

Конспект «Микросетки. Продолжение». Раздел 1

Поперечная ось и свойство align-items (флекс)

Во флексах свойство `align-items` управляет расположением элементов на поперечной оси.

Поперечная ось идёт перпендикулярно главной оси и по умолчанию направлена сверху вниз:

Во флекс-контейнере свойство `align-items` может иметь следующие значения:

- `stretch` — значение по умолчанию; элементы растягиваются на всю высоту поперечной оси.
- `flex-start` — элементы сжимаются до содержимого и располагаются в начале поперечной оси (по умолчанию сверху);
- `flex-end` — элементы сжимаются до содержимого и располагаются в конце поперечной оси (по умолчанию снизу);
- `center` — элементы сжимаются до содержимого и располагаются по центру поперечной оси;

Поворот главной оси, свойство flex-direction

Главную ось поворачивают, чтобы сохранить внутри флекс-контейнера направление потока сверху вниз. За направление главной оси отвечает свойство `flex-direction`. По умолчанию у него значение `row` (ряд), но его можно изменить на `column` (колонка):

```
flex-container {  
  display: flex;  
  flex-direction: column;  
}
```

В этом случае главная ось будет направлена сверху вниз, а поперечная — слева направо. В результате флекс-элементы выстроятся сверху вниз.

По умолчанию флекс-элементы сжимаются по главной оси и растягиваются по поперечной. Таким образом, если главная ось направлена слева направо, то элементы сжимаются по горизонтали и растягиваются по вертикали.

Если же главная ось направлена сверху вниз, то сжатие происходит по вертикали, а растяжение — по горизонтали.

Получается, чтобы при поворнутой оси выровнять элемент по горизонтали, нужно задать ему выравнивание по поперечной оси.

Свойство `align-self` (флекс)

Свойство `align-self` задаётся флекс-элементу и говорит, как ему расположиться на поперечной оси. Значения у этого свойства такие же, как у `align-items`: `stretch` (значение по умолчанию), `flex-start`, `flex-end` и `center`.

```
.element {  
  align-self: flex-end;  
}
```

Свойство `order`

Чтобы изменить визуальный порядок элементов, удобно использовать свойство `order`. В качестве значения свойство принимает число, причём оно может быть как положительным, так и отрицательным. По умолчанию у всех элементов свойство `order` равно нулю.

```
.element {  
  order: 5;  
}
```

Элементы выстраиваются от меньшего значения `order` к большему. Если у нескольких элементов одинаковое значение, используется их порядок в разметке.

Свойство `order` работает только в грид- и флекс-контейнерах.

Отступы у флекс-элементов

Внешние отступы у соседних флекс-элементов складываются.

Чтобы не запутаться и получить именно те размеры, которые указаны в макете, верстальщики добавляют элементам внешние отступы только с одной стороны. Часто внешние отступы задают в направлении потока. Если элементы выстроены горизонтально, то отступ задают справа, а у последнего элемента обнуляют:

Если элементы выстроены вертикально, то отступ добавляют снизу. Исключение — самый последний элемент (например, подвал страницы), ему при необходимости задают отступ сверху:

В этом случае, даже если изменить порядок секций, они не слипнутся, а между ними не появятся лишние отступы.

Уменьшение изображений с сохранением пропорций

Если использовать картинку большего размера, чем родительский элемент, она выпадет из контейнера. Чтобы не допускать подобного, верстальщики добавляют картинкам такие стили:

```
img {  
  max-width: 100%;  
  height: auto;  
}
```

Свойство `max-width` задаёт максимальную ширину, а значение `100%` говорит, что элемент не должен становиться больше ширины родителя.

Свойство `height` задаёт высоту элемента. Значение `auto` используют, чтобы изображение не деформировалось и сохраняло свои пропорции. Если его не указать, то будет использовано значение атрибута `height` из разметки, и картинка, скорее всего, исказится.

Этот способ работает, потому что у CSS-свойств `max-width` и `height` приоритет выше, чем у атрибутов `width` и `height` в разметке.