



Лекция 8

Сетки. Позиционирование

Что мы уже знаем?

1. Flexbox представляет собой набор свойств, которые помогают нам строить сетки;
2. Флексовые боксы имеют две оси – главную и поперечную;
3. Флоаты никак не влияют на отображение флексов;
4. При указании контейнеру свойства **display: flex**, все его потомки становятся флексовыми боксами и получают значение **flex-direction: row**;
5. У флексов очень хорошая поддержка во всех современных браузерах;

Что мы уже знаем?

1. **display: flex (inline-flex);**
2. **flex-direction** – управляет направлением главной оси;
3. **justify-content** – выравнивает элементы вдоль главной оси;
4. **flex-wrap** – управляет переносом элементов;
5. **flex-flow** – сокращение для flex-direction и flex-wrap;
6. **align-items** – выравнивает элементы вдоль поперечной оси;
7. **align-self** – выравнивает элемент по отношению к контейнеру;

Еще свойства flexbox

1. **order** – определяет порядок элементов;
2. **flex-basis** – определяет базовый размер элемента;
3. **flex-grow** – определяет, то, насколько отдельный flex-блок может быть больше соседних элементов, если это необходимо;
4. **flex-shrink** – определяет, насколько flex-блок будет уменьшаться относительно соседних элементов внутри flex-контейнера в случае недостатка свободного места.

Позиционирование элемента

Относительное позиционирование

CSS-свойство **position** задает режим позиционирования элементов. Значением по умолчанию является **static**, которое означает «обычное позиционирование».

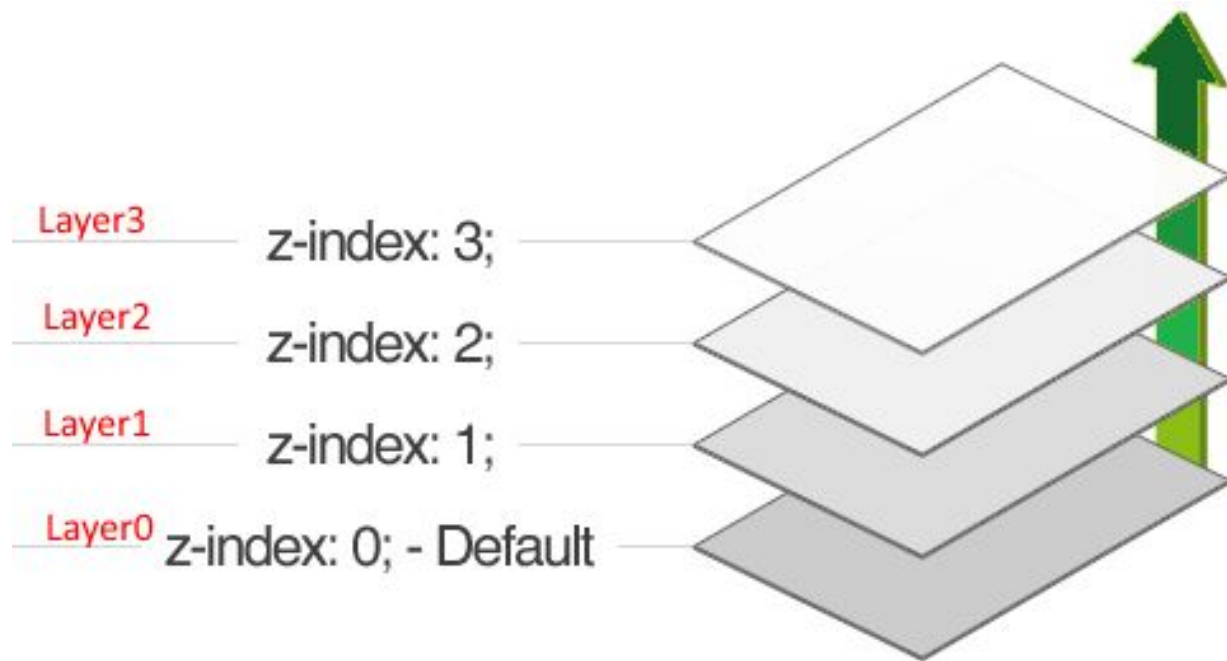
Значение **relative** обозначает «относительное позиционирование». Такой элемент остается в потоке документа, однако его можно перемещать при помощи свойств **top**, **bottom**, **right** и **left**.

Абсолютное позиционирование

Абсолютно позиционированные элементы (**position: absolute**) выпадают из потока документа. Место, которое они занимали, отдается соседним элементам. Ширина по умолчанию зависит от содержимого.

Z-index

Любые позиционированные элементы на веб-странице могут накладываться друг на друга в определенном порядке, имитируя тем самым третье измерение, перпендикулярное экрану. Каждый элемент может находиться как ниже, так и выше других объектов веб-страницы, их размещением по z-оси и управляет **z-index**. Это свойство работает только для элементов, у которых значение **position** задано как **absolute**, **fixed** или **relative**.





1. z-index: 10

The diagram consists of four overlapping squares on a light gray background. A red square is at the top left. A green square is positioned to the right of the red square, partially overlapping it. A cyan square is positioned below the red square, partially overlapping it. A blue square is positioned to the right of the cyan square, partially overlapping it. The blue square has a yellow border. Each square contains a numbered label indicating its z-index value.

2. z-index: 100

3. z-index: 150

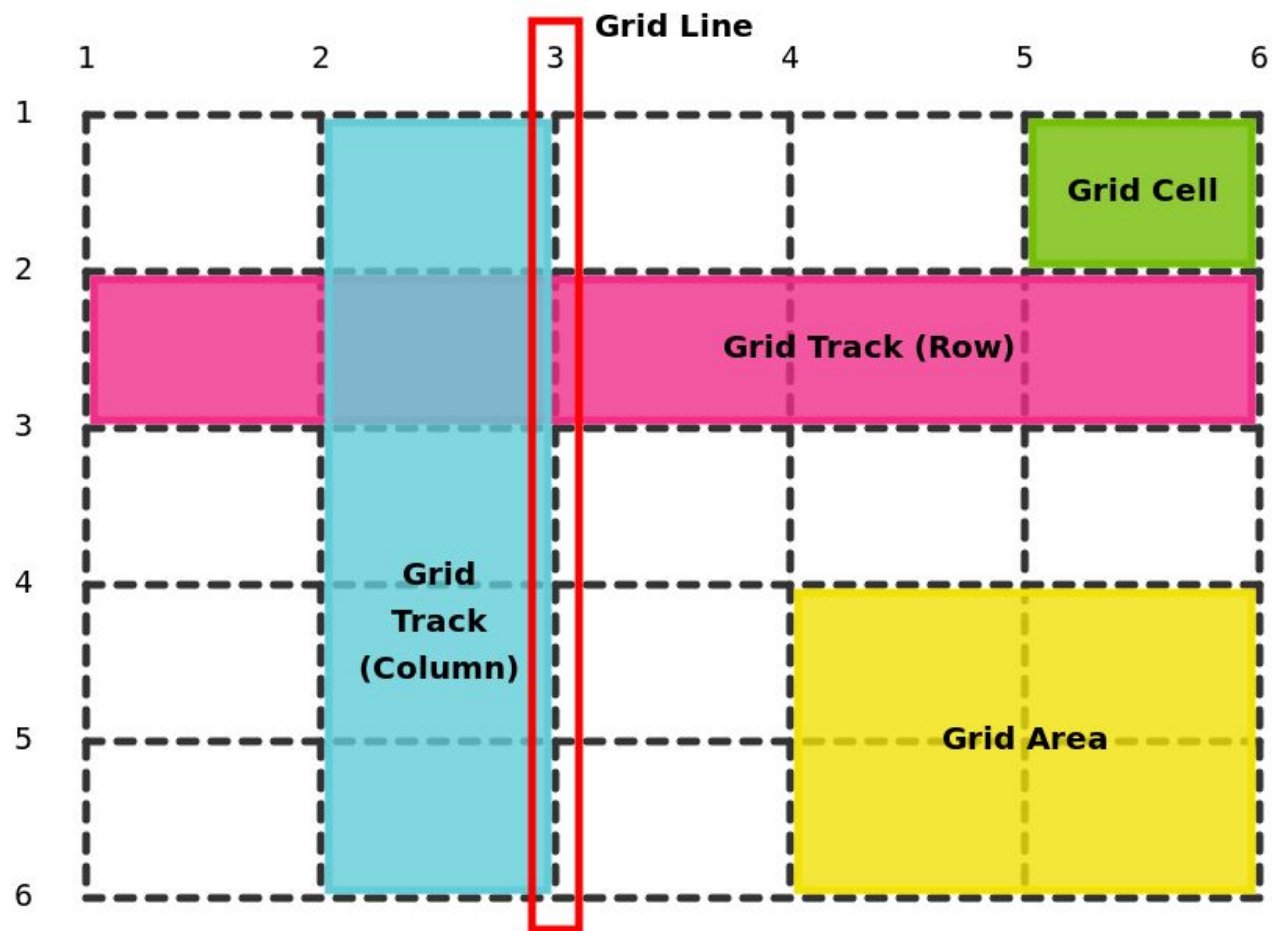
4. z-index: 50

CSS Grid

Что за штука?

CSS Grid это новая технология, которая позволяет удобнее строить сетки. Гриды не призваны заменить Flexbox.

Главное отличие — это то, что **Flexbox работает только в одном измерении**. Из этого следует, что мы можем размещать flex-элементы только вдоль главной оси или вдоль поперечной оси. CSS Grid **позволяет работать с разметкой в двухмерном пространстве** и выравнивать содержимое в обоих измерениях.



display: grid;

Задается элементу-контейнеру, дочерние элементы которого становятся гридовыми. Визуально сразу ничего не меняет (в отличии от **display: flex**).

В Firefox есть удобная функция, показывающая строки и колонки сетки.

```
.container {  
  background-color: #33d656;  
  border: 2px dotted #222;  
  display: grid;  
}
```

style.css:5

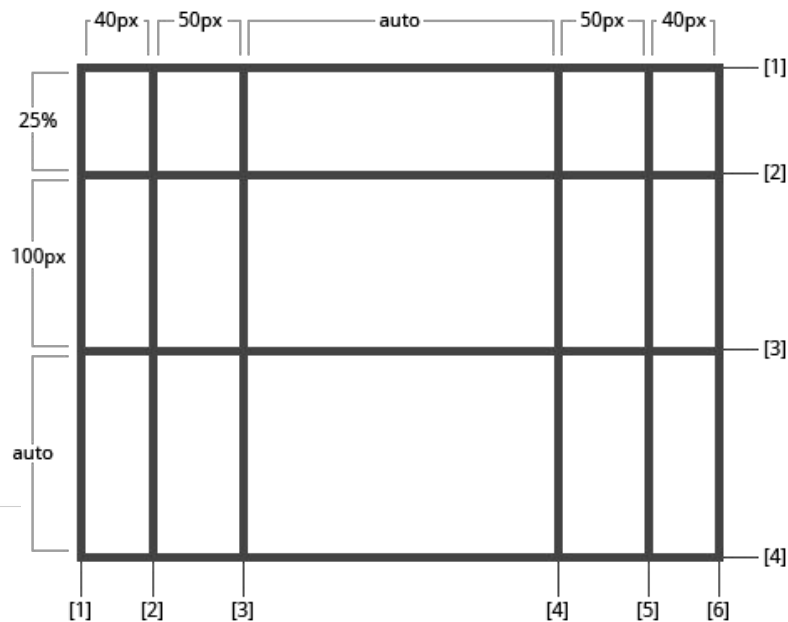


Создаем колонки и строки

Свойства **grid-template-columns** и **grid-template-rows**

используются для указания размеров колонок и строк.

```
.container {  
  grid-template-columns: 40px 50px auto 50px 40px;  
  grid-template-rows: 25% 100px auto;  
}
```



Указываем расстояние между ячейками

Свойство **grid-gap** задает расстояние между колонками и рядами. Также можно отдельно указать свойства **grid-column-gap** и **grid-row-gap**.

Выравнивание строк, столбцов и сетки

Свойство **justify-items** выравнивает содержимое вдоль оси **строки**.

Свойство **align-items** выравнивает содержимое вдоль оси **столбца**.

Свойства **justify-content** и **align-content** выравнивают **сетку** вдоль оси колонки и оси строки соответственно **по отношению к контейнеру**.

Выравнивание содержимого отдельных ячеек

Свойство **justify-self** выравнивает содержимое элемента вдоль оси **строки**.

Свойство **align-self** выравнивает содержимое элемента вдоль оси **столбца**.

Эти значения применяются к содержимому **внутри отдельного элемента**.

Управление порядком отображения

Свойство **grid-auto-flow** позволяет изменять порядок расположения элементов в контейнере.



grid-auto-flow: row;

grid-auto-flow: column;

grid-template-areas и **grid-area**

Свойство **grid-area** дает название элементу чтобы можно было ссылаться на него с помощью шаблона созданного через **grid-template-areas**.

Свойство **grid-template-areas** определяет шаблон сетки ссылаясь на имена областей, которые заданы с помощью свойства **grid-area**.