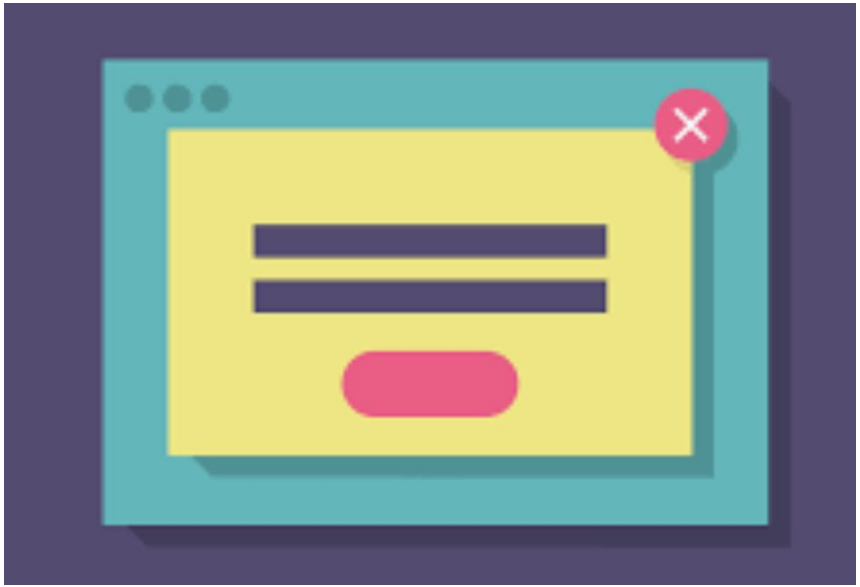
CSS Advance

Ventana modal (Popup)

Edición Web I

Ventana emergente

Ventana oculta que aparece por delante de la web cuando presionamos un botón.



Edición Web I
Jonathan Faus

1. Crear el botón que abre el popup
2. Crear la ventana emergente (div)

#popu1: tanto el botón como el popup tienen que tener el mismo ID.

- Al primer **div id="popup"** le asignaremos un valor fixed, es la capa que se superpone a la web con un color semitransparente y estará por detrás del popup.
- El segundo div es la ventana modal que creamos.




```
<a class="open-btn" href="#popup1">open</a>


<!-- POPUP -->
<div id="popup1">

    <div>
        <a href="#">close</a>
        <h4><strong>titulo</strong>
        <p>aquí mi contendio...</p>
    </div>

</div>
```

.overlay: definimos el estilo de la capa semitransparente:

Position: fixed
Necesitamos que sea fixed para que no se mueva con el scroll, ponemos todos los parámetros de posición a 0 (top,bottom, left, right).

Background semitransparente para que veamos un poco la web que se escondo por detrás de esta capa.

Visibility: hidden este parámetro hace que no sea visible.




```css
.overlay {
    position: fixed;
    top: 0;
    bottom: 0;
    left: 0;
    right: 0;
    background: rgba(0, 0, 0, 0.7);
    transition: 0.5s;
    visibility: hidden;
    opacity: 0;
  }
  .overlay:target {
    visibility: visible;
    opacity: 1;
  }
```

```html
<a class="open-btn" href="#popup1">open</a>


<!-- POPUP -->
<div id="popup1" class="overlay">

    <div>
        <a href="#">close</a>
        <h4><strong>titulo</strong>
        <p>aquí mi contendio...</p>
    </div>

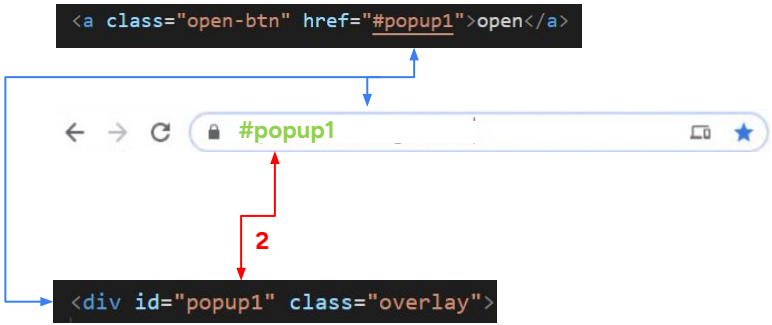
</div>
```

Edición Web I
Jonathan Faus

Pseudoclase :target

Nos permite que se ejecute el código de la pseudoclase cuando en la barra del navegador coincide el ID con el ID del elemento html que tiene la clase principal.

1. Al pulsar el botón le decimos que vaya al id **#popup1** y este se carga en la barra de navegación



2. Cuando coincide el ID del elemento html y del navegador se ejecuta la pseudoclase **overlay: target**

!!! Overlay no tiene visibiliad... pero como cargamos el código de overlay:target se le asignamos visibiliad cunado pulsamos el botón.

```css
.overlay {
    position: fixed;
    top: 0;
    bottom: 0;
    left: 0;
    right: 0;
    background: rgba(0, 0, 0, 0.7);
    transition: 0.5s;
    visibility: hidden;
    opacity: 0;
  }
  .overlay:target {
    visibility: visible;
    opacity: 1;
  }
```

```html
<a class="open-btn" href="#popup1">open</a>

<!-- POPUP -->
<div id="popup1" class="overlay">

    <div>
        <a href="#">close</a>
        <h4><strong>titulo</strong>
        <p>aquí mi contendio...</p>
    </div>

</div>
```

Transición entre hidden y visible

Visibility: hidden hace que el objeto no sea visible.

Hacer una transición entre el estado hidden y el estado visible **NO ES POSIBLE**, por eso ponemos ***opacity:0*** y ***opacity:1***... porque el parámetro opacity si nos pemite hacer una transición.

```css
.overlay {
    position: fixed;
    top: 0;
    bottom: 0;
    left: 0;
    right: 0;
    background: rgba(0, 0, 0, 0.7);
    transition: 0.5s;
    visibility: hidden;
    opacity: 0;
  }
  .overlay:target {
    visibility: visible;
    opacity: 1;
  }
```

```html
<a class="open-btn" href="#popup1">open</a>

<!-- POPUP -->
<div id="popup1" class="overlay">

    <div>
        <a href="#">close</a>
        <h4><strong>titulo</strong>
        <p>aquí mi contendio...</p>
    </div>

</div>
```

Ventana Popup

Definimos tamaño, color y demás estilos.

Truco: con **margin:150px auto** centramos la ventana
También podemos usar Flex

Añadimos una transición de 0,5s

Position:relative porque luego el botón de cerrar lo pondremos de modo absoluto.

Botón Close

Colocamos el botón de forma absoluta.
Le damos estilo al link (color, tamaño...)

En el parámetro href del enlace ponemos # o cualquier ID diferente.

Como la **#** es diferente a **#popup1** los valores de la pseudoclase *.overlay:Target* desaparecen y vuelve a lo valores iniciales de la clase *.overlay* con el parámetro **visibility:hidden**

```css
.popup {

  margin: 150px auto;
  padding: 20px;
  background: #fff;
  border-radius: 5px;
  width: 30%;
  max-height: 30%;
  position: relative;
  transition: 0.5s;
}

.popup h2 {
  margin-top: 0;
  color: #333;
  font-family: Tahoma, Arial, sans-serif;
}
```

```css
.close {
  position: absolute;
  top: 20px;
  right: 30px;
  font-size: 16px;
  font-weight: bold;
  text-decoration: none;
  color: #000000;
}
close:hover {
  color: #d80606;
}
```

```html
<div id="popup1" class="overlay">
    <div class="popup">
        <a class="close" href="#">close</a>
        <h4>titulo</h4>
        <p>aquí mi contendio...</p>
    <div>

</div>
```

Añadir una animación independiente a la ventana popup

Cuando cargamos la ventana aparece con una transición de opacidad que hemos creado entre .overlay y .overlay:target.

Como en el Html el div de la ventana está dentro del overlay (overlay es el padre) la transición afecta a todo el contenido:

- Overlay
- ventana



¿Cómo añadimos una transición independiente a la ventana?

```css
.overlay {
    position: fixed;
    top: 0;
    bottom: 0;
    left: 0;
    right: 0;
    background: rgba(0, 0, 0, 0.7);
    transition: 0.5s;
    visibility: hidden;
    opacity: 0;
}
.overlay:target {
    visibility: visible;
    opacity: 1;
}
```

```html
<!-- POPUP -->
<div id="popup1" class="overlay">

    <div>
        <a href="#">close</a>
        <h4><strong>titulo</strong>
        <p>aquí mi contendio...</p>
    </div>

</div>
```

La transición de opacidad afecta a overlay y a la ventana.

Ahora tenemos que crear una transición solo para la ventana, y esta transición se ha de efectuar cuando la ventana esté visible.

Usamos selectores descendientes:

.overlay:target .popup { código transición }

Este código solo afectará a la ventana moviéndola verticalmente.




```css
.overlay:target .popup {
  transition: 0.5s;
  transform: translateY(150px);
}
```

```html
<!-- POPUP -->
<div id="popup1" class="overlay">
    <div class="popup">
        <a class="close" href="#">close</a>
        <h4>titulo</h4>
        <p>aquí mi contendio...</p>
    <div>
</div>
<!-- end POPUP -->
```