

## Вёрстка многослойных элементов интерфейса

Выполнен на 51%

Введение

Углублённая теория

Позиционирование

Как работает position: sticky

z-index

Многослойность с помощью CSS Grid Layout

Про vh, vw и другие единицы измерения

Словарь терминов

Методика вёрстки многослойных элементов

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, сложный уровень

Кейс 9, сложный уровень

Кейс 10, сложный уровень

Главная / Моя обучение / Вёрстка многослойных элементов интерфейса / Углублённая теория /

## Многослойность с помощью CSS Grid Layout

Как только вы установите для элемента `display: grid`, начнёт работать *CSS Grid Layout*. *CSS Grid Layout* определяет сетку элемента и то, как будут расположены в ней дочерние элементы. Если вкратце, процесс чем-то напоминает игру в морской бой. Сначала вы разлиновываете поле в двумерной системе координат (грид-контейнер). Затем в определённых квадратах рисуете корабли своей флотилии (грид-элементы): однопалубные, двухпалубные, трёхпалубные и так далее.

Подробнее о технологии *CSS Grid Layout* вы можете узнать из навыка «Построение сеток на гридах по макету».

Вложенные элементы грид-контейнера могут располагаться таким образом, что они перекрываются и накладываются друг на друга, как будто это стопка листов. В этом есть сходство со позиционированными элементами.

Рассмотрим пример:

```
<div class="grid">
  <header class="layer-1">
  </header>

  <main class="layer-2">
  </main>

  <aside class="layer-3">
  </aside>

  <footer class="layer-4">
  </footer>
</div>
```

```
.grid {
  display: grid;

  grid-template-columns: 100px 100px 100px 100px;
  grid-template-rows: 100px 100px 100px 100px;

  width: 400px;
  height: 400px;
}

.layer-1 {
  grid-column: 1 / -1;
  grid-row: 1 / 2;
}

.layer-2 {
  grid-column: 1 / -1;
  grid-row: 2 / -1;
}

.layer-3 {
  grid-column: 2 / 3;
  grid-row: 1 / 5;
}

.layer-4 {
  grid-column: 1 / -1;
  grid-row: 3 / 4;
}
```



Мы подкрасили блоки, чтобы была видна многослойность

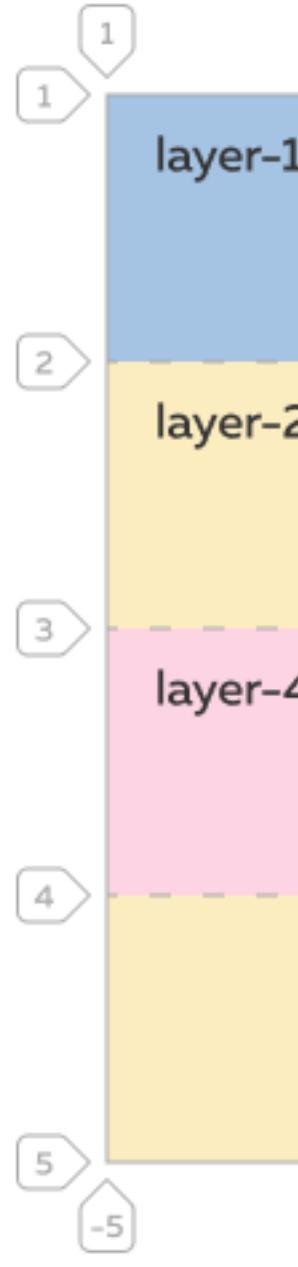
Как мы видим, элементы `layer-3` и `layer-4` расположены в тех же ячейках, что и `layer-1` и `layer-2` и наслаиваются друг на друга. Порядок слоя зависит от появления в разметке: первый слой ниже всех, затем идёт второй слой, выше — третий слой и над всеми остальными слоями — четвёртый.

Мы можем явно управлять очередностью слоёв в случае наложения. На наслаивающиеся грид-элементы действует свойство `z-index` (здесь есть сходство с абсолютно спозиционированными элементами). Чем выше `z-index`, тем выше элемент в «стопке». Если у двух элементов одинаковый `z-index`, то выше будет располагаться тот, что идёт следующим в разметке.

Дополним предыдущий пример и зададим `z-index` для третьего слоя:

```
.layer-3 {
  z-index: 1;
}
```

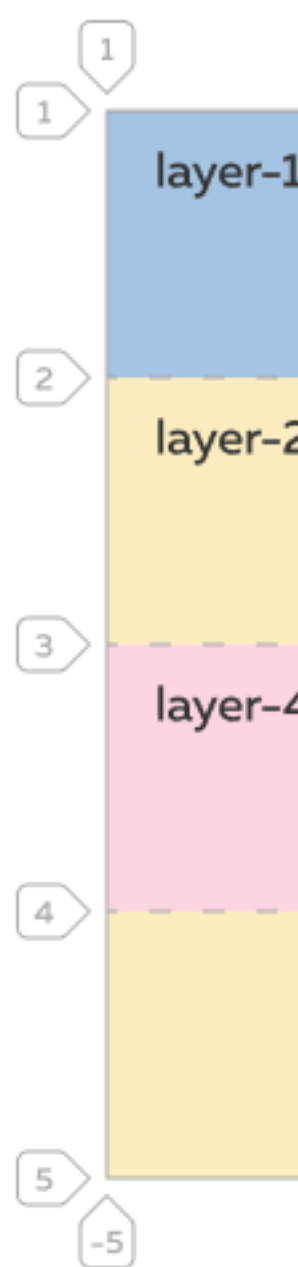
Порядок слоёв изменился. Третий слой стал выше всех: так как только ему задан `z-index`, он стал считаться выше всех остальных элементов.



Слой 3 имеет самое высокое значение `z-index` и поэтому перекрывает остальные слои в местах пересечений.

Дополним пример ещё:

```
.layer-1 {
  z-index: 2;
}
```



Теперь у первого слоя есть приоритет из-за самого большого значения `z-index`.

Порядок слоёв снова изменился: первый слой теперь выше всех, так как у него значение `z-index` больше, чем у остальных. Ниже него идёт третий слой, а за ним все остальные элементы, «отсортированные» в порядке появления в разметке.

Если многослойность можно добавить и с помощью позиционирования, и с помощью `display: grid`, что тогда выбрать? Выбор зависит от конкретного кейса и того, что вам нужно получить в итоге. Важное отличие этих двух способов заключается в следующем, в первом случае элемент «вырван» из потока, и соседние элементы о нём ничего не знают, во втором, элемент расположен в потоке и его размеры влияют на соседей.

— Что будет, если смешать `display: grid` и `position: absolute`?

Укажите грид-контейнеру `position: relative`, а элементу внутри него — `position: absolute`. Тогда элемент будет «вырван» из потока, а значения абсолютного смещения позиционирования рассчитаются от внешних краёв грид-области, в которую он был помещён.

```
.grid {
  position: relative;
  display: grid;

  grid-template-columns: 100px 100px 100px 100px;
  grid-template-rows: 100px 100px 100px 100px;

  width: 400px;
  height: 400px;
}

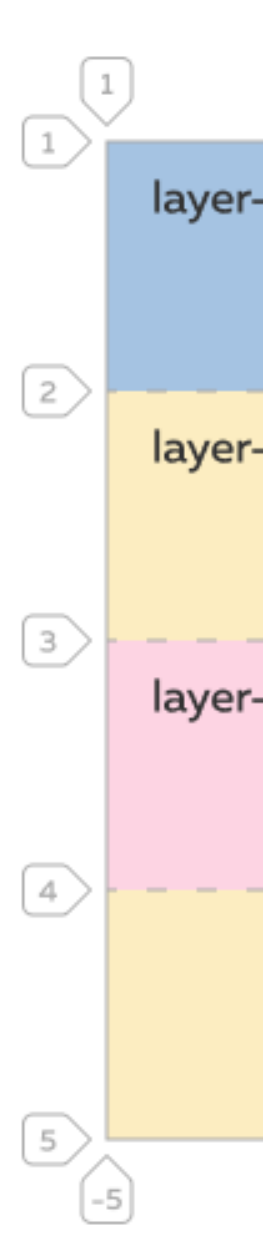
.layer-1 {
  grid-column: 1 / -1;
  grid-row: 1 / 2;
}

.layer-2 {
  grid-column: 1 / -1;
  grid-row: 2 / -1;
}

.layer-3 {
  grid-column: 2 / 3;
  grid-row: 1 / 5;

  position: absolute;
  top: 20px;
  left: 20px;
  width: 100px;
  height: 100px;
}

.layer-4 {
  grid-column: 1 / -1;
  grid-row: 3 / 4;
}
```



Абсолютно спозиционированный элемент внутри грид-контейнера

< z-index

Про vh, vw и другие единицы измерения >