

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 / Углублённая теория /

Сокращённая запись

Большинство свойств грида можно записать очень развёрнуто или сокращённо. Например, грид-интервалы можно описать как `column-gap + row-gap`, а можно свойством `gap`, с одним или двумя значениями.

В целом, всю важную информацию о гриде можно записать в одно свойство `grid-template`: названия линий, количества колонок и их размеры, области и их размеры... Получится, например, так:

```
.grid {
  grid-template:
    [header-top]          /* имя первой линии рядов */
    "b b b s s s" 1fr    /* первый ряд и его высота */
    [header-bottom main-top] /* имена второй линии рядов */
    "c c c c m m" auto   /* второй ряд и его высота */
    [main-bottom]         /* имя последней линии рядов */
    /                     /* переходим к характеристикам столбцов */
    1fr 2fr;              /* их два, вот их размер */
}
```

Такая запись не позволяет использовать `repeat`, а также делает код менее удобным для чтения. Однако программы-минификаторы, для которых важна как можно более короткая запись, могут использовать именно такой формат.

Как вы уже видели выше, свойства `grid-row-start` и `grid-row-end` могут быть «упакованы» в свойство `grid-row`, а свойства `grid-column-start` и `grid-column-end` — в свойство `grid-column`.

Для задания координат есть ещё более «объединяющее» свойство — `grid-area`. Оно включает сразу четыре значения координат и расшифровывается так: `grid-area: grid-row-start / grid-column-start / grid-row-end / grid-column-end;`.

```
.grid {
  grid-area: 5 / 1 / 6 / 2;

  /* или то же самое */
  grid-row-start: 5;
  grid-column-start: 1;
  grid-row-end: 6;
  grid-column-end: 2;
}
```

Ознакомились со статьёй?

Сохранить прогресс

Повторы в гриде

Продвинутая теория



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