



HTML и CSS. Адаптивная вёрстка и автоматизация

Уровень 2, с 21 ноября 2022 по 30 января 2023

Меню курса

[Главная](#) / [3. Препроцессоры и автоматизация](#) /

3.15. Стилизация

Информация по проекту

 Проект ещё не выбран

 Репозиторий ещё не создан

Работа над заданием

Чтобы выполнить задание и отправить его на проверку, необходимо [выбрать наставника](#).

« На расширенном тарифе достаточно стилизовать главную страницу.

1. Убедитесь, что ветка `master` вашего форка имеет актуальное состояние: прошлое задание полностью выполнено, принято наставником и обновлено в `master` вашего форка.
2. В форке проекта создайте ветку `module3-task1`. В файле `source/less/style.less` или `source/sass/style.scss` добавьте CSS-код:

```
body { background: red; }
```

3. В терминале перейдите в папку проекта и запустите команду `npm run start` и проверьте как собрался ваш проект. А затем нажмите `Ctrl + C`.
4. Удостоверьтесь, что после компиляции в папке `css` появился файл `style.css`. Подключите этот файл на свои страницы — их фон должен стать красным. Если сработало, то сборка работает.

5. В терминале в папке проекта запустите команду `npm start`. Внесите изменения в файл препроцессора, например, поменяйте фон страниц на синий, и сохраните файл. Если страница перезагрузилась и изменения на страницах видны, то отслеживание изменений работает.

6. Создайте структуру препроцессорных файлов:

- создайте каждому БЭМ-блоку отдельный препроцессорный файл;
- импортируйте файлы БЭМ-блоков в файл `source/less/style.less` или `source/sass/style.scss`;
- вспомогательные файлы с переменными, примесями и так далее создавайте на своё усмотрение.

7. Начните верстку вашего проекта с помощью препроцессора:

- задайте фоновые цвета и фоновые изображения для блоков;
- параметры текста: шрифты, размеры шрифта, межстрочные расстояния, цвет текста;
- добавьте внутренние и внешние отступы и обводки для крупных сеточных блоков.

Мелкие декоративные элементы верстать не нужно.

8. Создайте пулреквест из ветки `module3-task1` вашего форка в мастер-репозиторий и нажмите кнопку «Отправить на проверку» в интерфейсе курса.

« При выполнении задания ориентируйтесь на критерии:

- Б7. В CSS отсутствует `!important`
- Б12. Использован CSS-препроцессор
- Б13. Код стилей должен быть разбит на несколько частей
- Б28. Единообразное написание и форматирование кода в HTML, файлах CSS-препроцессора и JavaScript (включая файлы автоматизации)
- Б29. Отсутствует транслит в названиях классов, атрибутах, переменных CSS-препроцессора, названиях примесей и так далее
- Д4. Для стилизации не использованы `#id`
- Д6. Для блока, у которого есть фоновое изображение, прописан фоновый цвет, который соответствует преобладающему цвету изображения (пока изображение не загружено, страница выглядит похоже на макет)
- Д8. Запрещено использовать цветовые функции для изменения цветовых значений в коде

- Д9. Примеси не используются для генерации правил с вендорными префиксами
- Д10. Вложенность селекторов не больше двух уровней
- Д11. Родительский селектор & используется только для псевдоэлементов, псевдоклассов и модификаторов
- Д12. Не используются расширения (extend)
- Д16. Нет глобальных стилей тегов
- Д19. Код соответствует правилам в EditorConfig
- Д20. Код соответствует правилам в Stylelint

⚠ Если вы обнаружили ошибку или неработающую ссылку, выделите ее и нажмите Ctrl + Enter

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

