

## Свойство display, тип бокса 2.0

Прошлая практическая работа объяснила принцип формирования потока документа, а также один из вероятных вариантов отображения свойства display - grid (сетка по технологии [CSS Grid Layout](#)). Грид представляет собой пересекающийся набор горизонтальных и вертикальных линий, образующих колонки и строки. Элементы могут быть помещены в грид в пределах линий этих колонок и строк.

Теперь нам необходимо рассмотреть новое значение свойства display - flex.

Флексбокс — это CSS-механизм, предназначенный для построения сеток и создания сложных раскладок блоков.

Другие механизмы, с помощью которых мы раньше строили сетки, задумывались совсем не для этого: плавающие блоки нужны для создания блоков, которые текст должен обтекать, а таблицы используются для разметки табличных данных.

Флексбокс задумывался для создания «гибких» раскладок и хранит много тонкостей и чудес, о которых мы поговорим в этой работе. А пока начнём с простого. Как включить флексбокс?

```
* {                               //выбор элемента
  display: flex;                  //меняем стандартный метод отображения на "гибкий"
}
```

Элемент с display: flex; превращается во «флекс-контейнер» и внутри него начинает происходить вся магия гибкой раскладки.

Непосредственные потомки этого элемента превращаются во «флекс-элементы» и начинают играть по новым правилам.

## Задание

1. Перейдите по ссылке <https://flexboxfroggy.com/#ru>
2. Пройди игру до конца (24 шага)

# Создание адаптивного горизонтального меню

Несмотря на всю простоту и банальность игры - она затрагивает почти все основные свойства технологии. Теперь необходимо закрепить это дело на практике!

Флексбокс полезен при создании блоков с дочерними элементами динамической длины. Хороший пример — меню.

Часто встречаются дизайны, в которых пункты равномерно распределены по блоку меню. Первый пункт примыкает к левой части блока меню, а последний — к правой, причём с небольшими внутренними отступами.

Эту задачу можно попытаться решить, задав фиксированные отступы и ширину пунктам меню. Но такой способ не подойдёт, если количество пунктов меню или подписи внутри них будут изменяться.

И тут на помощь приходит флексбокс. Зададим меню раскладку флексбокса, пункты станут флекс-элементами. С помощью свойства распределения элементов `justify-content` добьемся нужного результата. Мы создавали навигационное меню в одной из первых работ, давайте его изменим

## Задание

1. Откройте файл `index.html` и файл стилей, который к нему относится в текстовом редакторе
2. В файле стилей блоку меню (тег `nav`) задайте свойству `display` значение `flex`
3. Свойству `align-items` задайте значение `center`
4. Свойству `justify-content` задайте значение `space-around`

# Переполнение блока

Для начала разберем что мы написали:

1. Первым действием мы изменили тип нашего элемента с блока на флекс-элемент. Теперь навигационное меню становится родителем для всех вложенных элементов.
2. Свойство `align-items` отвечает за выравнивание элементов относительно оси направления флекс-контейнера. По умолчанию имеет значение `start`, что в нашем случае означает выравнивание по верхней кромке контейнера. Изменив его на `center`, элементы выстроились относительно центра самого высокого из них( кнопка-картинка в данном случае)
3. `justify-content: space-around`; - великолепное свойство позволяющее автоматически распределить элементы по горизонтали (в нашем случае).  
Примеры других значений:

`justify-content: space-between`;

*/\* Равномерно распределяет все элементы по ширине флекс-блока. Первый элемент в начале, последний в конце \*/*

`justify-content: space-around`;

*/\* Равномерно распределяет все элементы по ширине флекс-блока. Все элементы имеют полноразмерное пространство с обоих концов \*/*

`justify-content: space-evenly`;

*/\* Равномерно распределяет все элементы по ширине флекс-блока. Все элементы имеют равное пространство вокруг \*/*

`justify-content: stretch`;

*/\* Равномерно распределяет все элементы по ширине флекс-блока. Все элементы имеют "авто-размер", чтобы соответствовать контейнеру \*/*

Пока все выглядит великолепно. Обновив страницу вы можете видеть, что элементы выровнялись относительно центральной оси блока по вертикали, а также расположились внутри родительского блока а комфортном для них расстоянии. А мы в свою очередь написали лишь 3 строчки кода! Технология на грани фантастики

Но, сайт растёт, количество дополнительных страниц увеличивается очень быстро, давайте проверим как на это отреагирует наше меню

## Задание

1. Скопируйте и тут же вставьте любой из пунктов меню ~30 раз
2. Посмотрите на свою верстку

## Свойство обертки

Печальное зрелище, не правда ли? При увеличении количества элементов родительский блок не придумал ничего лучше, кроме как продолжать увеличиваться горизонтально, в страхе потревожить соседние элементы. Техническая “стеснительность” сыграла с дизайном нашего меню злую шутку.

Для решения данного рода стандартных ситуаций существует еще одно свойство, отвечающее за правила вывода элементов внутри родительского блока - flex-wrap

## Задание

1. Задайте родителю свойство flex-wrap со значением wrap;
2. Изменим ширину родительского элемента с помощью width равным 50%;
3. Добавим также равные отступы с обеих сторон, используя margin: 0 auto;

# Не забывайте о детях

Посмотрите на свое меню! Оно уменьшилось как блок, но элементы в нем ведут себя по строгим правилам. В тесноте, да не в обиде! Теперь сколько угодно изменяя блок элементы будут располагаться с учетом установленных правил. Это конечно не значит что при уменьшении родителя до микронов дети также расщепятся - контентное переполнение победить не удастся.

Наконец, мы добились равноправия в конкретно взятой ячейке. Но как писал Джордж Оруэлл в повести "Скотный двор" - «Все животные равны. Но некоторые животные равнее других». Давайте зададим особые правила для одного из элементов нашего флекс-контейнера. Остановим выбор не самом отличном от других - кнопке-картинке. Для задания особых правил воспользуемся свойством flex - это сокращённое свойство, определяющее способность гибкого элемента растягиваться (flex-grow) или сжиматься (flex-shrink) для заполнения собой доступного свободного пространства. flex-basis - базовый размер относительно родителя. По умолчанию выглядит так:

```
flex: 0 1 auto;
```

Вместо такой записи можно использовать полную:

```
flex-grow: 0  
flex-shrink: 1  
flex-basis: auto
```

Но краткость - сестра таланта

## Задание

1. Выберите своей целью последний элемент. Можете добавить кнопке-картинке класс или айди, а можно использовать относительное таргетирование `nav > *:last-child`, используем именно его;
2. Позволим элементу растягиваться и сжиматься, а базовую ширину установим на уровне 50%;
3. Проверьте результат в браузере;
4. Добавьте в стили элемента `color: red`;
5. Измените таргет с `last-child` на `first-child`. Проверьте браузер
6. Измените таргет с `first-child` на `nth-child(6)`. Проверьте браузер

## Вертикальное меню.

Еще одно свойство, ускоряющее формирование наших элементов - `flex-direction`, что означает главное направление нашего флекс элемента. Значение по умолчанию `row` - в линию, что мы и видим при отрисовке нашего меню. Учтите, что при этом оси форматирования нашего меню сместятся - `align-items` станет выравнивать элементы относительно горизонтальной оси, а `justify-content` относительно вертикальной, поэтому учитывайте это при задании значений и размеров элементов

### Задание

1. Измените `flex-direction` значение на `column`.

# Конспект «Флексбокс»

## Флексбокс

Флексбокс — это CSS-механизм, который позволяет контролировать размер, порядок и выравнивание элементов по нескольким осям, распределение свободного места между элементами и многое другое.

Чтобы включить флексбокс, нужно задать элементу свойство `display: flex;`.  
После этого:

Элемент с `display: flex;` превращается во «флекс-контейнер».  
Непосредственные потомки этого элемента превращаются во «флекс-элементы» и начинают распределяться по новым правилам.  
Главная и поперечная оси  
Оси — одно из основных понятий во флексбоксах.

В обычном потоке документа блоки и текст располагаются слева направо и сверху вниз.

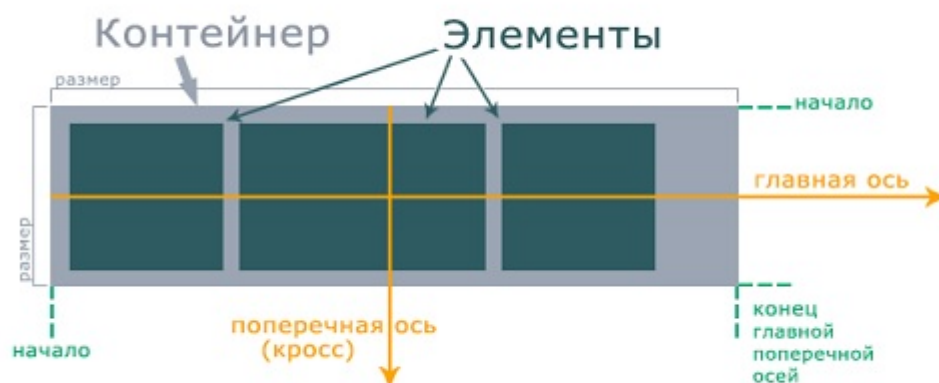
В привычной блочной модели направления «лево», «право», «верх» и «низ» неизменны. Но внутри флекс-контейнера эти понятия могут изменяться, потому что там можно изменять обычное направление потока.

Главной осью флекс-контейнера является направление, в соответствии с которым располагаются все его дочерние элементы. Поток флекс-элементов «течёт» вдоль главной оси от её начала к её концу.

Поперечной осью называется направление, перпендикулярное главной оси.

Вдоль этой оси работают «вертикальные» выравнивания.

Поперечная ось во флексбоксе



По умолчанию главная ось направлена слева направо, но её можно разворачивать во всех направлениях с помощью свойства `flex-direction`, которое задаётся для флекс-контейнера. Значения свойства:

Значение по умолчанию `row` — главная ось направлена слева направо.  
`column` — главная ось направлена сверху вниз.  
`row-reverse` — главная ось направлена справа налево.  
`column-reverse` — главная ось направлена снизу вверх.  
Поперечная ось всегда перпендикулярна главной оси и поворачивается вместе с ней:

Если главная ось направлена горизонтально, то поперечная ось смотрит вниз.  
Если главная ось направлена вертикально, то поперечная ось смотрит направо.  
Таким образом, поперечная ось никогда не смотрит вверх или влево, и свойства для поворота поперечной оси нет.

Распределение флекс-элементов  
Выравнивание по главной оси

CSS-свойство `justify-content` определяет то, как будут выровнены элементы вдоль главной оси. Доступные значения `justify-content`:

Значение по умолчанию `flex-start` — элементы располагаются у начала главной оси.

`flex-end` — элементы располагаются в конце главной оси.

`center` — элементы располагаются по центру главной оси.

`space-between` — элементы располагаются так, что расстояния между соседними одинаковые, а между элементами и краями флекс-контейнера отступов нет.

`space-around` — элементы располагаются так, что расстояния между соседними одинаковые, а между элементами и краями флекс-контейнера есть отступ, равный половине расстояния между соседними элементами.

`space-evenly` — расстояния между соседними элементами и краями флекс-контейнера одинаковые.

Выравнивание по поперечной оси

CSS-свойство `align-items` определяет то, как будут выровнены элементы вдоль поперечной оси. Доступные значения `align-items`:

Значение по умолчанию `stretch` — элементы растягиваются на всю «высоту» флекс-контейнера.

`flex-start` — элементы располагаются у начала поперечной оси.

`flex-end` — элементы располагаются в конце поперечной оси.

`center` — элементы располагаются по центру поперечной оси.

`baseline` — элементы выравниваются по базовой линии текста внутри них.

Распределение элементов по главной оси задаётся для всего флекс-контейнера и на все флекс-элементы действует одинаково, задать какому-то элементу отличное от других распределение по главной оси нельзя.

Поперечное выравнивание можно задать каждому элементу отдельно. Для этого используется свойство `align-self`, которое задаётся для самих



флекс-элементов, а не для флекс-контейнера. У свойства `align-self` те же самые значения, что и у `align-items`.

## Перенос флекс-элементов

Если флекс-элементов в контейнере станет больше, чем может уместиться в один ряд, то:

Они будут сжиматься до минимально возможной ширины.

Даже если им задать ширину, механизм флексбокса может её уменьшить.

Если они перестанут помещаться в контейнер и после уменьшения, то выйдут за его пределы, но продолжат располагаться в один ряд.

Чтобы этого не происходило, нужно воспользоваться свойством флекс-контейнера `flex-wrap`. У него есть два значения:

Значение по умолчанию `nowrap` — перенос флекс-элементов на новую строку запрещён.

`wrap` — разрешает перенос флекс-элементов на новую строку. Ряды элементов располагаются вдоль поперечной оси, первый ряд — в начале поперечной оси, последний — в конце.

`wrap-reverse` — также разрешает перенос флекс-элементов на новую строку. Ряды элементов располагаются в обратном порядке: первый — в конце поперечной оси, последний — в начале.

Выравнивание строк флекс-контейнера

Свойство `align-content` управляет выравниванием рядов флекс-элементов вдоль поперечной оси. У него и свойства `justify-content` очень похожие значения:

Значение по умолчанию `stretch` — растягивает ряды флекс-элементов, при этом оставшееся свободное место между ними делится поровну. Отображение строк при этом зависит от значения `align-items`:

Если у `align-items` задано значение `stretch`, то элементы в строках растягиваются на всю высоту своей строки.

Если значение отлично от `stretch`, то элементы в строках ужимаются под своё содержимое и выравниваются в строках в зависимости от значения `align-items`.

`flex-start` — располагает ряды флекс-элементов в начале поперечной оси.

`flex-end` — располагает ряды флекс-элементов в конце поперечной оси.

`center` — располагает ряды флекс-элементов в середине поперечной оси так, что отступов между соседними рядами нет, а расстояния между первым рядом и краем флекс-контейнера равно расстоянию между последним рядом и другим краем.

`space-between` — равномерно распределяет ряды флекс-элементов вдоль поперечной оси, расстояния между соседними рядами одинаковые, отступов у краёв нет.

`space-around` — равномерно распределяет ряды флекс-элементов вдоль поперечной оси, расстояния между соседними рядами одинаковые, отступы у краёв равны половине расстояния между соседними рядами.

`space-evenly` равномерно распределяет ряды вдоль поперечной оси, расстояния между соседними рядами и у краёв одинаковые.

Свойство `align-content` «перекрывает» заданное значение `align-items`, которое управляет выравниванием флекс-элементов вдоль поперечной оси. Это происходит и в случае, когда есть только один ряд флекс-элементов, и когда рядов несколько.

Ранее в спецификации было описано другое поведение:

Если есть только один ряд флекс-элементов, то работает `align-items`.

Если есть несколько рядов, то работает `align-content`.

В начале 2019 года поведение было актуализировано согласно спецификации во всех современных браузерах, теперь его можно встретить только в старых браузерах.

Порядковый номер флекс-элемента

Порядок следования флекс-элементов в потоке можно изменять с помощью свойства `order`, порядкового номера флекс-элемента, не меняя при этом HTML-код.

По умолчанию порядковый номер флекс-элементов равен 0, а сортировка элементов производится по возрастанию номера. Порядковый номер задаётся целым числом, положительным или отрицательным.

## Применение флексбоксов

Идеальное выравнивание

С помощью флексбокса можно отцентровать элемент по вертикали и горизонтали так, чтобы центровка сохранялась при изменении размеров элемента или контейнера.

Для этого нужно задать контейнеру раскладку флексбокса, а дочернему флекс-элементу `margin: auto`. В этом случае флекс-элемент уменьшит свой размер под содержимое и отцентрируется по вертикали и горизонтали.

## «Гибкое» меню

Флексброкс будет полезен, если нужно создать раскладку, в которой пункты равномерно распределены по блоку меню, при чём первый пункт примыкает к левой части блока меню, а последний — к правой, с небольшими внутренними отступами.

Чтобы это сделать, нужно задать меню раскладку флексбокса, тогда пункты станут флекс-элементами. Затем с помощью свойства распределения элементов `justify-content: space-around`; можно добиться нужного результата.

Если вы добавите в меню ещё один пункт, отступы между пунктами меню будут «гибко» меняться, подстраиваясь под новые условия.

## Сортировка элементов на CSS

Используя одновременно флексбокс и селектор по выделению чекбокса `:checked ~`, можно с помощью этого селектора управлять порядком флекс-элементов, изменяя направление главной оси с помощью `flex-direction`.

Лучше всего эффект работает, когда направление главной оси меняется с «сверху вниз» на «снизу вверх». При этом флекс-контейнер должен находиться в разметке на одном уровне с чекбоксом.

## Блоки одинаковой высоты

В обычной блочной модели есть фундаментальный недостаток — соседние блоки ничего не знают друг о друге, поэтому их высоты нельзя «связать». При этом надо учитывать, что содержимое блоков может быть разным и их высота может меняться.

На флексбоксах можно реализовать раскладку с блоками одинаковой высоты — флекс-элементы по умолчанию растягиваются на всю высоту контейнера. Для этого достаточно задать родительскому блоку `display: flex`;