

Вёрстка навигационных элементов интерфейса

Выполнен на 0%

Углублённая теория

Методики вёрстки элементов навигации

Содержимое основной навигации

Хлебные крошки

Якоря: лендинг, оглавление материала

Расширенный футер

Кейс 1, лёгкий уровень

Кейс 2, лёгкий уровень

Кейс 3, лёгкий уровень

Кейс 4, средний уровень

Кейс 5, средний уровень

Кейс 6, средний уровень

Кейс 7, сложный уровень

Кейс 8, сложный уровень

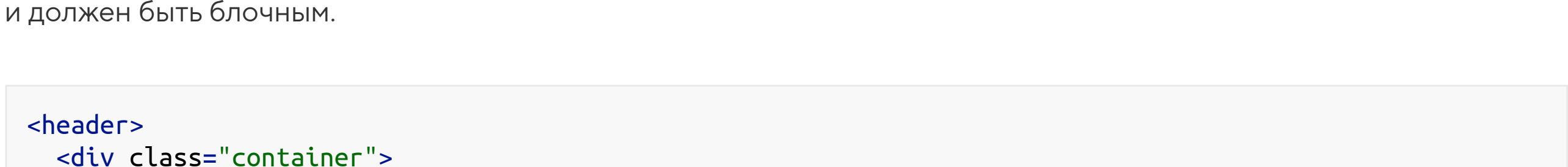
Главная / Моя обучение / Вёрстка навигационных элементов интерфейса / Методики вёрстки элементов навигации /

Якоря: лендинг, оглавление материала

Лендинг

Лендинг — это такой сайт, основная информация которого размещена на одной странице. Обычно это продающий сайт. Зачастую у лендинга технически больше одной страницы (например, есть страница с подробными условиями или даже целая лента новостей), но основной контент — это достаточно стандартный набор блоков, который выстроен в столбик на одной-единственной странице. И раз этой информации достаточно много, её нужно структурировать. У лендинга появляется меню, и оно должно красиво прокрутить нас в то место, где на странице лежит эта информация.

Понятно, что красивую прокрутку обеспечивает скрипт, который мы разберём в навывке по *JavaScript*. Но саму навигацию мы сейчас разберём.



Макет основной навигации лендинга

Сама вёрстка будет абсолютно такая же, как если бы это было меню нормального обычного сайта: список ссылок, логотип, возможно, кнопки действий или пользовательское меню. Разница будет в ссылках: ссылка должна быть якорем. Мы достаточно подробно говорили о том, как устроен якорь, в [материале про ссылки](#).

Сразу разметим полезные классы и обёртки, которые нам понадобятся. Из обёрток появляется стандартный контейнер для ограничения макета и обёртка для блока с номером телефона: он достаточно комплексный и должен быть блочным.

```
<header>
<div class="container">
  <nav class="main-nav">
    <a class="logo" href="/index.html">
      
      <span class="visually-hidden">Наверх</span>
    </a>
    <p class="brand-description">Натуральный <strong>клееновый сыр</strong></p>
    <ul class="site-nav">
      <!-- У этих ссылок мы прописываем адрес заранее, он будет вести к ключу, зашитому в код соответствующих блоков -->
      <li><a class="nav-link" href="#composition">Состав</a></li>
      <li><a class="nav-link" href="#feedback">Отзывы</a></li>
      <li><a class="nav-link" href="#price">Цена</a></li>
    </ul>
    <div class="call-banner">
      <p>
        <span>8 999 888 77 66</span><strong>
          звонок бесплатный
        </strong>
      </p>
    </div>
  </div>
  <!-- Есть вероятность, что откроется новое окно, а не прокрутится страница до формы заказа, поэтому не пишем конкретный адрес -->
  <a class="button" href="#">Закazać</a>
</nav>
</header>
```

Стилизация кажется тоже достаточно простой:

```
/* глобальные стили, которые нужны, чтобы всё заработало. Скорее всего будут применяться в нормальной ситуации не к body, а к общему контейнеру сайта, но у нас нет всего сайта */
body {
  min-width: 960px;
  margin: 0;

  font-family: "Montserrat", "Arial", sans-serif;
  font-size: 18px;
  line-height: 28px;
  font-style: normal;
  font-weight: normal;
  color: #333333;
}

background-color: #ffffff;

a {
  text-decoration: none;
}

img {
  max-width: 100%;
  height: auto;
}

/* То, что скрыто. Мы скрываем в основном то, что нужно только для доступности, вроде описания ссылки с логотипа наверх */
.visually-hidden {
  position: absolute;
  clip: rect(1px, 1px, 1px, 1px);
  padding: 0;
  border: 0;
  height: 1px;
  width: 1px;
  overflow: hidden;
}

/* Контейнер, ограничивающий фиксированный макет */
.container {
  max-width: 920px;
  margin: 0 auto;
  padding-right: 20px;
  padding-left: 20px;
  background-color: #f8eee4;
}

/* Выбрали метод построения сетки */
.main-nav {
  display: flex;
  align-items: center;
  padding-top: 18px;
  padding-bottom: 18px;
}

.main-nav p {
  font-size: 11px;
  font-weight: normal;
}

/* Описали ссылки только для этого блока */
.main-nav a {
  text-decoration: none;
  color: inherit;
}

.main-nav ul {
  list-style: none;
  display: flex;
}

.brand-description {
  line-height: 1em;
  padding-right: 10px;
  padding-left: 10px;
}

/* Стилизуем кнопки только для этого блока */
.main-nav .button {
  display: block;
  max-width: 120px; /* чтобы сетка не сломалась, ограничим размер кнопки */
  padding-top: 7px;
  padding-bottom: 7px;
  padding-right: 16px;
  padding-left: 16px;
  text-align: center;
  color: #f877fd;
  background-color: #f85c2b;
  transition: opacity 0.3s, color 0.3s, background-color 0.3s; /* можно было написать all 0.3s, но это неэффективное свойство и им непросто управлять */
}

.button:hover,
.button:focus {
  color: #f877fd;
  opacity: 0.6;
  background-color: rgba(248, 91, 43, 0.4);
}

.site-nav {
  margin-left: 90px;
}

.site-nav a {
  padding-top: 5px;
  padding-bottom: 5px;
  padding-right: 10px;
  padding-left: 10px;
}

/* Блок с телефоном */
.call-banner {
  display: grid;
  grid-template-columns: auto auto;
  grid-column-gap: 10px;
  align-items: center;
  position: relative;
  padding-right: 20px;
  padding-left: 60px;
}

.call-banner::before {
  content: "";
  position: absolute;
  top: 14px;
  left: 20px;
  width: 30px;
  height: 30px;
  background-image: url("assets/call_3n.svg");
  background-size: 30px 30px;
  background-repeat: no-repeat;
  background-position: center;
}

.call-banner span {
  font-weight: bold;
  font-size: 14px;
  line-height: 16px;
  color: #f85c2b;
}
```

Самое интересное во всей этой истории — это, разумеется, ответная часть якоря, которая должна быть защита в то место, куда нужно присколотить пользователя, которую нужно не забыть, не потерять, не перепутать ни буквы, а то ничего не сработает. Очень важно проверить работу каждого якоря: первые 4 могут быть правильными, а пятый — таким же, как четвертый.

Выглядеть обратная часть якоря в нашем случае будет так:

```
— {id="composition"};

— {id="feedback"};

— {id="price"}.
```

Эти элементы нужно вставить туда, куда будет скролиться страница при нажатии на якорь в начале страницы.

Важно: в ключах якорей не нужно писать ничего того, что вы не хотели бы показать своему заказчику, потому что якорь отобразится в *URL* и будет виден всем, кто знает, куда смотреть.

Оглавление материала

Навигация по материалу нужна в нескольких случаях: когда вы в Википедии, когда вы верстаете ресурс для чтения книг или других объёмных материалов, когда у вас большие статьи (например, научного характера) или если на сайте планируется масштабная работа с документами и протоколами.

7 мифов о программистах

Программирование — большой мир, в который страшно войти со стороны. Рассказываем, чего точно не стоит бояться, и помогаем развеять популярные мифы.

Содержание статьи

1. Программирование не для меня
2. Страшно сделать ошибку
3. Ошибка — конец света
4. Сложно сделать первый проект
5. Невозможно научиться самостоятельно
6. Код можно никому не показывать
7. Сразу после курсов платят по 200 тысяч в месяц

1. Программирование не для меня

У некоторых языков действительно очень высокий порог входа — не всегда даже понятно, с чего начать.

Макет навигации по отдельному материалу

Вполне ожидаемо, что этот блок будет обёрнут в отдельный `<div>`. Вот какая разметка у нас получилась:

```
<article class="article">
  <h1>7 мифов о программистах</h1>
  <div class="article-container">
    <p class="intro">Программирование — большой мир, в который страшно войти со стороны. Рассказываем, чего точно не стоит бояться, и помогаем развеять популярные мифы.</p>
    <div class="article-menu">
      <h3>Содержание статьи</h3>
      <ol>
        <li><a href="#part-1">Программирование не для меня</a></li>
        <li><a href="#part-2">Страшно сделать ошибку</a></li>
        <li><a href="#part-3">Ошибка — конец света</a></li>
        <li><a href="#part-4">Сложно сделать первый проект</a></li>
        <li><a href="#part-5">Невозможно научиться самостоятельно</a></li>
        <li><a href="#part-6">Код можно никому не показывать</a></li>
        <li><a href="#part-7">Сразу после курсов платят по 200 тысяч в месяц</a></li>
      </ol>
    </div>
    <h2 id="part-1">1. Программирование не для меня</h2>
    <p>У некоторых языков действительно очень высокий порог входа — не всегда даже понятно, с чего начать.</p>
  </article>
```

URL ключа (`part-1` и последующие) мы специально сделали не значащими, а нарочито функциональными: мы не знаем наверняка, какой контент будет опубликован при запуске сайта, и мы считаем, что всё, что нам отдают в макете — это рыба (текст для видимости, чаще всего ничего не значащий и нужный только чтобы показать, как выглядит дизайн с каким-то реалистичным наполнением), и ориентироваться на неё не следует.

Что же тут будет особенного со стилизацией: нестандартные номера, придётся переподключать счётчик (это весьма подробно разбирали в материале про списки, а подчёркивание ссылок доходит до края, захватывая в том числе переподключённую нумерацию).

```
.article-menu ol {
  margin-bottom: 30px;
  padding: 0;
  list-style: none;
  counter-reset: num; /* отключил текущую нумерацию */
}

.article-menu li {
  position: relative;
  counter-increment: num; /* подключили счётчик в нужном месте и теперь можем настраивать */
  margin-bottom: 15px;

  font-family: "Merr1weather", "Book Antiqua", serif;
  font-style: normal;
  font-weight: normal;
  font-size: 16px;
  line-height: 28px;
  color: #5a6eb6;
}

.article-menu li::before {
  content: counter(num) "."; /* включили номера и точки поставили после номеров */
  position: absolute;
  left: 0;
  top: 0;
}

.article a {
  color: inherit;
  display: inline-block;
  padding-left: 23px; /* сделали увеличенную зону клика, чтобы было удобнее в неё попадать и чтобы её подчеркнуть */
  border-bottom: 1px dashed #5a6eb6;
  text-decoration: none;
}
```

Вопрос того, как именно это будет работать, лежит в области бэкенда и будущих навыков по *JavaScript*.

Ознакомились со статьёй?

Сохранить прогресс

< Хлебные крошки

Расширенный футер >