



HTML и CSS. Адаптивная вёрстка и автоматизация

Уровень 2, с 21 ноября 2022 по 30 января 2023

Меню курса

[Главная](#) /

Критерии

Подготовка и проверка личных проектов проводится по базовым и дополнительным критериям.

Базовые критерии охватывают наиболее важные требования к проекту и проверяют основные знания и навыки. Для успешной защиты личного проекта должны быть выполнены все базовые критерии.

Дополнительные критерии проверяют то, насколько студент внимателен к деталям, и оценивают проект с точки зрения шлифовки его качества и оптимизации. Выполнение этих критериев необходимо для защиты на 100%.

Во время финальной защиты баллы за выполнение дополнительных критериев добавляются только при выполнении всех базовых.

Для подачи проекта на защиту нужно сверстать мобильное, планшетное и десктопное состояния трёх страниц проекта.

Базовые



Разметка [Свернуть все](#)

- ▲ **Б1.** Выполнена HTML-разметка всех страниц проекта и всех элементов на этих страницах.

Этот критерий говорит о том, что все страницы проекта и все скрытые и дополнительные элементы должны быть размечены. Например, всплывающие и появляющиеся элементы, модальные окна, все слайды в слайдере.

^ Б2. Грубые ошибки в разметке отсутствуют.

Грубые ошибки:

- Ссылки сделаны не тегом `<a>`, а другими тегами;
- Использование строчных элементов для создания крупных (сеточных) блоков;
- Абзацы сделаны не тегами `<p>`, а `

`.

Негрубые ошибки:

- Отсутствие семантических тегов `<header>`, `<footer>`, `<section>` и других.

Важно: тег `<a>` — элемент с «прозрачной» моделью содержимого. Поэтому если `<a>` вложен в элемент с фразовым содержимым, например в ``, то внутри самого `<a>` может быть только фразовый контент. А если `<a>` вложен в элемент с потоковым содержимым, например в `<div>`, то и содержимое у `<a>` может быть потоковое, включая другой `<div>`.

Поэтому в ссылку можно включать целые секции, если `<a>` вложен в потоковый блок:

Верно:

```
<aside>
  <h2>Ограниченные предложения</h2>
  <a href="/buy">
    <section>
      <figure>
        
        <figcaption>
          Набор для путешествий «Baxter of California»
        </figcaption>
      </figure>
      <p>Всего за 100 рублей!</p>
    </section>
  </a>
</aside>
```

Неверно:

```
<span>
  <a href="/buy">
    <section>

      <h2>Акция!</h2>
```

```
<p>Всего за 100 рублей!</p>
</section>
</a>
</span>
```

Б3. Документ проходит проверку на валидность <https://validator.w3.org/nu/>.

- ^ Б4. В разметке отсутствует дублирование кода для одного и того же элемента, с помощью которого элемент отображается в разных местах страницы на разных версиях: мобильной, десктопной, планшетной. Этот критерий не касается элементов, которые скрываются или показываются в разных версиях.

Неверно:

```
<div class="block block--for-mobile">
  <h2 class="block__title">Заголовок</h2>
  <p class="block__text">Произвольный текст</p>
</div>

<div class="block block--for-desktop">
  <h2 class="block__title">Заголовок</h2>
  <p class="block__text">Произвольный текст</p>
</div>
```

- ^ Б5. Отсутствуют типовые ошибки в разметке по методологии.

Отсутствуют следующие ошибки методологии БЭМ:

- Создание элемента без родительского блока. Это означает, что не может быть элемента `block__element`, если выше по дереву нет DOM-элемента с именем `block`.
- Создание элемента для элемента. **Неверно:**

```
<div class="block__element1">
  <div class="block__element1__element2"></div>
</div>
```

Верно:

```
<div class="block__element1">
  <div class="block__element2"></div>
</div>
```

- Создание модификатора для модификатора. **Неверно:**

```
<div class="block block--mod1--mod2"></div>
```

Верно:

```
<div class="block block--mod1 block--mod2"></div>
```

- Использование модификатора без блока или элемента, который он модифицирует (при использовании модификатора у тега должно быть как минимум два класса: класс блока/элемента и класс модификатора). **Неверно:**

```
<div class="block--mod"></div>
```

Верно:

```
<div class="block block--mod"></div>
```

Допустимые стили именования БЭМ-сущностей: **Международный:** `block__elem--mod`

^ **B32.** Названия полей форм привязаны к своим полям с помощью `<label>`.

Для улучшения взаимодействия пользователя с элементами форм должна быть реализована активация поля при нажатии на его описание. Для этого необходимо связывать элемент формы с его описанием, используя тег `</label>` и идентификаторы.

Верно: элемент формы `radio` связан с его описанием через идентификатор.

```
<input type="radio" id="spb">
<label for="spb">Санкт-Петербург</label>
```

Верно: элемент формы `radio` и подпись обернуты в `<label>`

```
<label>
  <input type="radio"> Санкт-Петербург
</label>
```

Неверно: описание никак не связано с элементом формы.

```
<input type="radio" id="spb"> Санкт-Петербург
```

Стилизация [Свернуть все](#)

^ Б6. Раскладка блоков на странице сделана на флексах и гридах

Гриды используются для основных структурных блоков страницы (шапка, главный контент, подвал) и для создания колонок в контенте страниц.

Пример реализации стики-футера:

```
<html class="page" lang="ru">
  <body class="page-body">
    <header></header>
    <main></main>
    <footer></footer>

</html>
```

```
...body-  
</html>
```

```
.page {  
  height: 100%;  
}  
  
.page-body {  
  min-height: 100%;  
  display: grid;  
  grid-template-rows: min-content 1fr min-content;  
}
```

Пример колонок в контенте страницы

```
<main class="page-catalog">  
  <h1 class="page-title">  
    Средства для ухода  
  </h1>  
  <ul class="breadcrumbs">  
    <!-- ... -->  
  </ul>  
  <section class="filters">  
    <!-- ... -->  
  </section>  
  <section class="catalog">  
    <!-- ... -->  
  </section>  
</main>
```

```
.page-catalog {  
  display: grid;  
  grid-template-columns: 180px 1fr;  
  align-content: start;  
}  
  
.page-catalog .page-title {  
  grid-column: 1 / -1;  
}  
  
.page-catalog .breadcrumbs {
```

```
.page-category .list-items {  
  grid-column: 1 / -1;  
}
```

Для остальной раскладки используются флексы: навигация, карточки, пагинация, хлебные крошки и другие.

```
.site-navigation {  
  display: flex;  
  flex-wrap: wrap;  
  width: 620px;  
}
```

Использование гридов для основных структурных блоков и флексов для всех остальных — это рекомендация, а не требование. Главное — разумно использовать возможности и флексов, и гридов.

^ 57. В CSS отсутствует `!important`.

Допускается использование `!important` при обосновании его необходимости в комментарии.

^ 58. Подключены правильные шрифты, их размеры, цвет и толщина равны соответствующим параметрам в макетах и техническом задании.

Необходимо, чтобы в личном проекте использовались именно те шрифты, которые указаны в макете. При этом их размеры, цвета, и жирность должны быть равны соответствующим параметрам в макетах и техническом задании.

^ 59. Нестандартные шрифты подключены локально. Формат шрифтов должен быть `woff2` и `woff`.

Все нестандартные шрифты должны быть подключены в CSS из папки `fonts` с помощью правила `@font-face`. Подключаемые шрифты должны быть в формате `woff2` и `woff`.

Подключение шрифтов с помощью внешних ресурсов, таких как Google Fonts

подключение шрифтов с помощью внешних ресурсов, таких как @font-face, считается ошибкой.

Перечисление шрифтов при подключении должно начинаться с woff2, а следом за ним уже woff. Например:

```
@font-face {  
  font-family: "Oswald";  
  src: url("../fonts/oswaldregular.woff2") format("woff2"),  
       url("../fonts/oswaldregular.woff") format("woff");  
  font-weight: 400;  
  font-style: normal;  
  font-display: swap;  
}
```

В таком случае woff скачиваться не будет, если браузер поддерживает woff2.

▲ Б10. Указаны альтернативные варианты шрифта и тип семейства в конце перечисления `font-family`.

Альтернативный веб-безопасный шрифт и тип семейства необходимо указывать для того, чтобы в случае отсутствия основного шрифта изменения внешнего вида шрифтов на странице были минимальны.

Порядок шрифтов следующий:

1. Основной шрифт;
2. Веб-безопасный;
3. Тип шрифта.

Список веб-безопасных шрифтов можно [посмотреть здесь](#).

Важно: альтернативный веб-безопасный шрифт должен быть такого же типа, что и основной, но подбирать максимально похожий шрифт не нужно.

Верно: указан альтернативный веб-безопасный шрифт и его тип семейства.

```
body {  
  font-family: "PT Sans Narrow", "Arial", sans-serif;  
}
```



```
/* Кому-то нравится Arial, кому-то Tahoma */
body {
  font-family: "PT Sans Narrow", "Tahoma", sans-serif;
}
```

Неверно: указан только основной шрифт.

```
body {
  font-family: "PT Sans Narrow";
}
```

Неверно: указан только основной шрифт и тип семейства, альтернативный веб-безопасный шрифт отсутствует.

```
body {
  font-family: "PT Sans Narrow", sans-serif;
}
```

Неверно: Times New Roman — шрифт с засечками, а основной шрифт — без засечек.

```
body {
  font-family: "PT Sans Narrow", "Times New Roman", sans-serif;
}
```

- ▲ **Б11.** При наполнении контентом (как в макете) элементы каждой версии страницы (мобильной, планшетной и десктопной) соответствуют макету.

Вёрстка проверяется на соответствие макету только на указанных вьюпортах:

— **Седона:**

- 320px — мобильная версия,
- 768px — планшетная версия,
- 1200px — десктопная версия.

— **Кэт энерджи:**

- 320px — мобильная версия,

- **768px** — планшетная версия,
- **1440px** — десктопная версия.

— **Мишка:**

- **320px** — мобильная версия,
- **768px** — планшетная версия,
- **1150px** — десктопная версия.

Размеры контентной области определены крайними направляющими в макетах. Области, которые выходят за крайние направляющие (например, фоновые изображения и интерактивные карты), должны подстраиваться под ширину экрана/вьюпорта и упираться в его границы.

Важно: на любом брейкпоинте не должно быть горизонтальной прокрутки страницы.

При проверке допускаются:

- Вертикальная погрешность не более 10 пикселей, горизонтальная погрешность не более 5 пикселей;
- Различия в отображении шрифтов, связанные со сглаживанием на различных платформах.

Для проверки страниц на соответствие макету следует пользоваться расширением для браузеров [Pixel Perfect](#), прозрачным окном из [Zeplin.io](#) или библиотекой [Pixel Glass](#). Проверка должна осуществляться при 100% масштабе системы.

CSS-препроцессор [Свернуть все](#)

^ [B12](#). Использован CSS-препроцессор.

Писать CSS-код можно только с помощью CSS-препроцессора. Можно использовать: Less или Sass.

Стилевой код, написанный без препроцессора, является ошибкой.

^ [B13](#). Код стилей должен быть разбит на несколько частей.

Используя CSS-препроцессор, нужно разделить код таким образом, чтобы каждый блок был в своём файле. Для каждого БЭМ-блока нужно создать отдельный файл стилей

стилей.

Исключение возможно для минимума глобальных стилей (стилизация по тегам, они должны быть описаны в одном файле), для CSS-нормализатора и для селекторов модульной сетки.

Все файлы должны импортироваться в главный стилевой файл, который будет компилироваться в CSS-файл.

Вариант **верного** разбиения файла стилей:

```
less/  
  normalize.less  
  grid.less  
  blocks/  
    block1.less  
    block2.less  
    ...  
  style.less
```

Вариант **неверного** разбиения файла стилей:

```
less/  
  normalize.less  
  global.less  
  index-page.less  
  header.less  
  photo-page.less  
  style.less
```

Вариант **неверного** содержания файла `header.less` со стилями блока. Внутри файла с блоком `header` находятся стили другого блока — `logo`:

```
.header {  
  padding: 20px;  
}  
  
.header__top {  
  background-color: #cccccc;  
}
```

```
.logo {  
  width: 200px;  
}
```

Адаптивность [Свернуть все](#)

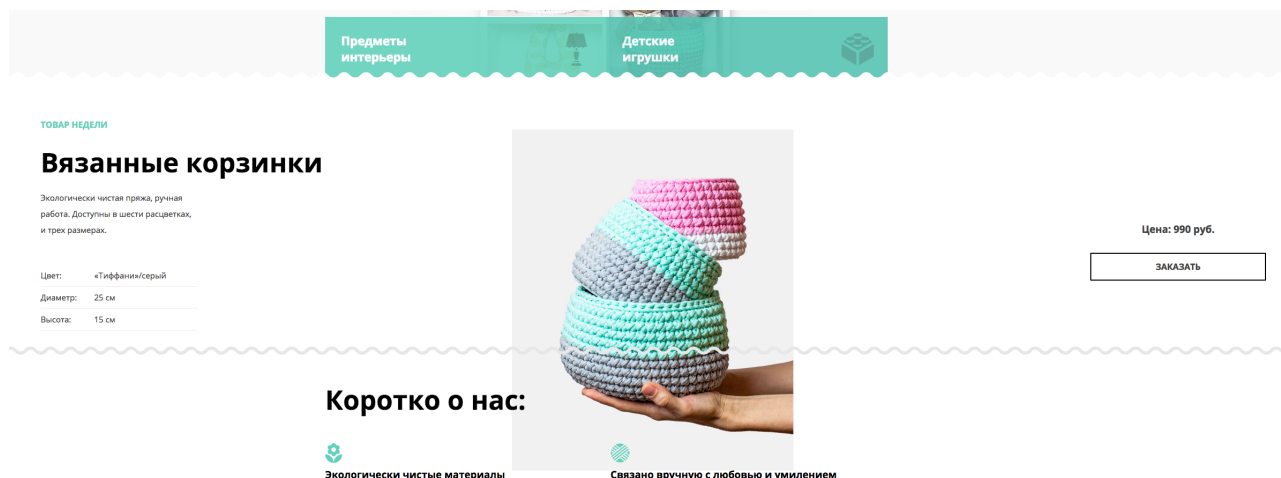
- ^ **Б14.** Выполнена вёрстка трёх состояний каждой страницы: мобильной, планшетной и десктопной.

Сетка в промежутках между основными точками может быть фиксированной или резиновой.

Важно: вне указанных точек вёрстка не должна «разваливаться»:

- все элементы продолжают располагаться по сетке, заданной в макете;
- блоки не выпадают из родителей и из общего центрирующего блока;

Неверно:



Верно:

ТОВАР
НЕДЕЛИ

Вязаные корзинки

Экологически чистая пряжа, ручная работа. Доступны в шести расцветках, и трех размерах.

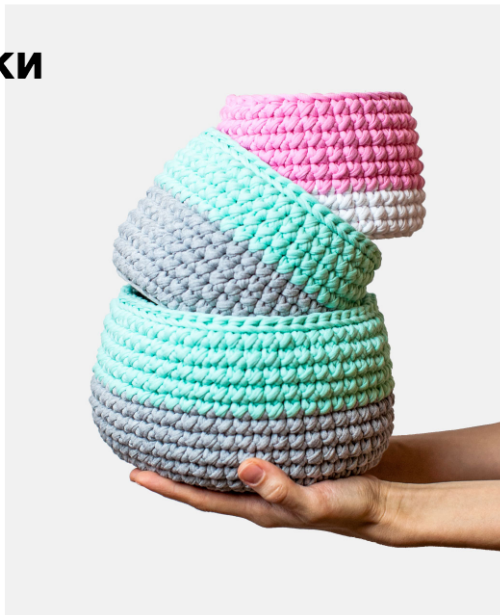
Цвет: «Тиффани»/серый

Диаметр: 25 см

Высота: 15 см

Цена: 990 руб.

ЗАКАЗАТЬ



^ B15. В разметке есть правильный вьюпорт тег.

Адаптивность должна работать на планшетах и мобильных устройствах. То есть не только при изменении ширины окна браузера.

^ B16. Для микросеток использованы флексы или гриды.

Следующие компоненты должны быть реализованы с помощью флексов или гридов:

— Седона:

- разделы формы, поля и подписи формы на странице отзыва;
- раскладка фотографий на странице «Фото и видео»;
- блок описания фотографии (название, автор, лайки) на странице «Фото и видео»;
- видеоплеер на странице «Фото и видео».

— Мишка:

- шапка сайта (лого, основное меню, поиск, корзина, новые поступления, распродажа, доставка);
- раскладка товаров и блок описания товара (название, описание, цена,

корзина) на странице «Каталог товаров».

— **Кэт энеджи:**

- блок «Живой пример» на главной странице;
- разделы формы, поля и подписи формы на странице «Подбор программы»;
- раскладка товаров на странице «Каталог продукции».
- блок «Дополнительные товары» на странице «Каталог продукции»

^ **Б17.** Выполнена ретинизация растровой графики.

Графика должна быть подготовлена как минимум для двух вариантов экранов: с обычной и двойной плотностью пикселей. Обязательны для ретинизации следующие компоненты:

— **Седона:**

- задний фон в промоблоке под главным меню на всех страницах;
- фоны (или изображения) в промоблоках «Настоящий городок» и «Там есть Мост Дьявола» на главной странице;
- контентные изображения на странице «Фото и видео».

— **Мишка:**

- задний фон в промоблоке на главной странице;
- изображение товара недели на главной странице;
- задний фон в блоке отзывов на главной странице;
- контентные изображения товаров в каталоге.

— **Кэт энеджи:**

- задний фон и изображение товара в промоблоке под главным меню на главной странице;
- изображения котов в блоке «Живой пример» на главной странице;
- контентные изображения товаров в каталоге;
- задний фон в блоке «Дополнительные товары» в каталоге.

^ **Б19.** Логотип должен адаптироваться.

Логотип должен адаптироваться к мобильному, планшетному и десктопному состоянию проекта.

^ Б20. Выполнено кадрирование контентных изображений.

С помощью тега `<picture>` выполнено кадрирование контентных изображений:

— **Седона:**

- на странице «Фото и видео».

— **Кэт энерджи:**

- на странице «Каталог продукции».

— **Мишка:**

- на странице «Каталог товаров».

Кадрирование в данном случае подразумевает загрузку разных фотографий на разном разрешении экрана.

Например:

```
<picture>
  <source media="(min-width: 1200px)" srcset="img/photo-1-desktop.jpg">
  <source media="(min-width: 768px)" srcset="img/photo-1-tablet.jpg">
  
</picture>
```

Графика [Свернуть все](#)

^ Б18. Форматы графики.

Выбран подходящий формат изображений. Например:

- JPEG для фотографий;
- SVG для векторных изображений;
- PNG для всех прочих.

Зачем нужен этот критерий

Использование подходящих форматов изображений важно для того, чтобы изображения отображались без погрешностей и имели оптимальный размер при загрузке.

^ Б21. «Нежная» ретинизация.

При ретинизации изображений учитывается плотность пикселей устройства пользователя и возвращаются разные изображения. Для контентных изображений должен быть использован `srcset`, а для фонов — медиавыражения.

Например, ретинизация контентного изображения:

```

```

Ретинизация фонового изображения:

```
.page-header__logo {
  background-color: #000000;
  background-image: url("img/bg-header@1x.jpg");
}

@media (min-resolution: 2dppx) {
  .page-header__logo {
    background-image: url("img/bg-header@2x.jpg");
    background-size: 1338px auto;
  }
}
```

Б22. Процесс сборки личного проекта настроен с помощью Gulp.

Б23. Все зависимости проекта должны быть указаны в файле `package.json`. Команда `npm ci` должна установить всё необходимое для того, чтобы сборка проекта работала.

^ Б24. Сборка проекта запускается командами `build` и `start`:

- `npm run build` — запускает сборку проекта;
- `npm run start` — запускает сборку проекта `build`, а после запускает `server`.

Также при сборке не должно возникать никаких ошибок.

^ **Б25.** В результате сборки должна получаться папка `build` со всеми необходимыми файлами.

Особенности сборки:

- папка `build` должна находиться в корне проекта;
- папка `build` должна автоматически создаваться в момент сборки;
- в папке `build` должна быть папка `css` с минифицированным стилевым файлом с префиксами;
- в папке `build` должна быть папка `fonts` в которой лежат только шрифты в формате `woff` и `woff2`;
- в папке `build` должна быть папка `img` в которой лежат только оптимизированные изображения и собранные спрайты;
- HTML-файлы проекта должны лежать в корне папки `build`.

^ **Б26.** Папка `build` со всем её содержимым не должна попадать в Гитхаб.

Это значит, что папка `build` должна находиться в файле `.gitignore`.

Разное [Свернуть все](#)

^ **Б27.** Вёрстка идентично отображается в последних версиях браузеров Chrome, Firefox, Safari.

При проверке этого критерия необходимо смотреть на размеры и расположение блоков, внешнее сходство с макетом. Проверять работу анимации, если такая имеется. Допускаются небольшие отличия в отображениях шрифтов.

Важно: обязательно проверить вёрстку в браузерах: Chrome, Firefox, Safari (только для пользователей macOS).

Пользователям Windows тестировать вёрстку в Safari не надо

Б28. Единообразное написание и форматирование кода в HTML, файлах CSS-препроцессора и JavaScript (включая файлы автоматизации).

Критерий рассматривает единообразие в написании кода:

- используется один тип кавычек: в одном языке (HTML, CSS или JS) должны использоваться только кавычки определённого типа (двойные или одинарные). Однако между языками тип кавычек может отличаться;
- если размер отступа в два пробела, то таким он должен быть везде;
- названия классов должны быть оформлены единообразно;
- свойства, которые поддерживают несколько наборов значений (например, множественные фоны и множественные тени, нужно делать в одном формате: однострочном или многострочном).

Для поддержания единообразия в коде используйте [EditorConfig](#).

Важно: этот критерий учитывает именно единообразие, а не стиль написания и форматирования кода. Единообразие кодовой базы помогает всем участникам команды работать в одинаковых условиях.

Правильно

Код написан в одном стиле.

```
<section class="news">
  <h2 class="news-title">Новости</h2>
</section>

<section class="news">
  <h2 class="news-title">Галерея</h2>
</section>
```

Неправильно

Одинаковые элементы разметки имеют разные отступы и переносы.

```
<section class="news-title">
  <h2 class="news-title">
    Новости
  </h2>
</section>
```

```
<section class="news-title"><h2 class="news-title">Галерея</h2></section>
```

Правильно

Цвет в разных блоках используется в одинаковой нотации.

```
.news {  
  color: #000000;  
}  
  
.news-title {  
  color: #ff0000;  
  background-color: #ffffff;  
}
```

Неправильно

Один и тот же цвет написан в разных нотациях.

```
.news {  
  color: red;  
}  
  
.news-title {  
  color: rgb(255, 0, 0);  
  background-color: #ffffff;  
}
```

- ^ B29. Отсутствует транслит в названиях классов, атрибутах, переменных CSS-препроцессора, названиях примесей и так далее.

Для названий необходимо использовать английские слова и термины.

Верно: используются только английские слова.

```
.login-button {}  
.container {}  
.footer {}
```

Неверно: транслит с русского языка.

```
.knopka-vxoda {}  
.kontainer {}  
.podval {}
```

^ **Б30.** Мобильное меню должно быть работоспособным при отключённом JavaScript.

При реализации без JS:

1. Меню должно быть открыто по умолчанию и находиться в потоке (то есть не перекрывать остальной контент).
2. Кнопки, которые работают на JS, должны быть скрыты.
3. В комментариях HTML-файлов должны быть указаны классы (или класс), которые переключают состояния меню и внешний вид кнопки.

В HTML- и CSS-коде должны быть предусмотрены элементы и правила для трёх возможных состояний:

1. Открытое меню в потоке при отключённом JavaScript.
2. Открытое меню вне потока при включённом JavaScript.
3. Закрытое меню.

Скрипт открытия и закрытия меню писать необязательно. Но если он написан, то работать должен так:

1. При инициализации скрипта меню должно убираться из потока (если в макете открытое меню не в потоке) и закрываться.
2. При клике на гамбургер меню открывается, появляется кнопка-закрытие, убирается кнопка-гамбургер.
3. При клике на закрытие меню закрывается, убирается кнопка-закрытие, появляется гамбургер.

^ **Б31.** Проект соответствует техническому заданию.

Техническое задание каждого проекта можно посмотреть на [странице проектов](#).

В некоторых пунктах есть необязательные реализации, которые выполняются по желанию.

Обратите внимание, что:

- должна быть сделана вёрстка всех модальных окон, но реализовывать открытие/заккрытие необязательно;
- должна быть сделана разметка всех слайдов, но оживлять их переключение в слайдере необязательно.

При конфликте технического задания и критериев, считать техническое задание более приоритетным.

Поиск по материалам

Git

[Все материалы](#)

В самом начале



- ☐ [Пройдите опрос](#)
- ☐ [Укажите персональные данные](#)
- ☐ [Изучите регламент](#)
- ☐ [Прочитайте FAQ](#)
- ☐ [Добавьте свой Гитхаб](#)
- ☐ [Выберите наставника](#)
- ☐ [Создайте проект](#)

Мой наставник



[Выбрать наставника](#)

Работа с наставником

У вас осталось **10** из 10 консультаций.

[История](#)



Практикум

Тренажёры

Подписка

Для команд и компаний

Учебник по PHP

Курсы

HTML и CSS. Профессиональная вёрстка сайтов

HTML и CSS. Адаптивная вёрстка и автоматизация

JavaScript. Профессиональная разработка веб-интерфейсов

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

Профессии

Фронтенд-разработчик

JavaScript-разработчик

Фулстек-разработчик

Услуги

Работа наставником

Для учителей

Стать автором

Остальное

Написать нам

Мероприятия

Форум

Соглашение

Конфиденциальность

Сведения об образовательной организации

Лицензия № 4696



Участник

© ООО «Интерактивные обучающие технологии», 2013–2023

