

Построение сеток на гридах по макету

Выполнен на 0%

Введение

Теория

Базовая теория

Описание сетки

Явные и неявные координаты грид-элементов

Грид-области

Именованные грид-линии

Грид-интервал

Размеры в гриде

Выравнивание в гриде

Углублённая теория

Многослойность в гриде

Повторы в гриде

Сокращённая запись

Продвинутая теория

Поведение автоматически размещаемых элементов

Определение диапазона размеров

Автоматическое заполнение грида

Дополнительный материал

Инструменты для работы с гридами

Баги и ограничения

Методика построения сеток на гридах

Кейс 1, лёгкий уровень

Кейс 2, лёгкий уровень

Кейс 3, лёгкий уровень

Кейс 4, лёгкий уровень

Кейс 5, средний уровень

Кейс 6, средний уровень

Кейс 7, средний уровень

Кейс 8, средний уровень

Кейс 9, сложный уровень

Кейс 10, сложный уровень

Главная / Моё обучение / Построение сеток на гридах по макету / Теория / Дополнительный материал

Баги и ограничения

Гриды, несмотря на относительную новизну технологии, — отлично описанная система. Её реализовали и выпустили в современных браузерах в рекордные по меркам CSS сроки — за пару лет.

Но, к сожалению, в браузерной реализации гридов есть и баги, которые пока не исправлены:

- Про то, что `gap` не идеально работает с процентными значениями, мы рассказывали в материале про грид-интервалы. Для внешних и внутренних отступов грид-элементов также лучше не использовать размеры в процентах, так как браузеры их по-разному рассчитывают.
- Элемент `<fieldset>` не может быть грид-контейнером в *Google Chrome*, и это большое упущение, так как вынуждает искать обходные пути при вёрстке. Кнопки `<button>` тоже не могут быть грид-контейнерами, но это может быть полезно реже.
- Многострочное поле ввода `<textarea>` сворачивается до нулевого размера, когда в него вводят текст, правда, это происходит только в *Google Chrome* на *MacOS*.

Полный список багов можно посмотреть [здесь](#).

Также пока что не везде работают `subgrid`: это возможность распространять влияние грида не только на дочерние элементы, но и на элементы более глубокой вложенности.

Пример, в котором бы пригодился `subgrid`:

```
<article class="grid"> <!-- это грид-контейнер -->
  <h1></h1>           <!-- это грид-элемент -->

  <ul>                <!-- это грид-элемент -->
    <li></li>           <!-- это уже не грид-элемент :( -->
    <li></li>
    <li></li>
  </ul>

  <p></p>              <!-- это грид-элемент -->
</article>
```

Если мы хотим сделать элементы списка `<ul>` грид-элементами, это не сработает: элементом грида будет только сам список как непосредственный дочерний элемент грид-контейнера.

Свойство `display: contents;`, применённое для `<ul>`, могло бы выступить в качестве решения этой проблемы, но у этого свойства пока что тоже есть проблемы с реализацией в браузерах.

Ознакомились со статьёй?

Сохранить прогресс

Инструменты для работы с гридами

Методика построения сеток на гридах



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