display:

Включает flex свойство для элемента. Под это свойство попадает сам элемент и вложенные в него элементы: затрагиваются только потомки первого уровня - они станут элементами flex контейнера.

* **flex** - элемент растягивается на всю ширину и имеет свое полное пространство среди окружающих блоков. Происходит перенос строк в начале и в конце блока.
* **inline-flex** - элемент обтекается другими элементами. При этом его внутренняя часть форматируется как блочный элемент, а сам элемент — как встроенный.

flex и inline-flex отличаются тем что по-разному взаимодействуют с окружающими элементами, подобно display:block и display:inline-block.

flex-direction:

Изменяет направление главной оси контейнера. Поперечная ось меняется соответственно.

* **row (default)** - направление элементов слева направо (>)
* **column** - направление элементов сверху вниз (v)
* **row-reverse** - направление элементов справа налево (<)
* **column-reverse** - направление элементов снизу вверх (^)

flex-wrap:

Управляет переносом непомещающихся в контейнер элементов.

* **nowrap (default)** - вложенные элементы располагаются в один ряд (при direction=row) или в одну колонку (при direction=column) независимо от того помещаются они в контейнер или нет.
* **wrap** - включает перенос элементов на следующий ряд, если они не помещаются в контейнер. Так включается движение элементов по поперечной оси.
* **wrap-reverse** - тоже что wrap только перенос будет не вниз, а вверх (в обратном направлении).

flex-flow: {direction wrap}

* Объединяет оба свойства flex-direction и flex-wrap. Они часто используются вместе, поэтому чтобы писать меньше кода было создано свойство flex-flow.
* flex-flow принимает значения двух этих свойств, разделенные пробелом. Или можно указать одно значение любого свойства.

justify-content:

Выравнивает элементы по основной оси: если direction=row, то по горизонтали, а если direction=column, то по вертикали.

* **flex-start (default)** - элементы будут идти с начала (в конце может остаться место).
* **flex-end** - элементы выравниваются по концу (место останется в начале)
* **center** - по центру (место останется слева и права)
* **space-between** - крайние элементы прижимаются к краям (место между элементами распределяется равномерно)
* **space-around** - свободное пространство равномерно распределяется между элементами (крайние элементы не прижимаются к краям). Пространство между краем контейнера и крайними элементами будет в два раза меньше чем пространство между элементами в середине ряда.
* **space-evenly** - тоже что space-around, только расстояние у крайних элементов до краев контейнера такое же как и между элементами.

align-content:

Выравнивает ряды, в которых находятся элементы по поперечной оси. То же что justify-content только для противоположной оси.

Т.е. если flex-direction: row, то это свойство будет выравнивать невидимые ряды по вертикали (¦). Тут важно заметить, что высота блока должна быть задана жестко и должна быть больше высоты рядов иначе сами ряды будут растягивать контейнер и любое их выравнивание теряет смысл, потому что между ними нет свободного места... А вот когда flex-direction: column, то ряды движется по горизонтали (>) и ширина контейнера почти всегда больше ширины рядов и выравнивание рядов сразу приобретает смысл...

Это свойство мало где нужно и чаще используется align-items (см.ниже).

* **stretch (default)** - ряды растягиваются заполняя строку полностью
* **flex-start** - ряды группируются в верхней части контейнера (в конце может остаться место).
* **flex-end** - ряды группируются в нижней части контейнера (место останется в начале)
* **center** - ряды группируются по центру контейнера (место останется по краям)
* **space-between** - крайние ряды прижимаются к краям (место между рядами распределяется равномерно)
* **space-around** - свободное пространство равномерно распределяется между рядами (крайние элементы не прижимаются к краям). Пространство между краем контейнера и крайними элементами будет в два раза меньше чем пространство между элементами в середине ряда.
* **space-evenly** - тоже что space-around, только расстояние у крайних элементов до краев контейнера такое же как и между элементами.

align-items:

Выравнивает элементы по поперечной оси внутри ряда (невидимой строки). Т.е. сами ряды выравниваются через align-content, а элементы внутри этих рядов (строк) через align-items и все это по поперечной оси. По главной оси такого разделения нет, там нет понятия рядов и элементы выравниваются через justify-content.

* **stretch (default)** - элементы растягиваются заполняя строку полностью
* **flex-start** - элементы прижимаются к началу ряда
* **flex-end** - элементы прижимаются к концу ряда
* **center** - элементы выравниваются по центру ряда
* **baseline** - элементы выравниваются по базовой линии текста

order:

* Позволяет менять порядок (позицию, положение) элемента в общем ряду.
* **По умолчанию: order: 0**
* По умолчанию элементы имеют order: 0 и ставятся в порядке их появления в HTML коде и направления ряда. Но если изменить значение свойства order, то элементы будут выстраиваться в порядке значений: -1 0 1 2 3 .... Например если одному из элементов указать order: 1, то сначала будут идти все нулевые, а потом элемент с 1.
* Так можно, например, первый элемент перекинуть в конец, при этом не меняя направление движения остальных элементов или HTML код.