**Grid Layout**

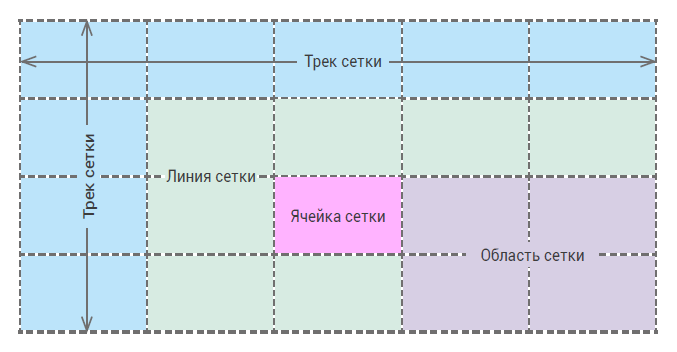
**Основные понятия и компоненты сетки**

**Контейнер сетки** — это блок, который создает область с сеткой и устанавливает контекст форматирования по типу сетки, то есть дочерние элементы располагаются согласно правилам компоновки сетки, а не блочной компоновки.

Контейнер сетки определяется при помощи ***display: grid*** или ***display: inline-grid***. Как только мы это сделаем, все прямые дочерние элементы контейнеры автоматически станут элементами сетки.

Когда мы создаем контейнер сетки, сетка по умолчанию имеет один столбец и одну строку, которые занимают полный размер контейнера.

Контейнеры сетки не являются блочными контейнерами, поэтому некоторые CSS-свойства в контексте контейнера сетки работать не будут.



1. ***Grid Container***

***Контейнер сетки*** — главный роди- тельский элемент для всех элементов сетки.

1. ***Grid Item***

***Элемент сетки*** — дочерний элемент (прямой потомок) контейнера сетки.

1. ***Grid Line***

***Линия сетки*** — линия, которая разделяет элементы сетки. Она может быть горизонтальной или вертикальной.

1. ***Grid Track***

***Трек сетки*** — пространство между двумя параллельными линиями сетки. Можно считать, что это строка или столбец сетки.

1. ***Grid Cell***

***Ячейка сетки*** — пространство между двумя соседними строками и двумя соседними столбцами линий сетки, аналогично ячейке в таблице. Это область, в которую можно что-то поместить. Ячейка является наименьшей единицей сетки, на которую можно ссылаться при позиционировании элементов сетки.

1. ***Grid Area***

***Область сетки*** — любое пространство между четырьмя линиями сетки может состоять из любого количества ячеек. Область сетки может быть не больше одной ячейки или размером со все ячейки сетки.

**Создание Grid-контейнера**

**свойство display:**

* ***grid*** — генерирует контейнер сетки уровня блока.
* ***inline-grid*** — генерирует контейнер сетки уровня строки.
* ***subgrid*** — если контейнер также является элементом другой сетки (вложенный контейнер), можно указать, что параметры для текущей сетки следует взять из родительской.

**строки и столбцы:**

* ***grid-template-rows*** — генерирует количество строк.
* ***grid-template-columns*** — генерирует количество столбцов.

***grid-template-columns: 100px 10em 30%;  
grid-template-rows: 10em 100px 100px;***

**Гибкие размеры треков: единица измерения fr**

**Fr** — единица измерения длины, которая позволяет создавать гибкие треки. При определении размеров треков, общий размер фиксированных строк или столбцов вычитается из доступного пространства контейнера-сетки. Оставшееся пространство делится между строками и столбцами с гибкими размерами пропорционально их коэффициенту. Если сумма размеров всех гибких треков меньше 1, они будут занимать только соответствующую часть оставшегося пространства, а не расширяться, чтобы заполнить все пространство полностью.

Если ширина или высота контейнера-сетки не заданы, треки сетки масштабируются по их содержимому, сохраняя при этом пропорции.

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

**Автоматические размеры**

Значение ***auto*** ориентируется на содержание элементов сетки одного трека. Минимальный размер трека рассматривается как минимальный размер элемента сетки, то есть ***Grid***использует свойства ***min-width или minheight***. Максимальный размер трека устанавливается из расчета на значение свойства ***max-content***. Трек может растягиваться при использовании свойств ***lign-content и justify-content***a.

***grid-template-columns: auto auto auto;  
grid-template-rows: auto auto auto;***

**Функция repeat**

Функция ***repeat()*** позволяет создавать фрагменты списка треков, которые повторяются.

***grid-template-columns: 50px repeat(2, 1fr 2fr);***

**размеры линий сетки:**

* ***grid-column-gap*** — устанавливает ширину вертикальной линии сетки (промежуток между столбцами).
* ***grid-row-gap*** — устанавливает высоту горизонтальной линии сетки (промежуток между строками).
* ***grid-gap*** — сокращенная форма записи для и ***grid-row-gap + gridcolumn-gap***.

Размеры треков сетки можно задавать с помощью различных значений, используя относительные единицы длины, например, ***em, vh, vw; абсолютные единицы длины px; и проценты %***. Размеры в % высчитываются по ширине или высоте контейнера-сетки.

***grid-column-gap: 10px;  
grid-row-gap: 10px;  
grid-gap: 20px 10px;***

**Пример: Grid-контейнер**

* **grid**
* **grid-template**
* **grid-template-gap**
* **grid-fr**
* **grid-fr-gap**
* **grid-fr-repeat**
* **grid-position-span**
* **grid-position-name**
* **grid-position-name-span**
* **grid-area**

**Позиционирование элементов и Grid-линии**

Позиционирование элементов сетки определяется расположением линий сетки и диапазоном сетки — количеством треков, которые занимает элемент. По умолчанию элемент сетки занимает одну ячейку трека на каждой оси.

**Позиционирование элементов сетки с помощью номеров линий сетки**

* ***grid-column-start*** — задает начальную линию расположения элемента в столбце.
* ***grid-column-end*** — задает конечную линию расположения элемента в столбце.
* ***grid-row-start*** — задает начальную линию расположения элемента в строке.
* ***grid-row-end*** — задает конечную линию расположения элемента в строке.

**Сокращенные формы записи для позиционирования**

* ***grid-column*** — задает начальную и конечную линию расположения элемента в столбце.
* ***grid-row*** — задает начальную и конечную линию расположения элемента в строке.
* ***grid-area*** — задает начальную и конечную линию расположения элемента в строке и столбце.

***grid-column: 1 / 4;  
grid-row: 1 / 2;  
grid-area: 2 / 3 / 4 / 4;***

**Объединение ячеек с помощью ключевого слова span**

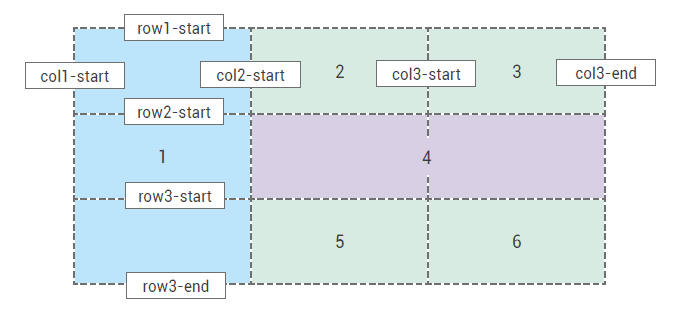
Ключевое слово span дает возможность объединять ячейки без указания начальной и конечной линии трека.

**Именованные линии сетки**

Имена линиям можно задать явно, используя свойства ***grid-templaterows и grid-template-columns*** или неявно — путем создания именованных областей сетки с помощью свойства ***grid-template-areas.***

Имя линии может быть любым. При задании в значении свойства, его следует взять в квадратные скобки. Ключевое слово ***span*** не может использоваться в качестве названия линии сетки.

***grid-template-columns: [col1-start] 100px [col2-start] auto [col3-start] auto [col3-end];  
grid-template-rows: [row1-start] auto [row2-start] auto [row3-start] auto [row3-end];***



Теперь для позиционирования элементов в сетке можно использовать имена линий сетки.

***grid-column: col1-start / col3-end;  
grid-row: row1-start / row3-end;***

**Именованные строки и столбцы и ключевое слово span**

Строкам и столбцам можно также задать одно и то же имя, а потом задавать расположение элементов по этим именам и использовать ключевое слово ***span*** для объединения ячеек.

В следующем примере создаются 3 строки и 3 столбца, и для каждой строки и столбца задается одинаковое название:

***grid-template-columns: [col] auto [col] 150px [col] auto;  
grid-template-rows: [row] auto [row] 150px [row] auto;  
grid-column: col 1;  
grid-row: row / span 3;***

**Области сетки**

Мы можем создать именованные участки сетки для размещения содержимого. Для этого мы сначала определяем элементы нашей разметки для сетки, используя свойство ***grid-area.***

**Выравнивание в Grid**

**Выравнивание всей сетки**

Иногда размер сетки может быть меньше, чем размер самого контейнера сетки. Такое может случиться, если заданный размер строк и столбцов является фиксированным (в px). В этом случае мы можем установить выравнивание для сетки в контейнере с помощью свойств ***justify-content и align-content.***

**Свойство justify-content выравнивает всю сетку относительно вертикальной оси.**

* ***start*** — выравнивает сетку по левому краю контейнера.
* ***end*** — выравнивает сетку по правому краю контейнера.
* ***center*** — выравнивает сетку по центру.
* ***stretch*** — растягивает сетку на всю ширину контейнера.
* ***space-around*** — размещает единицу пустого пространства между элементами и половину единицы по краям.
* ***space-between*** — размещает единицу пустого пространства между элементами, без отступов по краям.
* ***space-evenly*** — размещает единицу пустого пространства между элементами, включая края сетки.

**Свойство align-content выравнивает сетку относительно горизонтальной оси.**

* ***start*** — выравнивает сетку по верхнему краю контейнера.
* ***end*** — выравнивает сетку по нижнему краю контейнера.
* ***center*** — выравнивает сетку по центру.
* ***stretch*** — растягивает сетку на всю высоту контейнера.
* ***space-around*** — размещает единицу пустого пространства между элементами и половину единицы по краям.
* ***space-between*** — размещает единицу пустого пространства между элементами, без отступов по краям.
* ***space-evenly*** — размещает единицу пустого пространства между элементами, включая края сетки.

**Выравнивание всех элементов сетки**

Иногда возникает необходимость выровнять контент внутри ячеек. Мы можем установить выравнивание для контента с помощью свойств ***justify-items и align-items.*** Данные значения применяются ко всем ячейкам сетки.

**Свойство justify-items выравнивает содержимое внутри ячейки относительно вертикальной оси.**

* ***start*** — выравнивает содержимое по левому краю ячейки.
* ***end*** — выравнивает содержимое по правому краю ячейки.
* ***center*** — выравнивает содержимое по центру.
* ***stretch*** — растягивает содержимое на всю ячейку (значение по умолчанию).

**Свойство align-items выравнивает содержимое внутри ячейки относительно горизонтальной оси.**

* ***start*** — выравнивает содержимое по верхнему краю ячейки.
* ***end*** — выравнивает содержимое по нижнему краю ячейки.
* ***center*** — выравнивает содержимое по центру.
* ***stretch*** — растягивает содержимое на всю ячейку (значение по умолчанию).

**Выравнивание отдельных элементов сетки**

Установить индивидуальное поведение для ячейки можно с помощью свойств ***justify-self и align-self.***

**Свойство justify-self выравнивает текущий элемент относительно вертикальной оси.**

* ***start*** — выравнивает элемент по левому краю зоны.
* ***end*** — выравнивает элемент по правому краю зоны.
* ***center*** — выравнивает элемент по центру зоны.
* ***stretch*** — растягивает элемент по горизонтали на все свободное место (значение по умолчанию).

**Свойство align-self выравнивает текущий элемент относительно горизонтальной оси.**

* ***start*** — выравнивает элемент по верхнему краю зоны.
* ***end*** — выравнивает элемент по нижнему краю зоны.
* ***center*** — выравнивает элемент по центру зоны.
* ***stretch*** — растягивает элемент по вертикали на все свободное место (значение по умолчанию).

Цвета:

.grid-item {

    background: #FFC0CB;

    border: 1px solid var(--color-dim-gray);

    padding: 10px;

}

.grid-item:nth-child(2) {

    background: #D8BFD8;

}

.grid-item:nth-child(3) {

    background: #DB7093;

}

.grid-item:nth-child(4) {

    background: #FFFF00;

}

.grid-item:nth-child(5) {

    background:#ADFF2F;

}

.grid-item:nth-child(6) {

    background: #98FB98;

}

.grid-item:nth-child(7) {

    background: #87CEFA;

}

.grid-item:nth-child(8) {

    background: #9370DB;

}

.grid-item:nth-child(9) {

    background: #C0C0C0;

}