

Вёрстка многослойных элементов интерфейса

Выполнен на 51%

ВведениеУглублённая теория

+

Методика вёрстки многослойных элементов

−

Вёрстка кнопки «Наверх»Вёрстка «липкой» горизонтальной навигацииВёрстка «липкого» sidebarВёрстка окна с соглашением об использовании CookieВёрстка пользовательской подсказкиВёрстка компонента Карусель/СлайдерВёрстка компонента Выпадающий списокВёрстка компонента Чат-ботВёрстка сложного менюВёрстка компонента для расширенного поискаВёрстка компонента Tinder cardВёрстка модального окнаКейс 1, лёгкий уровень

+

Кейс 2, лёгкий уровень

+

Кейс 3, лёгкий уровеньКейс 4, лёгкий уровеньКейс 5, средний уровень

+

Кейс 6, средний уровеньКейс 7, средний уровеньКейс 8, сложный уровень

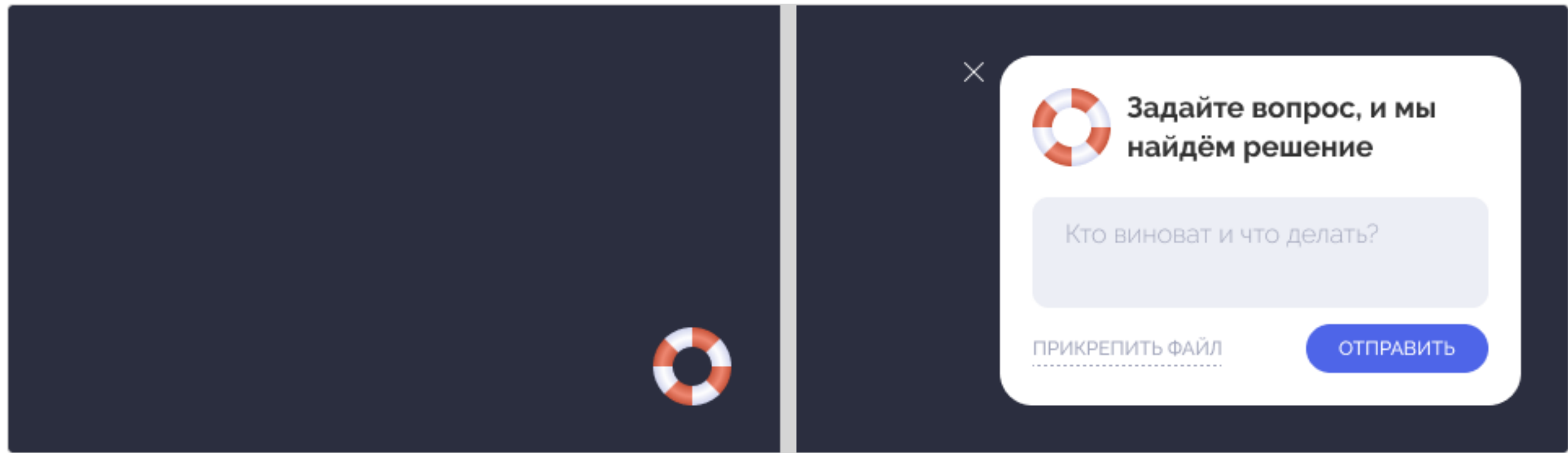
+

Кейс 9, сложный уровеньКейс 10, сложный уровень

Главная / Моё обучение / Вёрстка многослойных элементов интерфейса / Методика вёрстки многослойных элементов /

Вёрстка компонента Чат-бот

Окно чата можно увидеть практически на каждом сайте, неважно продающий он или нет. Часто это один из стандартных сервисов для ответа на вопросы пользователей. Вот лишь некоторые примеры сайтов, где есть такие чаты: [beeline.ru](#), [dom.ru](#), [sportmaster.ru](#) и другие. Окно с чатом обычно находится в правом нижнем углу сайта и перекрывает основной контент страницы. Чаше всего сначала окно свёрнуто и отображается в виде иконки либо кнопки.



Чат-бот на сайте

Поведение и расположение компонента говорит о том, что, во-первых, он спозиционирован, во-вторых, для него используется фиксированное позиционирование, его положение задаётся относительно окна сайта. В свёрнутом состоянии это обычно кнопка или ссылка, а в развёрнутом — отдельная страница, которая загружена через

`<iframe>`.

Разметка компонента может выглядеть так:

```
<!-- Кнопка для открытия чата -->
<button class="chat-button" type="button">Открыть окно чата</button>
<!-- Страница с чатом -->
<iframe class="chat-frame" id="chat" scrolling="no" frameborder="0" src="chat.html"></iframe>
```

Разметка самого чата может быть такой:

```
<section class="dialog">
  <header class="dialog-header">
    <h2>Задайте вопрос, и мы найдём решение</h2>
  </header>
  <div class="dialog-content">
    <form class="dialog-form form-message" action="#" method="post">
      <label class="form-message-label" for="message">Введите сообщение</label>
      <input class="form-message-field field" name="message" id="message" type="text" placeholder="Кто виноват и что делать?">
      <label class="form-message-label-file" for="attach">Прикрепить файл</label>
      <input class="form-message-file" name="attach" id="attach" type="file">
      <button class="form-message-button" type="submit">отправить</button>
    </form>
    <button class="dialog-close" type="button">Закрыть окно</button>
  </section>
```

Стили компонента начнём разбирать со стилей кнопки.

```
.chat-button {
  position: fixed;
  right: 30px;
  bottom: 30px;
  z-index: 10;
  width: 48px;
  height: 48px;
  padding: 0;
  border: none;
  background-image: url("../img/lifebuoy.svg");
  background-position: center;
  background-repeat: no-repeat;
  background-color: transparent;
  cursor: pointer;
}
```

Скрывать и показывать кнопку можно за счёт CSS-свойства `display`. `display: none` — скрываем, `display: block`; — показываем. Без *JavaScript* тут не обойтись. В коде будет меняться видимость кнопки и `iframe`. Один элемент скрывается, другой показывается.

`iframe` также будет с `position: fixed`, но его лучше ограничить по высоте и ширине.

Например, так:

```
.chat-frame {
  position: fixed;
  right: 0;
  bottom: 0;
  z-index: 10;

  min-width: 380px;
  min-height: 275px;

  width: 380px;
  height: 275px;

  max-height: calc(100vh - 60px);
  max-width: 70%;

  border: none;
  box-sizing: border-box;
}
```

А вот содержание окна может быть стилизовано как угодно. Например, в нашем случае так:

```
.dialog {
  position: relative;
  flex-flow: column nowrap;
  flex-direction: column;
  flex-wrap: nowrap;
  display: flex;
  padding: 20px;
  background-color: #ffffff;
  border-radius: 20px;
  box-sizing: border-box;
}

.dialog-close {
  position: absolute;
  left: -27px;
  top: 0;
  width: 20px;
  height: 20px;
  padding: 0;
  border: none;
  background-color: transparent;
  background-image: url("../img/cross.svg");
  background-position: center;
  background-repeat: no-repeat;
  cursor: pointer;
}

.dialog-header {
  min-height: 48px;
  padding-left: 55px;
  background-image: url("../img/lifebuoy.svg");
  background-position: left center;
  background-repeat: no-repeat;
}

.dialog-content {
  flex-grow: 1;
  display: flex;
  flex-direction: column;
  align-items: flex-start;
  max-height: 300px;
  overflow-y: auto;
}

.dialog-form {
  display: grid;
  grid-template-areas:
    "message message"
    "file button";
  gap: 10px;
}

.form-message-field {
  grid-area: message;
}

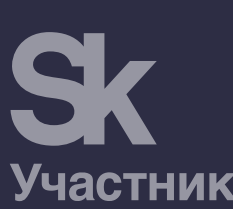
.form-message-button {
  grid-area: button;
  justify-self: end;
  ...
}

.form-message-file {
  display: none;
}

.form-message-label-file {
  grid-area: file;
  ...
}
```

[← Вёрстка компонента Выпадающий список](#)

[Вёрстка сложного меню →](#)



Практикум

Тренажёры

Подписка

Для команд и компаний

Учебник по PHP

Профессии

Фронтенд-разработчик

JavaScript-разработчик

Фулстек-разработчик

Услуги

Работа наставником

Для учителей

Стать автором

Курсы

HTML и CSS. Профессиональная вёрстка сайтов

HTML и CSS. Адаптивная вёрстка и автоматизация

JavaScript. Профессиональная разработка веб-интерфeyсов

JavaScript. Архитектура клиентских приложений

React. Разработка сложных клиентских приложений

Node.js. Профессиональная разработка REST API

Node.js и Nest.js. Микросервисная архитектура

TypeScript. Теория типов

Алгоритмы и структуры данных

Паттерны проектирования

Webpack

Vue.js 3. Разработка клиентских приложений

Git и GitHub

Анимация для фронтендеров

Блог

С чего начать

Шпаргалки для разработчиков

Отчеты о курсах

Информация

Об Академии

О центре карьеры

Остальное

Написать нам

Мероприятия

Форум