Вёрстка многослойных

элементов интерфейса

Углублённая теория

Методика вёрстки

многослойных элементов

Вёрстка кнопки «Наверх»

🖪 Вёрстка «липкого» sidebar

использовании Cookie

🖪 Вёрстка компонента Чат-бот

Bёрстка компонента Tinder card

Вёрстка модального окна

🖸 Кейс 1, лёгкий уровень

№ Кейс 2, лёгкий уровень

№ Кейс 3, лёгкий уровень

№ Кейс 5, средний уровень

🖸 Кейс 6, средний уровень

№ Кейс 7, средний уровень

🖸 Кейс 8, сложный уровень

🕠 Кейс 9, сложный уровень

Кейс 10, сложный уровень

Вёрстка сложного меню

Вёрстка «липкой» горизонтальной

Вёрстка окна с соглашением об

Вёрстка пользовательской подсказки

🖪 Вёрстка компонента Карусель/Слайдер

🖪 Вёрстка компонента Выпадающий список

Вёрстка компонента для расширенного

Выполнен на 51%

Введение

## Вёрстка «липкой» горизонтальной навигации

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

Во-первых, сразу уточним, что такое «липкая» навигация. Сегодня этим термином называют навигацию, которая оставаться на своём месте.

всегда на виду у посетителя сайта. То есть, сколько бы пользователь ни прокручивал страницу вниз, меню будет

В качестве «липкой» навигации может выступать любая навигация на сайте, но чаще всего это горизонтальное меню.

Подтверждённый факт: качество взаимодействия пользователей с продуктом улучшается, когда все важные для

них опции находятся в их поле зрения.

Существует несколько вариантов, как ведёт себя такая навигация:

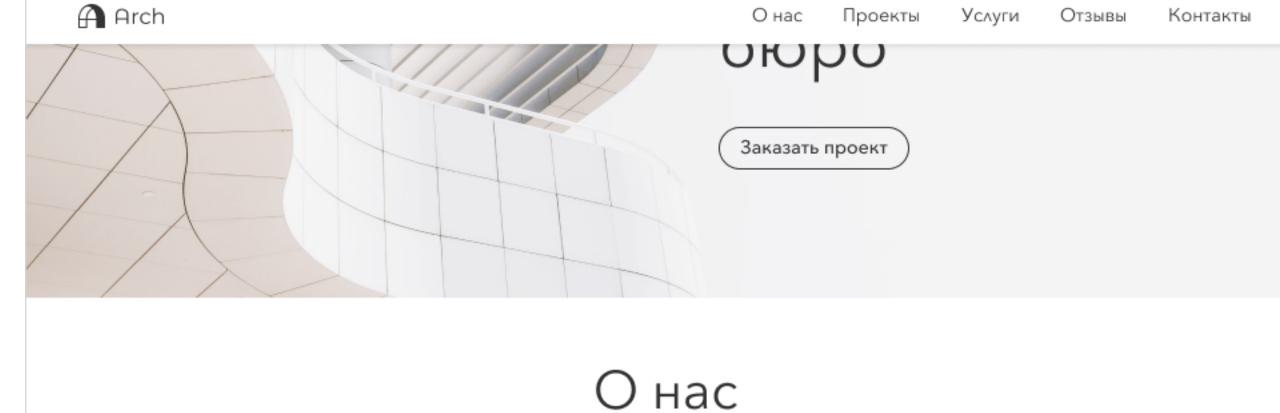
— «липкой» становится вся шапка (основная навигация по сайту, логотип, пользовательское меню) и при прокрутке остаётся на своём месте;

— лишь часть элементов навигации «залипает» и при прокрутке она превращается в компактный вариант.

Разберём реализацию обоих вариантов «липкой» навигации на сайте.

«Липкой» становится вся шапка

Arch Контакты Проекты Услуги Отзывы Архитектурное бюро Заказать проект Шапка сайта в обычном состоянии



## Создаём архитектурные проекты с 1998 года. Дорожим своей репутацией, поэтому к каждому проекту подходим ответственно.

Шапка сайта при прокрутке страницы вниз

В этом случае, как и с кнопкой «Наверх», можно использовать фиксированное позиционирование position:

fixed . Поскольку шапка идёт первой в потоке, мы можем получить нужное размещение и не указывая его позицию — CSS-свойства top и left. Но лучше их задать. Так мы сделаем нашу вёрстку устойчивой к изменению разметки. Чтобы растянуть шапку на всю ширину сайта, можем либо задать CSS-свойство width: 100%, либо установить CSS-свойство right в значение 0. Для элемента, который идёт следующим в разметке (в нашем случае основное содержимое сайта), добавим

внешний отступ сверху равный высоте шапки. Разметка:

```
<body class="page-content">
  <header class="header">
    <div class="header-wrapper container">
      <nav class="nav">
        <a class="nav-logo" href="index.html">
         <img src="img/logo.png" alt="Логотип" width="122" height="56">
        </a>
        <a class="nav-link" href="#">О наc</a>
         class="nav-item">
           <a class="nav-link" href="#">Проекты</a>
         </nav>
    </div>
  </header>
  <main class="page-main">
  </main>
 </body>
Стили:
```

```
.header {
   position: fixed;
   top: 0;
   left: 0;
   width: 100%;
   background-color: #ffffff;
   box-shadow: 0 4px 10px rgba(135, 135, 135, 0.1);
   z-index: 100;
 .page-main {
   margin-top: 106px;
При существующей разметке, чтобы заставить «залипать» шапку вверху страницы, подходит также и position:
```

.header { position: sticky; top: 0;

контейнер <body class="page-content">, в котором будет плавать «липкий» элемент <header class="header">.

sticky. Подобный пример рассматривался в разделе «Как работает position: sticky». У нас есть «липкий»

```
left: 0;
   width: 100%;
   background-color: #ffffff;
   box-shadow: 0 4px 10px rgba(135, 135, 135, 0.1);
   z-index: 100;
Преимущество использования этого варианта — шапка уже не перекроет часть содержимого страницы и внешний
отступ для блока <main class="page-main">, как в предыдущем примере, не нужен.
```

Компактная «липкая» навигация

Мужские

Женские

MHLWATCHSON Санкт-Петербург 🔻 **Б** Корзина

Новые поступления

Акции и скидки

На изображениях ниже показано как выглядит навигация в обычном состоянии и при прокрутке страницы вниз.



```
Новая коллекция часов Festina
                              уже на сайте
Компактная «липкая» навигация при прокрутке страницы вниз
«Залипают» лишь геолокация, логотип и пользовательское меню.
Без отдельного класса, который будет добавлен для элемента в нужный момент с помощью JavaScript,
```

не обойтись. По умолчанию все элементы находятся в потоке. Затем «липкая» навигация позиционируется с помощью position: fixed. При этом для скрываемого элемента (список ссылок) установим отрицательный внешний

<div class="nav-topbar">

Санкт-Петербург

</a>

<a href="#" class="nav-geo">

отступ сверху. С помощью управления внешним отступом, мы плавно скроем элемент. Для этого добавим элементу CSS-свойство transition: margin-top 0.5s linear; Элемент, который идёт следом (основное содержимое), получит внешний отступ сверху, равный высоте «залипшей» навигации. За счёт большого значения z-index мы может гарантировать, что шапку не перекроют другие спозиционированные элементы.

<img src="/img/nav.svg" alt="Иконка геолокации" width="20" height="20">

Как это выглядит в вёрстке: Разметка: <header class="header header-sticky"> <div class="header-wrapper container"> <nav class="nav">

```
<a class="nav-logo" href="#">
         <img src="/img/logo.png" alt="Логотип" width="185" height="22">
       </a>
       <a href="#" class="user-menu-link user-menu-link-cart">
            <img src="/img/cart.svg" alt="Корзина" width="22" height="22">
          </a>
         </div>
     <a href="#" class="nav-link">Мужские</a>
       class="nav-item">
         <a href="#" class="nav-link">Женские</a>
       </nav>
  </div>
 </header>
 <main class="main-page">
 </main>
Стили:
 .header {
  min-height: 102px;
  margin-bottom: 33px;
  border-bottom: 1px solid rgba(27, 24, 25, 0.1);
```

```
.nav {
 display: flex;
 flex-wrap: wrap;
.nav-topbar {
 display: flex;
 flex-wrap: wrap;
 align-items: center;
 width: 100%;
 padding-top: 27px;
 padding-bottom: 12px;
 background-color: #ffffff;
.nav-list {
 position: relative;
 display: flex;
 flex-wrap: wrap;
 justify-content: center;
 padding: 0;
 margin: 0 auto;
 width: 1110px;
 list-style: none;
 transition: margin-top 0.5s linear;
.header-sticky {
 position: fixed;
 top: 0;
 left: 0;
 width: 100%;
 background-color: #ffffff;
 z-index: 10;
.header-sticky + .main-page {
 margin-top: 135px;
.header-sticky .nav-list {
 margin-top: -120px;
```

- «Липкая» навигация может доставлять неудобство Распространённая ошибка в UX — слишком большая «липкая» шапка. В итоге для основного контента остаётся чрезвычайно мало места, и посетители сайта получают удушающий, «клаустрофобный» опыт.

Курсы

**<** Вёрстка кнопки «Наверх» Вёрстка «липкого» sidebar >

Фиксированная у вас шапка или нет, просмотр страниц должен быть комфортным для посетителей.

**Участник** 

HTML и CSS. Адаптивная вёрстка и автоматизация Для команд и компании Учебник по РНР Профессии Фронтенд-разработчик

Практикум

JavaScript-разработчик

Фулстек-разработчик

Работа наставником

Тренажёры

Подписка

Услуги

Для учителей

Стать автором

JavaScript. Профессиональная разработка веб-интерфейсов JavaScript. Архитектура клиентских приложений React. Разработка сложных клиентских приложений TypeScript. Теория типов

Vue.js 3. Разработка клиентских приложений

HTML и CSS. Профессиональная вёрстка сайтов

Отчеты о курсах Информация Об Академии О центре карьеры

Шпаргалки для разработчиков

Блог

С чего начать

Node.js. Профессиональная разработка REST API Node.js и Nest.js. Микросервисная архитектура Остальное Написать нам Алгоритмы и структуры данных Мероприятия Паттерны проектирования Форум

**⟨/⟩** → ♥ → **!!!** 

© ООО «Интерактивные обучающие технологии», 2013–2023

Анимация для фронтендеров

Webpack

Git и GitHub