

Курсы HTML и CSS для начинающих





Работа с макетом (flexbox)

Занятие 3





Разбор дизайн-макета страницы

Flexbox позволяет контролировать размер, порядок и выравнивание элементов по нескольким осям, распределение свободного места между элементами и многое другое.

Основные преимущества flexbox



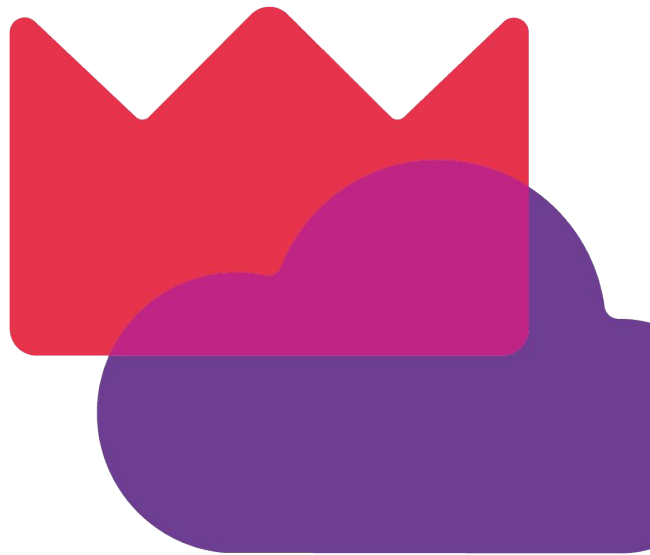
- 1 Элементы могут сжиматься и растягиваться по заданным правилам, занимая нужное пространство
- 2 Выравнивание по вертикали и горизонтали, базовой линии текста работает шикарно
- 3 Расположение элементов в html не имеет решающего значения. Его можно поменять в CSS



Основные преимущества flexbox



- 4 Элементы могут автоматически выстраиваться в несколько строк/столбцов, занимая все предоставленное место
- 5 Множество языков в мире используют написание справа налево **rtl (right-to-left)**, в отличии от привычного нам **ltr (left-to-right)**
- 6 Синтаксис CSS правил очень прост и осваивается довольно быстро

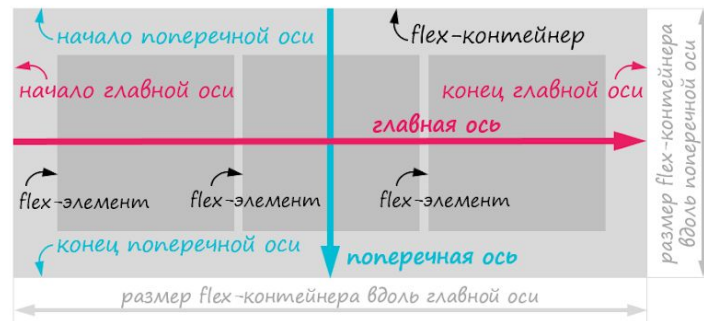


Основы Flexbox: контейнеры и элементы



Одним из основных понятий в **flexbox** являются оси.

- Главной осью flex-контейнера является направление, в соответствии с которым располагаются все его дочерние элементы.
- Поперечной осью называется направление, перпендикулярное главной оси.

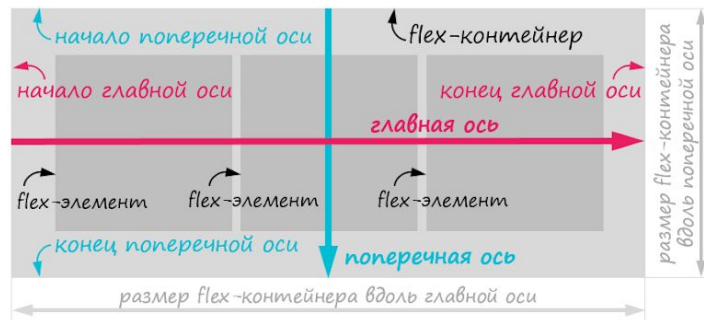


Основы Flexbox: контейнеры и элементы



Область документа, использующая flexbox, называется **flex контейнером**

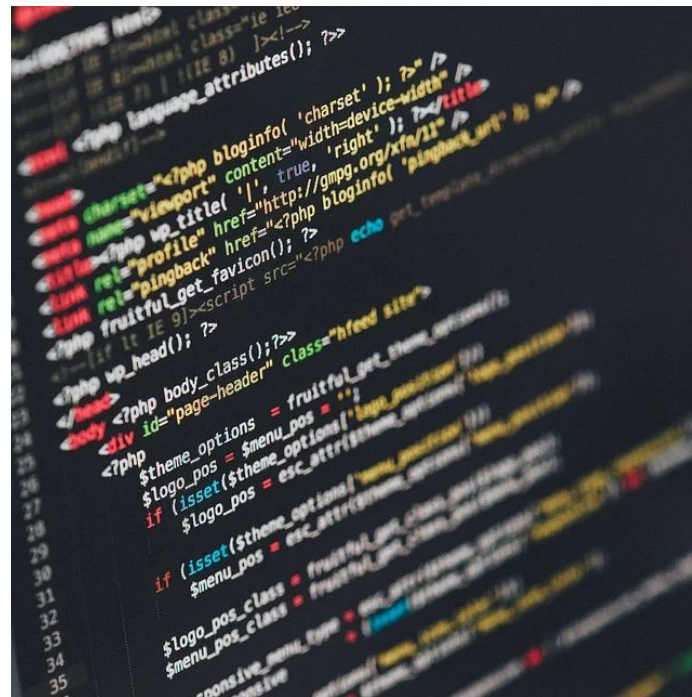
Чтобы создать flex контейнер, мы задаём значение *flex* или *inline-flex* для свойства `display` контейнера



Начальные значения



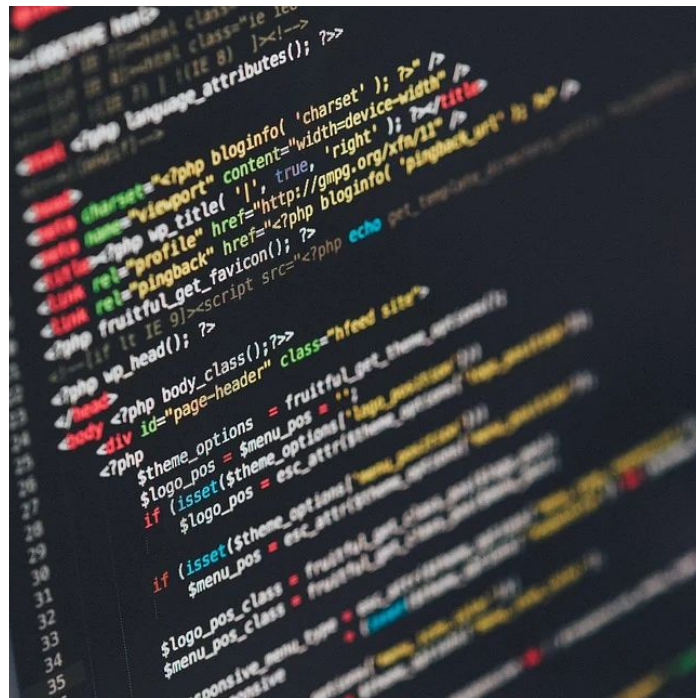
- Элементы отображаются в ряд (свойство **flex-direction** по умолчанию имеет значение **row**)
- Позиционирование элементов начинается от начала главной оси
- Элементы не растягиваются по основной оси, но могут сжиматься



Начальные значения



- Элементы будут растягиваться, чтобы заполнить размер побочной оси
- Свойству **flex-basis** задано значение auto
- Свойству **flex-wrap** задано значение nowrap

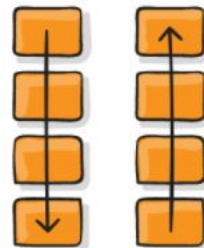
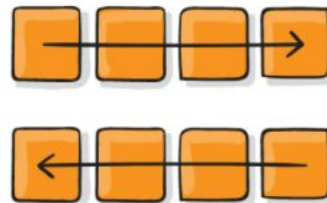


Свойства для родителя (flex контейнер)



Главная ось определяется свойством *flex-direction*, которая может принимать одно из следующих значений:

- row (по умолчанию): слева направо в ltr; справа налево в rtl
- row-reverse справа налево ltr; слева направо в rtl

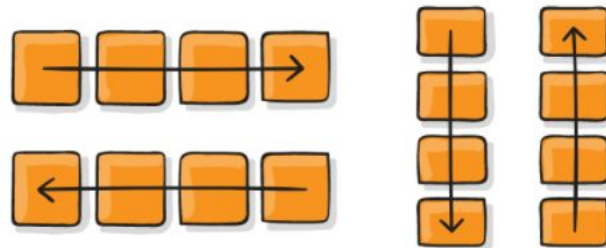


Свойства для родителя (flex контейнер)



Главная ось определяется свойством *flex-direction*, которая может принимать одно из следующих значений:

- column: так же, как и row, но сверху вниз
- column-reverse: то же самое, row-reverse, но снизу вверх





Побочная ось

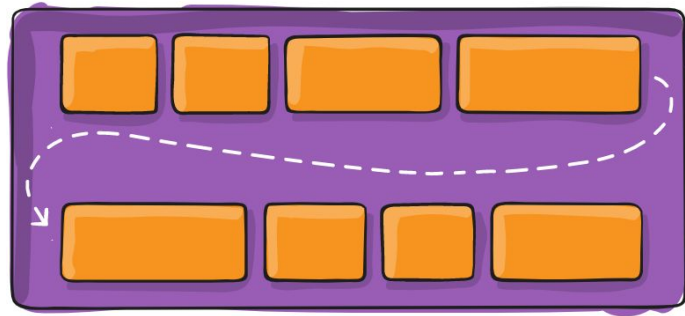
Побочная ось проходит перпендикулярно главной оси, поэтому, если свойство `flex-direction` (главная ось) задано как `row` или `row-reverse`, побочная ось будет проходить в вертикальном направлении.

Если свойство `flex-direction` задано как `column` или `column-reverse`, побочная ось будет проходить в горизонтальном направлении.

Flex-warp



- **nowrap (по умолчанию):** все flex элементы будут в одной строке
- **wrap:** flex-элементы будут перенесены на несколько строк сверху вниз
- **wrap-reverse:** flex-элементы будут перенесены на несколько строк снизу вверх





Justify-content

Это свойство определяет выравнивание
вдоль главной оси

Justify-content



- **flex-start (по умолчанию):** элементы сдвинуты в начало flex-direction направления
- **flex-end:** элементы сдвинуты ближе к концу flex направления
- **start:** элементы сдвинуты к началу writing-mode направления.
- **end:** элементы сдвинуты в конце writing-mode направления.

flex-start



flex-end



center



space-between



space-around



space-evenly



Justify-content



- **left:** элементы сдвинуты по направлению к левому краю контейнера, если это не имеет смысла flex-direction, тогда он ведет себя как start.
- **right:** элементы сдвинуты по направлению к правому краю контейнера, если это не имеет смысла flex-direction, тогда он ведет себя как start.
- **center:** элементы центрированы вдоль линии

flex-start



flex-end



center



space-between



space-around



space-evenly



Justify-content



- **space-between:** элементы равномерно распределены по линии; первый элемент находится в начале строки, последний элемент в конце строки
- **space-around:** элементы равномерно распределены по линии с одинаковым пространством вокруг них.space-evenly: элементы распределяются таким образом, чтобы расстояние между любыми двумя элементами (и расстояние до краев) было одинаковым.

flex-start



flex-end



center



space-between



space-around



space-evenly





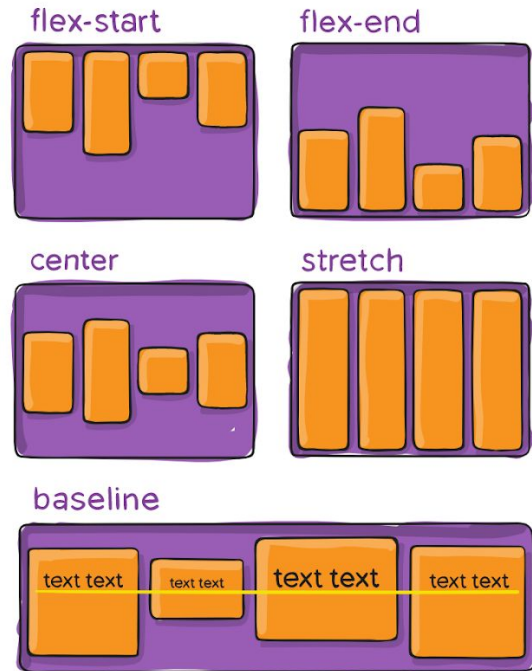
Align-items

Это свойство определяет поведение по умолчанию того, как flex элементы располагаются вдоль поперечной оси на текущей линии.

Align-items



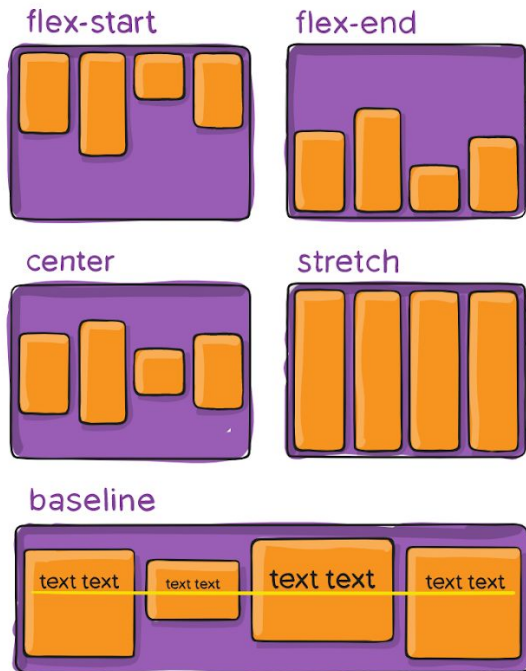
- **stretch (по умолчанию):** растягивать, чтобы заполнить контейнер (все еще соблюдаются min-width / max-width)
- **flex-start / start / self-start:** элементы размещаются в начале поперечной оси. Разница между ними невелика и заключается в соблюдении flex-direction правил или writing-mode правил.



Align-items



- **flex-end / end / self-end:** элементы располагаются в конце поперечной оси. Разница опять-таки тонкая и заключается в соблюдении flex-direction или writing-mode правил.
- **center:** элементы центрированы по поперечной оси
- **baseline:** элементы выровнены, по их базовой линии





Align-content

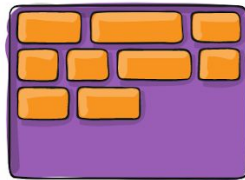
Это свойство выравнивает линии в пределах flex контейнера, когда есть дополнительное пространство на поперечной оси, подобно тому, как justify-content выравнивает отдельные элементы в пределах главной оси.

Align-content

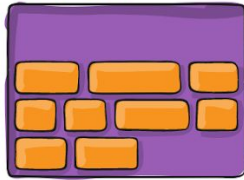


- **flex-start / start:** элементы, сдвинуты в начало контейнера. Более поддерживаемый flex-start использует flex-direction в то время как start использует writing-mode направление.
- **flex-end / end:** элементы, сдвинуты в конец контейнера. Более поддерживаемый flex-end использует flex-direction в то время как end использует writing-mode направление.

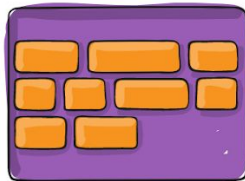
flex-start



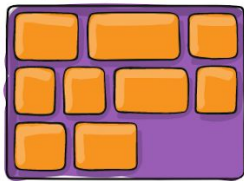
flex-end



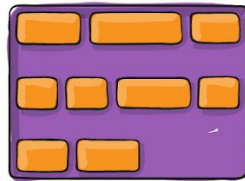
center



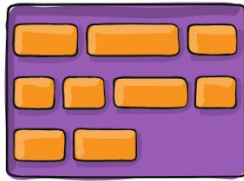
stretch



space-between



space-around

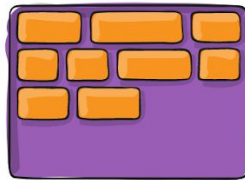


Align-content

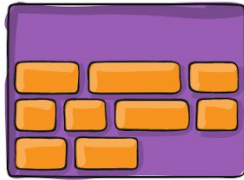


- **space-between:** элементы равномерно распределены; первая строка находится в начале контейнера, а последняя — в конце
- **space-around:** элементы равномерно распределены с равным пространством вокруг каждой строки
- **space-evenly:** элементы распределены равномерно, вокруг них одинаковое пространство

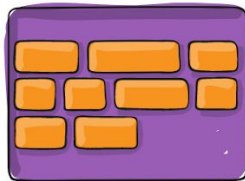
flex-start



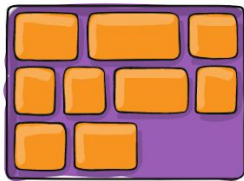
flex-end



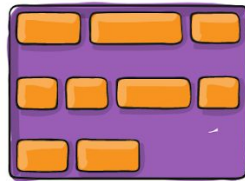
center



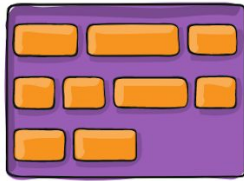
stretch



space-between



space-around

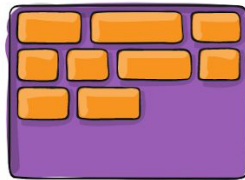


Align-content

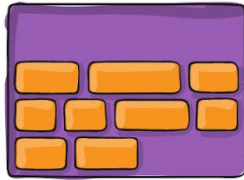


- **center:** элементы выровнены по центру в контейнере
- **stretch (по умолчанию):** линии растягиваются, чтобы занять оставшееся пространство

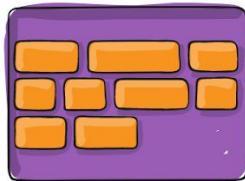
flex-start



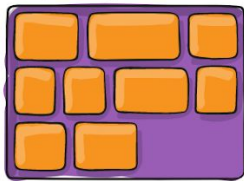
flex-end



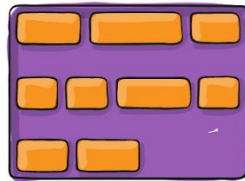
center



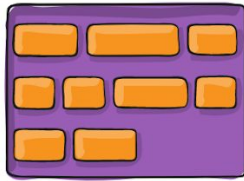
stretch



space-between



space-around



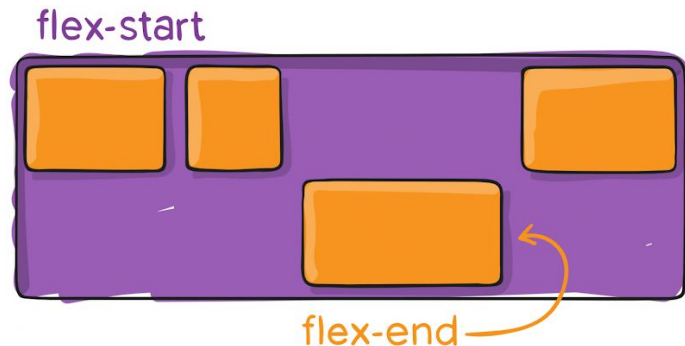
Свойства для первых дочерних элементов (flex элементы)



align-self

По умолчанию flex элементы располагаются в исходном порядке.

Однако свойство `order` управляет порядком их появления в контейнере flex.



Управление пропорциями и порядками блоков



order

По умолчанию flex элементы располагаются в исходном порядке.

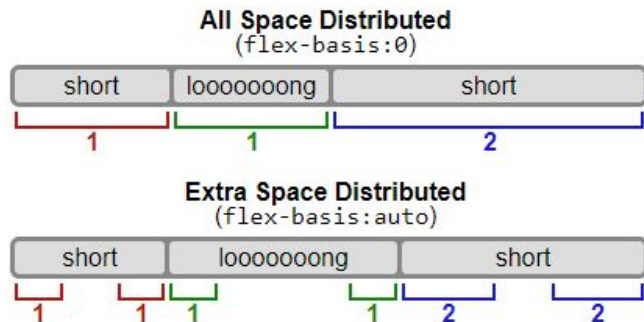
Однако свойство order управляет порядком их появления в контейнере flex.



Flex-basis



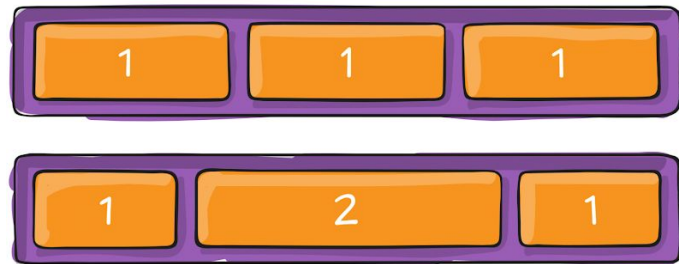
Это свойство определяет размер элемента по умолчанию перед распределением оставшегося пространства. Это может быть **длина** (например, 20%, 5rem и т.д.) Или **ключевое слово**



Flex-grow



Это свойство определяет способность **flex элемента** растягиваться в случае необходимости. Оно принимает значение от нуля, которое служит пропорцией.



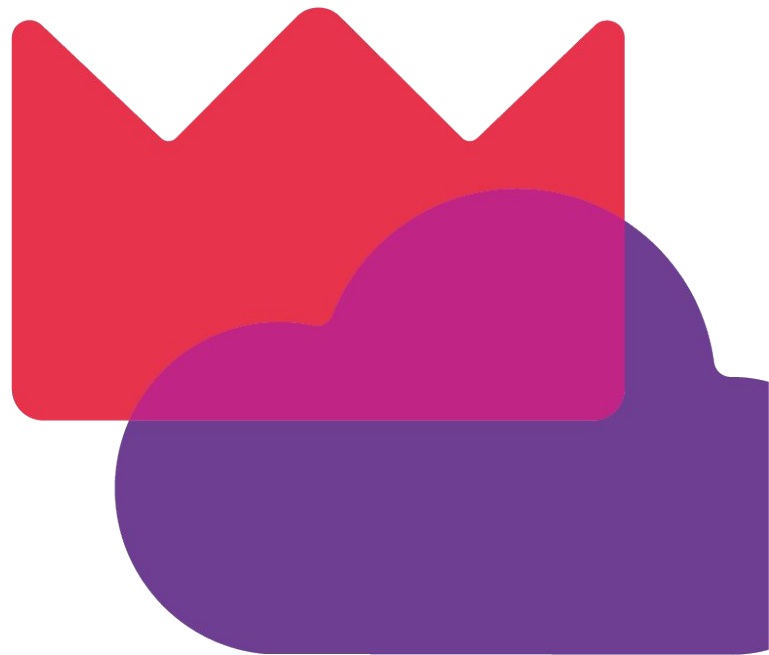


Flex-shrink

Это свойство определяет способность гибкого элемента сжиматься при необходимости

Домашнее задание:

- 1 Создание макета своей страницы-портфолио
- 2 Использовать все настройки расположения элементов в контейнере
- 3 Задать параметры и очередность блоков



Теперь вы знаете:

- ✓ Что такое флексбокс
- ✓ Свойства флекс-элементов
- ✓ Как задавать пропорции и порядок элементов

