Flexbox-вёрстка

normalize.css

Специальный файл, который применяет к HTML-элементам правила универсального отображения во всех браузерах.

Сайт проекта — https://meyerweb.com/eric/tools/css/reset/

flex-контейнер

Блок на странице, который содержит элементы. Внутри flex-контейнера элементы ведут себя необычно — они располагаются в линию по умолчанию и растянуты во всю высоту. Flex-контейнеры нужны, чтобы гибко настраивать поведение элементов внутри, например расставлять их, автоматически рассчитывая пространство между ними.

Такой контейнер создают правилом `display: flex;`

flex-элемент

Дочерние элементы flex-контейнера. Контейнер поступает со своими элементами как тренер с футболистами: размещает их по определённой схеме и прописывает поведение в меняющейся обстановке.

Свойства flex-контейнера

flex-direction

Направление внутри flex-контейнера. По умолчанию flex-элементы внутри контейнера отображаются в ряд слева направо в том порядке, в каком они прописаны в коде HTML-документа. Они следуют направлению потока. Flexbox-вёрстка позволяет преобразовать ряд в колонку и обратить порядок следования, направив его против потока. Для этого контейнеру придают свойство `flex-direction` (англ. direction, "направление"). Его возможные значения:

`flex-direction: row;` — обычный ряд (англ. row, "ряд"), элементы следуют в потоке, слева направо

`flex-direction: row-reverse;` — ряд в обратном порядке

`flex-direction: column;` — колонка

`flex-direction: column-reverse;` — колонка в обратном порядке

justify-content

Управление положением содержимого flex-контейнера. Сколько бы flex-элементов ни было в контейнере, остаётся ещё пустое пространство, которое можно распределять. Это делается в двух направлениях — вдоль оси, по которой расставлены элементы (т.е. вдоль строки/колонки), и поперёк.

Возможные значения свойства `justify-content`:

`flex-start` ("от начала flex-контейнера") установлено по умолчанию: сначала идут подряд flex-элементы, затем всё оставшееся пустое пространство

`flex-end` ("от конца flex-контейнера"): сначала идёт пустое пространство, а к концу контейнера прижаты элементы в порядке согласно значению свойства `flex-direction`

`center` ("по центру"): flex-элементы собраны посередине, а в начале и конце — две половины пустого пространства

`space-between` ("пространство между"): первый элемент прижат к началу flex-контейнера, последний — к концу, а остальные расставлены между ними с одинаковыми пустыми промежутками.

`space-around` ("пространство вокруг"): пустота разбита на одинаковой величины поля по обе стороны каждого flex-элемента. Все элементы расставлены ровно, разделены полосой величиной в два поля.

`space-evenly` ("делать интервалы равномерными"): все пустоты — и между элементами, и по краям — одинаковой величины.

Таблица значений flex-direction и justify-content

https://code.s3.yandex.net/web-developer/learning-materials/scheme_flex.pdf

align-items

Когда flex-элементы разной высоты выстроены в ряд, или столбец составлен из элементов разной ширины, их также можно выровнять по определённой линии. Предназначенное для этого свойство так и называется `align-items` . Эта линия проходит вдоль направления перечисления элементов. Так, если `flex-direction` задаёт расстановку в ряд, то линией может быть верхний край контейнера, нижний край, центральная ось и так называемая базовая линия — граница, где первый элемент разделяется на две части

Значения:

`stretch` ("растянуть") — установлено по умолчанию. Вместо выравнивания элементы растянуты от одного края контейнера до другого: в ряду сверху донизу, в колонке справа налево.

`flex-start` ("по началу flex-контейнера") — в простом ряду все элементы прижаты к верхнему краю контейнера, а в простой колонке к левому. Соответственно, в обращённом ряду (`flex-direction: row-reverse;`) выравнивание происходит по нижнему краю контейнера, а в обращённой колонке (`flex-direction: column-reverse;`) по правому.

`flex-end` ("по концу flex-контейнера") — в простом ряду все элементы прижаты к нижнему краю контейнера, а в простой колонке к правому.

`center` ("центрировать") — центральная ось контейнера делит каждый элемент пополам.

`baseline` ("по базовой линии") — базовые линии всех flex-элементов становятся продолжением базовой линии первого по порядку.

flex-wrap

Перенос элементов при сжатии окна браузера. По умолчанию flex-контейнер делает окно "резиновым": при сжатии или расширении окна просмотра браузера отдельные flex-элементы автоматически настраивают свою ширину или высоту так, чтобы поместиться в контейнер. Это поведение можно изменить так, чтобы при сжатии элементы переносились на новую строку или колонку.

Значения flex-wrap:

`wrap` — последние элементы заворачиваются (переходят) на новую строчку, если не умещаются на текущей

`wrap-reverse` ("перенос наоборот") позволяет устроить так, чтобы переносились не последние, а первые элементы строки/колонки.

`nowrap` — задано по умолчанию, отменяет переносы

Свойства flex-элементов

order

В правила тех flex-элементов, чей порядок следования нужно переопределить, вводится свойство `order` . Элемент, которому задано CSS-свойство `order: 1;` будет показан раньше элемента со свойством `order: 2;` (здесь могут быть любые целые числа, лишь бы одно из них было меньше другого). Элемент с отрицательным значением порядка (`order: -1;`) будет первым в контейнере, если у других элементов значения положительны или даже не заданы.

align-self

Иногда нужно, чтобы отдельный элемент не подчинялся общему правилу. На помощь приходит свойство `align-self` . Оно применяется непосредственно для уникального flex-элемента и принимает те же значения, что и `align-items`:

- `flex-start`,
- `flex-end`,
- `center`,
- `baseline`,
- `stretch`.

flex-basis

Иногда необходимо, чтобы один flex-элемент имел определенный размер, пока другие свойства flex-контейнера не начнут его сжимать или расширять. Для этого есть свойство `flex-basis` (от англ. "базис", базовый размер flex-элемента). Значение задаётся в пикселях.

Это ширина, если flex-элементы отображаются в ряд, и высота — если в колонку.

Дополнительно

Интерактивный сервис, позволяющий понять, как работают свойства flex-контейнера и flex-элементов

Flexbox Playground
https://demos.scotch.io/visual-guide-to-css3-flexbox-flexbox-playground/demos/