Grid Css

Grid – это css модуль, который представляет возможность встроенной сетки. На данный момент grid устоялся и реализован во всех крупных браузеров, за исключением IE для которого нужды префиксы.

**Введение в сетку.**

*Контейнер* – это родитель всех элементов сетки. Для его создания используется display: block и именно в нем указываются все *свойства контейнеров*.

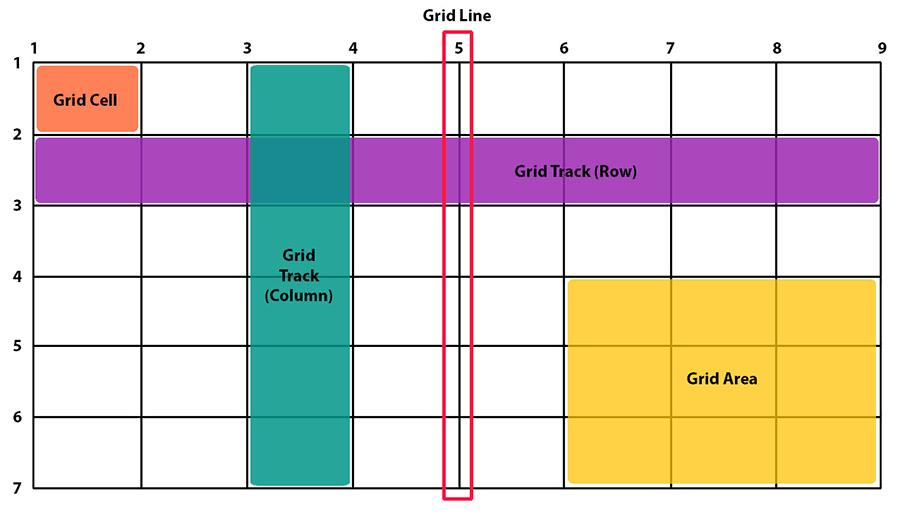
*Элементы* – это все, что внутри контейнера. Для них используют *свойства элементов*.

*Линии сетки* (Grid Line) – это линия, которая служит для разделения колонок и строк между собой. Первые называются *вертикальные*, а вторые *горизонтальные*.

*Ячейка* (Grid Cell) – это единица сетки, которая расположена между двумя соседними вертикальных и горизонтальных линий.

*Трек сетки* (Grid Track) – это пространство ячеек между двумя линиями.

*Область сетки* (Grid Area) – это пространство ячеек внутри четырех линий.



**Создание сетки grid-template.**

<div class="grid-container">

<div class="item">1</div>

<div class="item">2</div>

<div class="item">3</div>

<div class="item">4</div>

<div class="item">5</div>

<div class="item">6</div>

</div>

Для начала зададим grid-container свойство display: grid. Сейчас они расположились друг за другом. Для того чтоб расположить их сетку нужно создать саму сетку.

grid-template-columns: 30% 30% 30%;

Теперь у нас все содержимое расположится в три колонки. Кроме процентов есть px, em, rem, vh есть особая величина *фракции* (fr):

grid-template-columns: 1fr 0.5fr 2fr;

Кроме того есть полезная функция repeat(кол-во, значения), позволяет повторить:

grid-template-columns: repeat (2, 1fr 0.5fr);

Функция minmax(мин, макс) позволяет указать возможный размер. Если размер зайдет через предел, то он перестанет расширяться или сжиматься.

Функцию minmax используют, вместе с auto-fill иди auto-fit:

grid-template-columns: repeat (auto-fit, minmax (100px, 1fr));

Значение auto-fit повторяет до тех пор, пока – это возможно, т.е. даже если осталось свободное место он не выделит для них ячейки. Значение auto-fill, похоже, но оставляет пустое пространство в виде ячеек. В общем, этот «прием» используется, когда нужно выложить карточки товаров и т.д.

Кроме работы с колонками можно задать такие же значения и строке через свойство grid-template-columns:

grid-template-columns: 1fr 1fr;

grid-template-rows: 150px 200px;

**Промежутки gap.**

У линий, возможно, задать размер, который будет отображаться как промежуток рядами и строками. Для этого существуют свойства row-gap и column-gap, а также их колесо gap.

**Создание трека.**

Для трека используют свойства grid-column-start, grid-column-end и grid-row-start, grid-row-end.

Свойства grid-column-start и grid-row-start – определяют линию начало, а grid-column-end и grid-row-end – определяют линию, на которой надо закончить.

Специальное значение span позволяет указать, сколько нужно линий, а не конкретную линию. Для обоих пар существуют колеса grid-column и grid-row.

.item:first-child {

grid-column: 1 / span 4;

grid-row: 1 / span 2;

}

Также существует колесо, которое позволяет записать все значения сразу:

grid-area: 1 / span 4 / 1 / span 2

**Значения по-умолчанию.**

В примеры выше у нас из-за того что элемент вышел за предел сетки могут образоваться дополнительные линии или колонки, все они образуют *неявную сетку*.

Свойства grid-auto-rows и grid-auto-columns позволяют задать значение для дополнительных линий или колонок:

grid-auto-columns: 1fr;

grid-auto-rows: 1fr;

**Область сетки.**

Главное преимущество grid – это область сетки. Он настолько простой, что можно просто взглянуть, например [D:\alex-kudryavtsev\GridAreaExample](file:///D:\alex-kudryavtsev\GridAreaExample)

**Выравнивание**

Чтоб выровнить колонки относительно контента используется свойство justify-content. А для выравнивания строк используется свойство align-content. Существует колесо для них place-content:

place-content: justify-content align-content;

Все значения:

* center – выравнивание по центру
* start – примыкает элементы к началу
* end – примыкает элементы к концу
* stretch – значение по-умолчанию
* space-around
* space-between
* space-evenly

Чтоб выровнить элементы внутри ячейки используются значения justify-items и align-items, существует колесо place-items:

place-items: justify-item align-items;

Все значения:

* center
* start
* end
* stretch

Кроме того можно выровнить отдельный элемент свойствами justify-self, align-self и place-self.