de variabele "form". Vervolgens wordt er een "submit"-EventListener toegevoegd aan het formulier. Dit betekent dat wanneer de gebruiker op de knop drukt om het formulier in te dienen, de code binnen de eventlistener wordt uitgevoerd. De eerste actie binnen de EventListener is het gebruik van de "e.preventDefault()" methode om het standaardgedrag van het formulier te voorkomen. Normaal gesproken zou het formulier de pagina vernieuwen of naar een andere pagina gaan, maar met deze code wordt dat voorkomen. Daarna wordt de functie "validateForm()" opgeroepen om het formulier te valideren. Dit betekent dat er wordt gecontroleerd of alle vereiste velden zijn ingevuld en of de ingevoerde gegevens aan bepaalde criteria voldoen. Als het formulier succesvol is gevalideerd, wordt de functie "processForm()" opgeroepen om het formulier te verwerken.

```
function validateForm() {
  let isValid = true;

  // Controleer of de naam is ingevuld
  const name = document.querySelector('#name');
  if (!name.value.trim()) {
    isValid = false;
    name.classList.add('invalid');
  } else {
    name.classList.remove('invalid');
  }

  // Controleer of de e-mail geldig is
  const email = document.querySelector('#email');
  const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
  if (!email.value.trim() || !emailRegex.test(email.value.trim())) {
    isValid = false;
    email.classList.add('invalid');
  } else {
    email.classList.remove('invalid');
  }

  return isValid;
}
```

De variabele "isValid" wordt gebruikt om bij te houden of het formulier geldige gegevens bevat. De code bevat twee validatiecontroles, één voor de naam en één voor het e-mailadres.

Voor de naam wordt gecontroleerd of het veld is ingevuld. Als het veld leeg is, wordt de variabele "isValid" op false gezet en wordt de waarde "invalid" toegevoegd aan het element met id "name". Anders wordt de waarde "invalid" verwijderd.

Voor het e-mailadres wordt gecontroleerd of het veld leeg is of niet voldoet aan het e-mail patroon. Het e-mail patroon wordt gedefinieerd met behulp van een reguliere expressie (regex). Een reguliere expressie (regex) is een speciale sequentie van tekens die wordt gebruikt om patronen te definiëren en te matchen in teksten. Als het veld leeg is of niet overeenkomt met het patroon, wordt de variabele "isValid" op false gezet en wordt de waarde "invalid" toegevoegd aan het element met het id "email". Anders wordt de waarde "invalid" verwijderd.

Aan het einde van de functie wordt de waarde van "isValid" teruggegeven. Dit geeft aan of het formulier geldige gegevens bevat (true) of niet (false).

```
// Verwerk het formulier
function processForm() {
    alert('Bedankt voor je aanmelding!\nThank you for your registration!');
}
```

Binnen deze functie kun je de gewenste acties uitvoeren, zoals het verzenden van de ingevulde gegevens naar een server of het uitvoeren van een andere gewenste actie. Hier wordt een pop-up bericht weergegeven met de tekst "Bedankt voor je aanmelding! Thank you for your registration!". Dit bericht wordt weergegeven nadat het formulier succesvol is ingediend.

Video YouTube

Video insluiten

>

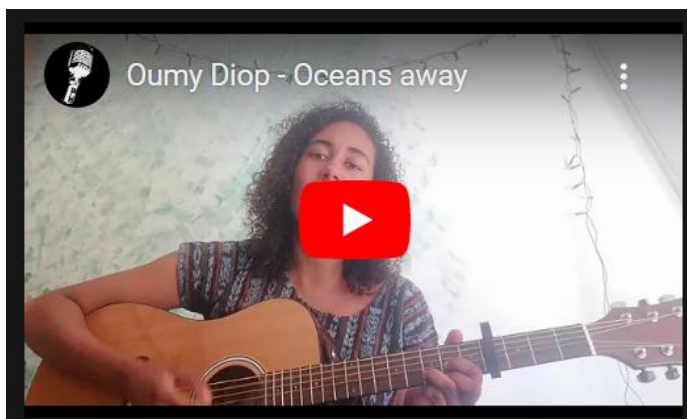
```
<iframe width="560" height="315"
src="https://www.youtube.com/embed/
-SMP32qfrdE" title="YouTube video
player" frameborder="0"
allow="accelerometer; autoplay;
clipboard-write; encrypted-media;
gyroscope; picture-in-picture; web-
share" allowfullscreen></iframe>
```

☐ Beginnen bij 0:00

OPTIES VOOR INSLUITEN

☒ Afspeelbediening tonen.

☐ Privacyinstellingen aanpassen



```
<iframe width="426" height="240" src="https://www.youtube.com/embed/-SMP32qfrdE"
title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write;
encrypted-media; gyroscope; picture-in-picture; web-share" allowfullscreen></iframe>
```

Via YouTube kun je de code krijgen om een YouTube video in te sluiten op je eigen website. De width en height heb ik aangepast naar mijn gewenste grootte, hierbij heb ik rekening gehouden met de verhouding van de video.

Klikbare afbeelding



```
// Selecteer alle afbeeldingselementen en de modale elementen
var afbeeldingen = document.querySelectorAll(".containerPhotoPage img");
var modaal = document.getElementById("vergroteAfbeelding");

// Selecteer de vergrote afbeelding en de sluitknop in het modale element
var vergroteAfbeelding = document.getElementById("vergroteAfbeeldingImg");
var sluitknop = document.getElementById("sluitknop");

// Voeg een klikgebeurtenis toe aan alle afbeeldingselementen
for (var i = 0; i < afbeeldingen.length; i++) {
  afbeeldingen[i].addEventListener("click", function() {
    modaal.style.display = "block";
    vergroteAfbeelding.src = this.src;
  });
}

// Voeg een klikgebeurtenis toe aan de sluitknop in het modale element
sluitknop.addEventListener("click", function() {
  modaal.style.display = "none";
});
```

Deze code zorgt ervoor dat wanneer je op een afbeelding klikt, deze wordt geopend met een vergrote versie van de afbeelding. De variabele "afbeeldingen" selecteert alle afbeeldingselementen binnen een element met de class "containerPhotoPage". De variabele "modaal" verwijst naar het element dat wordt gebruikt als het modale venster (pop-upvenster). Daarna worden de vergrote afbeelding en de sluitknop binnen het modale venster geselecteerd met de variabelen "vergroteAfbeelding" en "sluitknop". Een klikgebeurtenis wordt toegevoegd aan elk afbeeldingselement met behulp van een for-loop. Wanneer er op een afbeelding wordt geklikt, wordt het modale venster weergegeven en wordt de bron van de vergrote afbeelding ingesteld op de bron van het geklikte afbeeldingselement. Er wordt ook een klikgebeurtenis toegevoegd aan de sluitknop in het modale venster. Wanneer er op de sluitknop wordt geklikt, wordt het modale venster verborgen.