Главная / Моё обучение / Вёрстка навигационных элементов интерфейса / Методики вёрстки элементов навигации / Содержимое основной навигации

(I) CloudHosting

Вёрстка должна выглядеть так:

<nav>

Разберёмся с содержимым основной навигации сайта. В тег (<nav> попадает то, что составит единственную сквозную навигацию по всему сайту, и у разных сайтов это может быть меню, или меню и логотип, если логотип работает как ссылка на главную страницу, или логотип, меню и пользовательские ссылки (корзина и вход в систему, например).

ਲ Виталий Ризун 🔻

Зарегистрироваться

Войти

Разумеется, первое, что верстается во всех этих случаях — это семантическая разметка. О том, как вообще размечать списки, мы говорили в теоретической части, так что не будем останавливаться на нюансах и сверстаем наши макеты все сразу.

Контакты

```
Макет основной навигации сайта, горизонтальное отображение
    {\rm FF}_{\rm bags}
                                Сумки > Шопперы > Текстильные
     Рюкзаки
     Сумки
     Чемоданы
```

```
Аксессуары
     Подарочные карты
     Новая коллекция
     Бестселлеры
                                          BUT FIRST, COFFEE.
     Скидки и акции
                                             Tatte
Макет основной навигации сайта в столбик
```

```
<a href="/index.html">
     <img src="assets/logo.png" width="312" height="44" alt="Логотип">
     <span>Ha главную</span>
   </a>
   <l
     <a href="#">0 нас</a>
     <a href="#">Цены</a>
     <a href="#">Блог</a>
     <a href="#">Контакты</a>
   <a href="#">Войти</a>
     <a href="#">Зарегистрироваться</a>
   </nav>
Мы сделали ссылками не только сами пункты, но и логотип, даже учитывая, что мы находимся на главной
странице. Это спорное решение, и раньше мы обычно делали логотип на главной странице ссылкой без атрибута
href, ведьк логотип на остальных страницах должен возвращать на главную, а когда пользователь уже на ней, это
абсурдно. Сейчас мы посмотрели статистику использования некоторых сайтов и выяснили, что некоторые
пользователи копируют адрес сайта с логотипа, а некоторые — обновляют главную кликом на логотип. Теперь
```

мы делаем логотип полноценной ссылкой, тем более, что это проще реализовать. В этой вёрстке сама ссылка условная, обычно она зависит от самой структуры сайта. При этом мы стараемся дописывать элемент доступности — скрытое объяснение, что это за ссылка. Здесь это объёснение в теге . У нас получился правильный, чистый и прозрачный код, сейчас допишем в него технические обёртки и основные классы. В макете есть текущий пункт меню, мы его опознали по комментариям нашего дизайнера. Он получит человекочитаемый и на всякий случай уникальный класс, то есть такой, какой наверняка не будет использован в других местах на сайте.

<div class="container"> <nav class="main-nav"> На главную 0 нас

```
<a class="nav-link nav-link-current" href="#">Цены</a>
      <a class="nav-link" href="#" >Блог</a>
      <a class="nav-link" href="#">Контакты</a>
    <a class="button button-outline" href="#">Войти</a>
      <a class="button button-filled" href="#">Зарегистрироваться</a>
    </nav>
 </div>
А вот вёрстка вертикального меню:
 <nav class="main-nav">
   <a class="logo" href="/index.html">
    <img src="assets/methodology2_logo.png" width="85" height="30" alt="Логотип сайта">
    <span class="visually-hidden">На главную</span>
   </a>
```

Сумки Чемоданы

но нам нужно, чтобы он работал).

сайтах такого обычно не происходит */

img {

display: flex;

display: flex;

.site-nav {

.logo {

.site-nav {

margin-right: 10px;

margin-bottom: 0;

list-style: none;

.nav-link-current {

color: #4f4f4f;

opacity: 1;

.nav-link:hover,

.nav-link:focus {

opacity: 0.6;

display: block;

max-width: 315px;

.button-outline:hover,

color: #f8f7fd;

opacity: 0.6;

.nav-link {

content: "";

 ${
m FF}_{
m bags}$

Рюкзаки

Сумки

Чемоданы

Аксессуары

Подарочные карты

padding-left: 8px;

list-style: none;

которые вы встречали.

.button-outline:focus {

position: relative;

position: absolute;

.nav-link-current::after {

background-color: #03919a;

Закончим оформление: допишем подчёркивание для актуального элемента.

.button {

font-weight: 600;

margin-top: 0;

padding: 0;

.site-nav li {

align-items: center;

Рюкзаки

```
<a href="#">Аксессуары</a>
     <a href="#">Подарочные карты</a>
   <a href="#">Новая коллекция</a>
     <a href="#">Бестселлеры</a>
     <a href="#">Скидки и акции</a>
   <
      <a href="#">
        <img src="assets/insta_2m.svg" width="30" height="30" alt="Instagram">
        <span class="visually-hidden">Instagram</span>
 <!-- Социальные сети: иконка заменяет слово и читается однозначно, поэтому вставили как контентные
 изображения -->
      </a>
     <
      <a href="#">
        <img src="assets/fb_2m.svg" width="30" height="30" alt="Facebook">
        <span class="visually-hidden">Facebook</span>
      </a>
     <
      <a href="#">
        <img src="assets/vk_2m.svg" width="30" height="30" alt="Vkontakte">
        <span class="visually-hidden">Vkontakte</span>
      </a>
    </nav>
Обратите внимание, что мы сделали иконки социальных сетей контентными изображениями: такие элементы
мы верстаем как контентную картинку со скрытым объяснением не потому, что каждая иконка читается как
общепонятный знак, но и потому, что количество иконок может увеличиться, а работать с сайтом, вероятно, будет
контент-менеджер без доступа к стилям, так что нужно учесть возможность появления новых соцсетей (то же
касается иконок банков и тому подобных) и не писать им значащих классов, что неизбежно при переносе иконок
соцсетей в оформление, а также упростить по возможности добавление в список новых пунктов.
Теперь нужно построить из него сетки. Для этого, раз у нас фиксированный макет, зададим параметры этого
```

(и системе выстраивания классов в целом) мы выпустим отдельный навык. Вот свойства класса visually-hidden: .visually-hidden { position: absolute; clip: rect(1px, 1px, 1px, 1px);

самого фиксированного макета, важные параметры для картинок и ссылок, а также свойства класса visually-

hidden . Так как здесь макеты ограничены одним элементом (иногда — его непосредственным окружением),

мы используем и глобальные классы, и сбросы. О «Нормальном сбросе» есть статья в нашем блоге, а о БЭМ

padding: 0; border: 0; height: 1px; width: 1px; overflow: hidden;

Вот параметры, которые мы указываем как глобальные (и это всё-таки связано с тем, что это учебный макет,

```
body {
 min-width: 960px; /* мы делаем фиксированный макет */
  margin: 0; /* сбросы не делали, так что убираем отступы тут */
/* Параметры шрифта указываем здесь, потому что они относятся ко всему макету целиком */
  font-size: 16px;
  line-height: 22px;
  font-style: normal;
  font-weight: normal;
  color: #73a1a4;
 background-color: #ffffff;
  text-decoration: none; /* из-за того, что во всём макете выключено подчёркивание ссылок. На живых
```

max-width: 100%; height: auto; /* эти параметры нужны, чтобы картинки не теряли пропорции ни при каких обстоятельствах .container { max-width: 920px; /* это наше рабочее пространство */ margin: 0 auto; padding-right: 48px; padding-left: 48px; background-color: #ffffff; Теперь сетки. Мы, если смотреть стратегически, можем выбрать между флексами и гридами. Для обоих макетов, с которыми мы сейчас работаем, мы выберем флексбокс, потому что его достаточно. Выбирать флексбокс или грид — задача верстальщика, и её решение зависит от того, как ему удобнее. В макете с горизонтальным меню флексами будут и вся навигация в целом, и отдельные её части. .main-nav {

```
flex-shrink: 0; /* логотип нельзя сжимать */
 .user-nav {
   display: flex;
   flex-wrap: wrap;
   flex-shrink: 0; /* пользовательскую навигацию тоже нельзя сжимать */
Сетка выстроена, теперь добавим отступы, внешние и внутренние, фоны, уберём буллиты там, где они не нужны.
 .logo {
   width: 176px;
   margin-right: 60px;
```

flex-wrap: wrap; /* очень важно для навигации оставлять возможности переполнения */

margin-right: 10px; .site-nav li:last-child { margin-left: 5px;

```
.user-nav {
   margin-left: auto;
   margin-top: 0;
   margin-bottom: 0;
   padding: 0;
   list-style: none;
 .user-nav li {
   margin-top: 10px;
   margin-bottom: 10px;
   margin-left: 25px;
Вспомним про то, что большинство ссылок должны иметь состояния, опишем стили ссылок, дополним сами
ссылки настройками подключения этих состояний. Раз всё, что у нас есть в этом макете, это ссылки — на этом
мы практически закончим стилизацию.
 .logo {
   opacity: 1;
   transition: opacity 0.3s; /* применяем к начальному состоянию, чтобы красивое изменение было как при
 подключении состояния, так и при отключении */
 .logo:hover,
 .logo:focus {
   opacity: 0.6;
 .nav-link {
   display: block;
   padding: 5px;
   margin-left: 5px;
   margin-right: 5px;
   transition: opacity 0.3s;
   color: inherit;
```

padding-top: 5px; padding-bottom: 5px; padding-right: 18px; padding-left: 18px; text-align: center; border: 2px solid #03919a; border-radius: 32px; .button-outline { color: inherit; background-color: inherit; transition: color 0.3s, opacity 0.3s; .button-filled { color: #f8f7fd; background-color: #03919a; transition: color 0.3s, background-color 0.3s; /* Теоретически можно было написать all 0.3s, но это значение негативно влияет на скорость загрузки сайта и сложно расшифровывается */ .button-filled:hover, .button-filled:focus { color: inherit; background-color: inherit;

bottom: -3px; left: -1px; width: 120%; height: 8px; background-color: #adf5fa; background-position: center; background-repeat: no-repeat; Это было горизонтальное меню, оно типичнее и традиционнее, но вертикальное меню тоже достойно внимания. В вертикальном меню нужно убрать буллиты и дальше очень внимательно работать с отступами. Дело в том, что отступы от края вьюпорта (экрана) приписываются всему списку целиком, но очень хорошо, когда ссылка имеет свои отступы, чтобы пользователю было удобнее попадать по ссылке мышью (или пальцем на тачпаде), и при этом странно, если эти самые внутренние отступы ссылки очень велики, потому что тогда кликабельность может возникнуть в каком-то далёком от ссылки месте.

Сумки > Шопперы > Текстильные

```
Новая коллекция
    Бестселлеры
                                      BUT FIRST, COFFEE.
    Скидки и акции
                                        Tatte
Макет основной навигации сайта
Семантическая вёрстка осталась наверху страницы. А так могут выглядеть стили такого меню:
 .main-nav a {
   display: block;
   padding: 5px; /* вот собственно увеличенная зона клика, со всех сторон */
 .nav-list {
   margin: 0;
 /* вот отступы всего элемента */
   padding-top: 16px;
   padding-bottom: 9px;
```

Если меню вертикальное, то зачастую появляются разделители между разными тематическими списками. Нужно сделать их так, чтобы эти разделители не мешали появлению новых пунктов. Нужно понять логику, по которой такие разделители появляются и описать эту логику селекторами. Например, здесь это полоски между двумя списками. Мы выбрали селектор, который означает «такой тег
 - , непосредственно после которого следует

ещё один тег ». Подробнее про селекторы — в двух главах тренажёров: «Знакомство» и «Погружение».

```
.main-nav ul + ul::before {
   content: "";
   position: absolute;
   left: -20px;
   right: -20px;
   top: 0;
   display: block;
   height: 1px;
   background-color: #fffcfa;
   opacity: 0.3;
Наконец, укажем параметры для состояний ссылок. Даже если в макете нет информации о работе ссылок,
их следует указать. Ориентируйтесь на собственные представления о прекрасном или на удачные образцы,
```

.main-nav-link:hover, .main-nav-link:focus { opacity: 0.6; /* border-bottom: 1px solid #ffffff; — подчёркивание зоны клика */

```
/* background-color: #384461;
border-radius: 16px; — изменится фон, и края у него будут скруглены */
 Ознакомились со статьёй?
   Сохранить прогресс
                                                                         Хлебные крошки 🗦
Методики вёрстки элементов навигации
```

Участник

Подписка Для команд и компаний Учебник по РНР Профессии React-разработчик

Практикум

Работа наставником

Конфиденциальность

Тренажёры

JavaScript. Профессиональная разработка веб-интерфейсов JavaScript. Архитектура клиентских приложений React. Разработка сложных клиентских приложений Фронтенд-разработчик РНР. Профессиональная веб-разработка PHP и Yii. Архитектура сложных веб-сервисов Фулстек-разработчик Node.js. Разработка серверов приложений и API Бэкенд-разработчик Анимация для фронтендеров Вёрстка email-рассылок Услуги

Vue.js для опытных разработчиков

Сведения об образовательной организации Лицензия № 3026

⟨/⟩ → **♥** → **!!!**

HTML и CSS. Профессиональная вёрстка сайтов

HTML и CSS. Адаптивная вёрстка и автоматизация

Курсы

Информация Об Академии О центре карьеры Остальное Написать нам Мероприятия Форум

Блог

С чего начать

Отчеты о курсах

Шпаргалки для разработчиков

© ООО «Интерактивные обучающие технологии», 2013–2022