Ref. Galeria

<https://www.henriheymans.com/> - Focado na galeria

**Relatório Técnico Detalhado: Galeria de Imagens Popup na Seção "Recognitions & awards" do Site Henri Heymans**

**1. Descrição do Efeito Visual:**

* **Seção "Recognitions & awards":** Nesta seção, as imagens são dispostas em uma grade ou lista vertical, representando os reconhecimentos e prêmios de Henri Heymans.
* **Trigger do Popup:** Conforme você rola a página verticalmente através desta seção, ao passar (scroll) por cada "tópico" (cada item de reconhecimento/prêmio que contém uma imagem), um popup surge no canto inferior direito da tela.
* **Popup de Imagem:** O popup que aparece exibe a imagem em destaque do tópico que você está visualizando no momento.
* **Animação de Exibição do Popup:** O popup surge com uma animação suave, deslizando ou "aparecendo" no canto inferior direito.
* **Animação de Passagem de Imagem (Ao Scrollar):** Ao continuar rolando pela seção e passar para o próximo tópico com imagem, o popup atualiza a imagem exibida, novamente com uma animação de transição suave. A animação de transição de imagem dentro do popup também é um ponto chave.
* **Desaparecimento do Popup:** Quando você rola para fora da seção "Recognitions & awards" (tanto para cima quanto para baixo), o popup desaparece, também com uma animação suave de saída.

**2. Implementação Técnica Detalhada: Código e Animação**

Para entender a implementação, vamos decompor os componentes e o fluxo de animação:

**2.1. Estrutura HTML:**

* **Seção "Recognitions & awards" (Container Principal):** Esta seção é um container HTML que engloba todos os tópicos de reconhecimentos e prêmios. Poderia ser um <section> com um ID ou classe específica (ex: <section id="recognitions-awards"> ou <section class="awards-section">).
* **Tópicos Individuais (Items):** Cada reconhecimento ou prêmio é um item individual dentro da seção principal. Cada item provavelmente é um container (ex: <div>, <article>) com uma classe para identificá-los (ex: <div class="award-item">).
* **Imagens nos Tópicos:** Dentro de cada item, há um elemento <img> que representa a imagem em destaque do reconhecimento/prêmio.
* **Container do Popup (Inicialmente Oculto):** Fora da seção "Recognitions & awards", em algum lugar no HTML (geralmente no final do body), há um container HTML para o popup. Este container é **inicialmente oculto** por CSS (display: none; ou opacity: 0;). Poderia ser uma <div> com um ID ou classe específica (ex: <div id="image-popup-container" class="image-popup hidden">).
* **Imagem Dentro do Popup:** Dentro do container do popup, há um elemento <img> que será dinamicamente atualizado com a imagem do tópico atualmente em foco. (ex: <img id="popup-image" src="" alt="Imagem do Reconhecimento">).

**Estrutura HTML Conceitual (Simplificada):**

HTML

<section id="recognitions-awards">

<div class="award-item" data-image-src="caminho/para/imagem1.jpg">

<img src="caminho/para/imagem-miniatura-1.jpg" alt="Miniatura do Reconhecimento 1">

</div>

<div class="award-item" data-image-src="caminho/para/imagem2.jpg">

<img src="caminho/para/imagem-miniatura-2.jpg" alt="Miniatura do Reconhecimento 2">

</div>

</section>

<div id="image-popup-container" class="image-popup hidden">

<img id="popup-image" src="" alt="Imagem Popup">

</div>

* **data-image-src:** Observe o uso de um atributo data-image-src nos div.award-item. Este atributo é **fundamental**. Ele armazena o caminho para a **imagem de alta resolução** que será exibida no popup. A miniatura da imagem (<img src="caminho/para/imagem-miniatura-1.jpg">) é exibida diretamente no tópico, enquanto a imagem de alta resolução é carregada no popup quando necessário.
* **.hidden:** A classe .hidden no div#image-popup-container é usada para inicialmente ocultar o popup com CSS (ex: display: none; ou opacity: 0;).

**2.2. Animação CSS:**

* **Animação de Entrada/Saída do Popup:** A animação para fazer o popup aparecer e desaparecer provavelmente é feita com **CSS Transitions**. As propriedades CSS que podem ser animadas para criar o efeito de "slide-in" ou "fade-in" incluem:
  + opacity: Para um efeito de "fade-in" (aparecer gradualmente).
  + transform: translateY() ou transform: translateX(): Para um efeito de "slide-in" (deslizar de uma direção).
  + visibility: Para controlar a visibilidade (em conjunto com opacity ou transform).

**Exemplo de CSS para Animação de Entrada/Saída (Conceitual):**

CSS

.image-popup-container {

position: fixed; /\* Posicionamento fixo para o popup no canto inferior direito \*/

bottom: 20px;

right: 20px;

background-color: #fff; /\* Fundo do popup \*/

box-shadow: 0 0 10px rgba(0, 0, 0, 0.2); /\* Sombra \*/

border-radius: 5px;

overflow: hidden; /\* Para garantir que a imagem não vaze do container \*/

width: 300px; /\* Largura do popup \*/

height: auto; /\* Altura automática \*/

opacity: 0; /\* Inicialmente invisível \*/

transform: translateY(50px); /\* Inicialmente um pouco abaixo da posição final \*/

transition: opacity 0.3s ease-out, transform 0.3s ease-out; /\* Animação suave para opacity e transform \*/

pointer-events: none; /\* Para que o popup não bloqueie interações com elementos abaixo dele quando invisível \*/

}

.image-popup-container.visible {

opacity: 1; /\* Visível quando a classe 'visible' é adicionada \*/

transform: translateY(0); /\* Posição final (sem deslocamento vertical) \*/

pointer-events: auto; /\* Permite interações com o popup quando visível (se necessário) \*/

}

.image-popup-container.hidden {

opacity: 0; /\* Invisível quando a classe 'hidden' está presente \*/

transform: translateY(50px); /\* Volta para a posição inicial (deslocada para baixo) \*/

pointer-events: none; /\* Desativa interações quando escondido \*/

transition: opacity 0.3s ease-in, transform 0.3s ease-in; /\* Animação de saída (pode ser diferente da entrada) \*/

}

.image-popup-container img {

display: block; /\* Garante que a imagem ocupe toda a largura do container \*/

width: 100%;

height: auto;

}

* + **.image-popup-container.hidden e .image-popup-container.visible:** Classes CSS que controlam os estados "escondido" e "visível" do popup, manipulando opacity e transform para criar a animação. O JavaScript irá adicionar/remover essas classes para controlar a visibilidade.
  + **transition:** Define as propriedades que serão animadas e a duração e easing da animação.
  + **pointer-events: none; e pointer-events: auto;:** Importante para controlar se o popup interfere com eventos de mouse em elementos abaixo dele quando está visível ou escondido.
* **Animação de Transição de Imagem Dentro do Popup:** Para a transição suave entre as imagens no popup (ao rolar pelos tópicos), podem ser usadas técnicas como:
  + **Cross-fade com Opacity:** Quando uma nova imagem precisa ser exibida:
    1. A imagem atual no popup tem sua opacity animada para 0 (fade-out).
    2. Assim que a opacidade se aproxima de zero, a src da imagem no popup é atualizada para a nova imagem.
    3. A nova imagem tem sua opacity animada de 0 para 1 (fade-in).
  + **Transição de Transform (Scale ou Slide):** Similar ao cross-fade, mas usando transform: scale() (para zoom in/out) ou transform: translateX() (para slide horizontal) para animar a transição entre as imagens.
  + **Implementação com CSS Transitions e JavaScript:** A troca de imagens e as animações de transição dentro do popup provavelmente são orquestradas com JavaScript, que manipula as classes CSS ou estilos inline dos elementos de imagem para disparar as transições CSS.

**2.3. Código JavaScript (Lógica e Animação):**

* **Detecção do Scroll e Elementos Visíveis na Viewport:** O JavaScript precisa monitorar o evento de scroll da janela e detectar quando cada item da seção "Recognitions & awards" entra e sai da viewport (área visível da tela do navegador). A **Intersection Observer API** é ideal para isso.
  + **Intersection Observer API:** Permite observar quando um elemento HTML entra ou sai da viewport, ou cruza um determinado limiar de visibilidade dentro de outro elemento (ou da viewport). É muito eficiente para esse tipo de efeito baseado em scroll.
  + **Código JavaScript Conceitual (Usando Intersection Observer):**

JavaScript

document.addEventListener('DOMContentLoaded', function() { // Garante que o script roda após o DOM ser carregado

const awardItems = document.querySelectorAll('.award-item'); // Seleciona todos os itens de reconhecimento

const popupContainer = document.getElementById('image-popup-container');

const popupImage = document.getElementById('popup-image');

const observer = new IntersectionObserver(entries => { // Cria um Intersection Observer

entries.forEach(entry => { // Para cada item observado

if (entry.isIntersecting) { // Se o item está entrando na viewport

const imageSrc = entry.target.dataset.imageSrc; // Obtém o caminho da imagem do atributo data-image-src

if (imageSrc) { // Verifica se o atributo data-image-src existe

popupImage.src = imageSrc; // Atualiza a imagem no popup

popupContainer.classList.add('visible'); // Mostra o popup (animação de entrada via CSS)

popupContainer.classList.remove('hidden'); // Remove a classe 'hidden' para garantir que 'visible' tenha efeito

}

} else { // Se o item está saindo da viewport (pode ser para cima ou para baixo)

// Verifica se é o item que ATIVOU o popup (pode ser necessário rastrear o item ativo)

// Para evitar fechar o popup muito cedo ao rolar um pouco para cima/baixo dentro da seção

// (Lógica mais complexa pode ser necessária para um comportamento perfeito)

// Exemplo Simplificado: Sempre que um item sai da viewport, esconde o popup

popupContainer.classList.remove('visible'); // Esconde o popup (animação de saída via CSS)

popupContainer.classList.add('hidden'); // Adiciona 'hidden' para garantir que 'visible' seja removido

popupImage.src = ''; // Limpa a imagem no popup (opcional)

}

});

}, {

threshold: 0.5, // Dispara o callback quando 50% do elemento estiver visível na viewport (ajuste conforme necessário)

rootMargin: '0px 0px 0px 0px' // Margem ao redor da viewport de observação (padrão: viewport inteira)

});

awardItems.forEach(item => { // Observa cada item "award-item"

observer.observe(item);

});

});

* + - **IntersectionObserver:** Cria um novo observador de interseção.
    - **entries.forEach:** Itera sobre as "entries" (informações sobre cada item observado que teve uma mudança de interseção).
    - **entry.isIntersecting:** Verifica se o item está atualmente intersectando a viewport (ou seja, está visível).
    - **entry.target.dataset.imageSrc:** Acessa o valor do atributo data-image-src do elemento DOM que está sendo observado (o div.award-item).
    - **popupImage.src = imageSrc;:** Atualiza o atributo src da imagem dentro do popup com o caminho da imagem obtido do data-image-src.
    - **popupContainer.classList.add('visible'); e popupContainer.classList.remove('hidden');:** Adiciona a classe visible e remove hidden para mostrar o popup e ativar a animação de entrada via CSS.
    - **popupContainer.classList.remove('visible'); e popupContainer.classList.add('hidden');:** Remove visible e adiciona hidden para esconder o popup e ativar a animação de saída.
    - **observer.observe(item);:** Inicia a observação de cada elemento award-item pelo Intersection Observer.
    - **threshold: 0.5:** O callback do observer é disparado quando 50% do elemento observado se torna visível na viewport. Ajuste este valor conforme necessário para controlar o momento exato em que o popup aparece.

**3. Bibliotecas e Frameworks:**

* **Intersection Observer API:** É uma API nativa do navegador, **não requer bibliotecas externas**. É a forma mais eficiente e recomendada para detectar a visibilidade de elementos na viewport em relação ao scroll.
* **Bibliotecas de Animação JavaScript (Opcional, mas menos provável para este efeito específico):** Bibliotecas como **GreenSock (GSAP)** ou **Anime.js** poderiam ser usadas para criar animações mais complexas para o popup ou a transição de imagens, mas para este efeito, CSS Transitions e a Intersection Observer API são suficientes e mais performáticas. Frameworks JavaScript como **React, Vue, Angular** poderiam ser usados para estruturar o componente do popup e gerenciar o estado, mas também não são estritamente necessários para a funcionalidade básica.

**4. Resumo do Fluxo de Animação:**

1. **Inicialização:** A página carrega, e o popup está inicialmente oculto (classe .hidden aplicada via CSS). O Intersection Observer é configurado para monitorar os itens da seção "Recognitions & awards".
2. **Scroll para a Seção:** O usuário rola a página para baixo e a seção "Recognitions & awards" entra na viewport.
3. **Item Entra na Viewport:** Quando um item (.award-item) da seção entra na viewport (50% visível, por exemplo, configurado no threshold do Intersection Observer):
   * O Intersection Observer dispara o callback.
   * O JavaScript obtém o caminho da imagem de alta resolução do atributo data-image-src do item que entrou na viewport.
   * O JavaScript atualiza o src do elemento <img> dentro do popup com este novo caminho de imagem.
   * O JavaScript adiciona a classe .visible e remove .hidden do container do popup. Isso dispara a animação CSS de entrada do popup (slide-in/fade-in).
4. **Transição de Imagem (Opcional):** Se uma animação de transição de imagem (cross-fade, slide, etc.) for implementada (ex: com manipulação de opacity ou transform e transitions CSS), ela ocorre neste momento, logo após a atualização do src da imagem no popup.
5. **Scroll Continua na Seção:** Ao continuar rolando dentro da seção "Recognitions & awards" e passar para o próximo item, o processo se repete: o Intersection Observer detecta a entrada do novo item, o popup atualiza a imagem (com animação de transição opcional) para a imagem do novo item.
6. **Scroll Sai da Seção:** Quando o usuário rola para fora da seção "Recognitions & awards" (para cima ou para baixo), de modo que nenhum item da seção esteja mais na viewport:
   * O Intersection Observer dispara o callback para o último item que estava visível (agora não mais intersectando).
   * O JavaScript remove a classe .visible e adiciona .hidden ao container do popup. Isso dispara a animação CSS de saída do popup (slide-out/fade-out), escondendo-o suavemente.
   * (Opcional) O JavaScript pode limpar o src da imagem no popup (popupImage.src = '';) para liberar recursos ou garantir que nenhuma imagem residual permaneça no popup escondido.

<https://by-kin.com/work/dentons> - Focado na galeria

G6st6 da Galeria

https://photoyoshi.com/landscape/ - Focado na galeria

Referencia para visualizador de imagens:

https://www.awwwards.com/sites/takamitsu-motoyoshi

Site que faz uma analise sobre o site:

https://photoyoshi.com/

https://www.awwwards.com/inspiration/art-direction-and-photograpy-portfolio

Site que faz uma analise sobre o site:

https://saintlouvent.com/photography

Referencias para portifólio:

https://www.awwwards.com/sites/cris-araujo

Site que faz uma analise sobre o site:

https://cristianaaraujo.com/

https://www.awwwards.com/inspiration/portfolio-of-richard-ekwonye-creative-developer

Site que faz uma analise sobre o site:

https://www.richardekwonye.com/

https://www.awwwards.com/inspiration/home-page-horizontal-scroll-emma-is-social

Site que faz uma analise sobre o site:

https://emmaissocial.com/

https://www.awwwards.com/sites/anatole-touvron-portfolio

Site que faz uma analise sobre o site:

https://anatoletouvron.fr/project/keepgrading/

https://www.isabelmoranta.com/