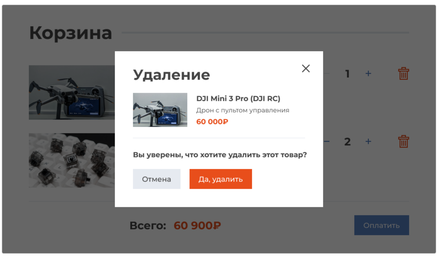
Вёрстка модального окна

Модальное окно — это элемент страницы, с которым взаимодействует пользователь: вводит какую-либо информацию, что-то выбирает и так далее. Модальное окно обычно выводится по центру окна браузера. При этом часто остальной контент блокируется полупрозрачным или заблюренным слоем. Например, как на изображении ниже.



Модальное окно для подтверждения удаления товара

Разбирать вёрстку традиционно начнём с разметки. В документе модальное окно обычно размещается в самом конце перед закрывающим тегом `<body>`. Для полупрозрачной подложки используют тег `<div>` с классом `modal`, само окно заворачивают в тег `<div>` с классом `modal-box`.

```
<body>
  <header>
  ...
  </header>
  <main>
  ...
  </main>
  <footer>
  ...
  </footer>
  <div class="modal">
    <div class="modal-box">
      <button class="modal-close-button">Закрыть</button>
      <section class="modal-content">
        <h3 class="modal-caption">Удаление</h3>
        <form class="form" action="#" method="post">
          ...
          <button class="form-button" type="reset">Отмена</button>
          <button class="form-primary-button" type="submit">Да, удалить</button>
        </form>
      </section>
    </div>
  </div>
</body>
```

Рассмотрим стили элементов внутри компонента. Начнём с подложки. Для неё используем `position: fixed;` и `top: 0; right: 0; bottom: 0; left: 0;`, чтобы растянуть на весь экран. `z-index` устанавливают так, чтобы модальное окно и подложка перекрыли все слои, что есть на странице.

Замечание

Как правильно устанавливать значение свойства `z-index`, чтобы не запутаться в слоях, читайте в статье Managing CSS Z-Index In Large Projects.

Чтобы подключить плавные переходы при открытии и закрытии окна, скрывать и показывать компонент будем за счёт CSS-свойств `visibility` и `opacity`.

Размещаем окно по центру экрана с помощью CSS-свойства `display: flex;`, затем выравниваем по главной и поперечной оси.

Скролл по вертикали для подложки отключаем, устанавливаем `overflow-y: hidden;`.

Для состояния *Модальное окно открыто* добавим специальный класс `modal-open`.

```
.modal {
  position: fixed;
  top: 0;
  right: 0;
  bottom: 0;
  left: 0;
  z-index: 10; /* Мы не знаем сколько всего слоёв на странице, поставим значение побольше */

  display: flex;
  align-items: center;
  justify-content: center;

  visibility: hidden;
  opacity: 0;

  background-color: rgba(12, 43, 61, 0.3);

  transition-duration: 0.2s;
  transition-property: opacity;

  overflow-y: hidden;
  pointer-events: none;
}

.modal-open {
  pointer-events: auto;
  visibility: visible;
  opacity: 1;
}
```

Разберёмся со стилями модального окна. Как правило, кнопка «Закрыть» расположена рядом с окном, можно её вынести в отдельный слой, задав `position: absolute;`. Точкой отсчёта для положения кнопки можно считать окно и задать ему `position: relative;`. Важно ограничить окно по высоте, например, так: `max-height: calc(100vh - 60px);`. *Помним, что для компонента мы отключили вертикальный скролл.* Для окон с большим количеством контента нужно предусмотреть автоматическое появление вертикальной прокрутки `overflow-y: auto` внутри контента.

```
.modal-box {
  position: relative;
  width: 380px;
  max-height: calc(100vh - 60px);
  background-color: #f4f7f9;
  box-shadow: 0 5px 40px rgba(0, 0, 0, 0.15);
  border-radius: 8px;
  overflow-y: auto;
  overscroll-behavior: contain;
}
```

Чтобы отключить прокрутку в родительском блоке, мы установили свойство `overscroll-behavior: contain;`.

Подробнее о том, как работает CSS-свойство `overscroll-behavior`, читайте в спецификации.

Теперь о нюансах.

Если прокручивать страницу вниз при открытом окне, то содержимое под полупрозрачной подложкой тоже перемещается, даже если оно скрыто. Разные библиотеки по-разному решают эту проблему. Например, в библиотеке компонентов bootstap 5 при открытии окна у элемента `<body>` убирают горизонтальный скролл и добавляют поле справа размером ширины полосы скроллирования (`17px`). Прокрутки страницы нет, но есть едва уловимое мерцание (*flicker*). Этот скачок возникает во время того, как полоса прокрутки исчезает и на его месте добавляется поле.

```
/* Для body добавляется встроенный стиль во время того, как модальное окно выведено */
element.style {
  overflow: hidden;
  padding-right: 17px; /* Место под скролл подложки модального окна */
}
```

В библиотеке daisyUI по-другому решили проблему со скроллированием. Для основного содержимого страницы добавили две обёртки `drawer` и `drawer-content`, первая занимает по высоте не больше высоты окна браузера, а вторая при необходимости добавляет вертикальный скролл. Получается, что во время прокручивания страницы, мы управляем скроллом не внутри тега `<body>`, а в внутри обёртки с классом `drawer-content`. В разметке и стилях это выглядит, примерно, так:

```
<body>
  <div class="drawer"><div class="drawer-content">
      <header>
      ...
      </header>
      <main>
      ...
      </main>
      <footer>
      ...
      </footer>
    </div></div>
  <div class="modal">
    <div class="modal-box">
      <button class="modal-close-button">Закрыть</button>
      <section class="modal-content">
        <h3 class="modal-caption">Удаление</h3>
        <form class="form" action="#" method="post">
          ...
          <button class="form-button" type="reset">Отмена</button>
          <button class="form-primary-button" type="submit">Да, удалить</button>
        </form>
      </section>
    </div>
  </div>
</body>
```

```
.drawer {
  display: grid;
  width: 100%;
  overflow: hidden;
  height: 100vh;
}

.drawer-content {
  height: auto;
  overflow-y: auto;
}
```

Замечание

В стандарте HTML есть элемент dialog и он сейчас с хорошей браузерной поддержкой, но его редко используют для создания модальных окон, хотя именно для этого он и создавался. Причина в том как стандартный компонент работает с клавиатурой. Подробнее читайте в статье. Есть множество библиотек, которые реализуют модальные окна и делают эти компоненты доступными, например, A11yDialog.

Вёрстка компонента Tinder card
Кейс 1, лёгкий уровень