

Blade Agency



Инструкция по вёрстке



Полезные ссылки

[Главная](#) / [Моё обучение](#) / [Blade Agency](#) /

Инструкция по вёрстке

Этап 1. Подготовка.

На первом этапе нужно установить на компьютер удобный редактор кода и создать структуру проекта: файл `index.html`, папку `css` с файлами `style.css` и `normalize.css`, а также папки для картинок и шрифтов. И всё, можно начинать вёрстку!

[Стартовый шаблон](#)

Этап 2. Разметка.

На этом этапе нужно создать разметку страницы в файле `index.html`. Изображения можно пока не подключать, мы подключим их на этапе работы с графикой.

[Пример готового результата на этом этапе](#)

Этап 3. Базовая стилизация.

На этом этапе мы добавим на страницу базовые стили. Стили для расположения крупных блоков пока что можно не делать.

[Пример готового результата на этом этапе](#)

Этап 4. Работа с графикой.

Главная задача этого этапа — экспортировать всё необходимое из макета в Figma и подключить изображения в разметке. SVG-изображения рекомендуется собрать в спрайт и подключить первым элементом в `body`. Декоративные изображения можно экспортировать, но подключать пока что не нужно.

[Пример готового результата на этом этапе](#)

Этап 5. Сетки.

Добавляем крупные структурные сетки. В вёрстке используется подход desktop-first, то есть сайт прежде всего корректно отображается на больших разрешениях экрана.

[Пример готового результата на этом этапе](#)

Этап 6. Декоративные элементы.

Всё готово, чтобы достилизовать всю страницу. Добавляем мелкие сетки, подключаем кастомные шрифты, фоновые изображения и оформляем остальные декоративные элементы, которые ещё не стилизовали раньше.

[Пример готового результата на этом этапе](#)

Этап 7. Адаптивные сетки и декоративные элементы.

С помощью медиа-выражений добавляем стили для адаптации вёрстки под более маленькие разрешения.

[Пример готового результата на этом этапе](#)

Этап 8. Адаптивная графика.

Подключаем в разметке и стилях адаптивные изображения для разных девайсов и экранов с разной плотностью пикселей.

В HTML используем элемент `picture`, с помощью которого подключаем картинки для разных разрешений экрана, для экранов с разной плотностью пикселей, а также webp-варианты картинок для поддерживающих браузеров.

В CSS подключаем картинки для экранов с двукратной плотностью пикселей и для разных разрешений с помощью медиа-выражений.

[Пример готового результата на этом этапе](#)

Этап 9. Добавление JavaScript.

Добавляем JavaScript-код, который оживляет слайдер, всплывающие окна, блок переключения языков, меню, а также подключает полифилл для тега `<picture>`, включает плавный скролл к якорям страницы и адаптирует таблицу.

Финальный результат

Полезные ссылки



Практикум

Тренажёры

Подписка

Для команд и компаний

Учебник по PHP

Профессии

Фронтенд-разработчик

React-разработчик

Фулстек-разработчик

Бэкенд-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-интерфейсов

JavaScript. Архитектура клиентских приложений

React. Разработка сложных клиентских приложений

PHP. Профессиональная веб-разработка

PHP и Yii. Архитектура сложных веб-сервисов

Node.js. Разработка серверов приложений и API

Анимация для фронтендеров

Вёрстка email-рассылок

Vue.js для опытных разработчиков

Регулярные выражения для фронтендеров

Шаблонизаторы HTML

Алгоритмы и структуры данных

Анатомия CSS-каскада

Блог

С чего начать

Шпаргалки для разработчиков

Отчеты о курсах

Информация

Об Академии

О центре карьеры

Остальное

Написать нам

Мероприятия

Форум