Занятие 4

Блочная модель и макет страницы

Блочные и строчные элементы

Блочные и строчные элементы

Блочные элементы:

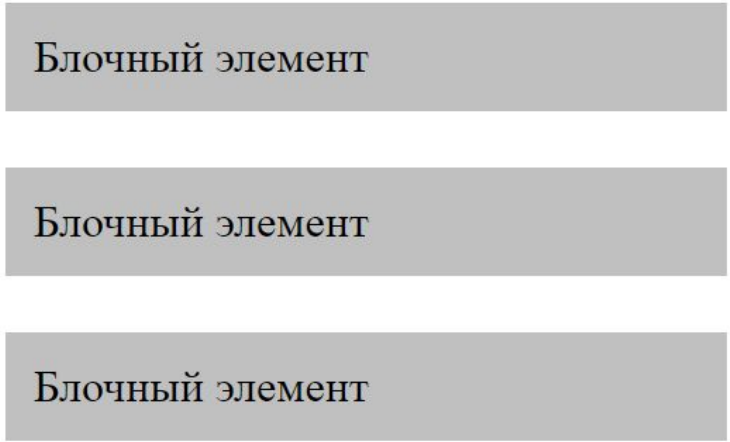
- Имеют отступ в начале и в конце, то есть располагаются на новой строке;
- Растягиваются по умолчанию на всю ширину страницы;
- Можно менять ширину и высоту (width и height свойства);
- Можно применять margin во всех сторон;

Строчные элементы:

- Располагаются в один ряд;
- Принимают размер контента внутри них;
- Нельзя менять ширину и высоту;
- Можно применять margin только слева и справа;

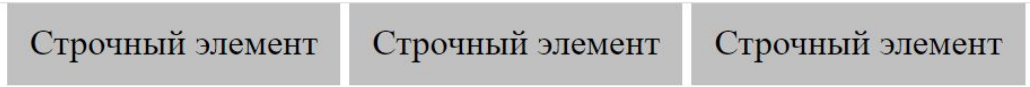
Блочные и строчные элементы

```
    <style>
        .block {
            margin: 20px;
            background: silver;
            padding: 10px
        }
     </style>
</head>
<body>
    <div class="block">Блочный элемент</div>
    <div class="block">Блочный элемент</div>
    <div class="block">Блочный элемент</div>
</body>
```



Блочные и строчные элементы

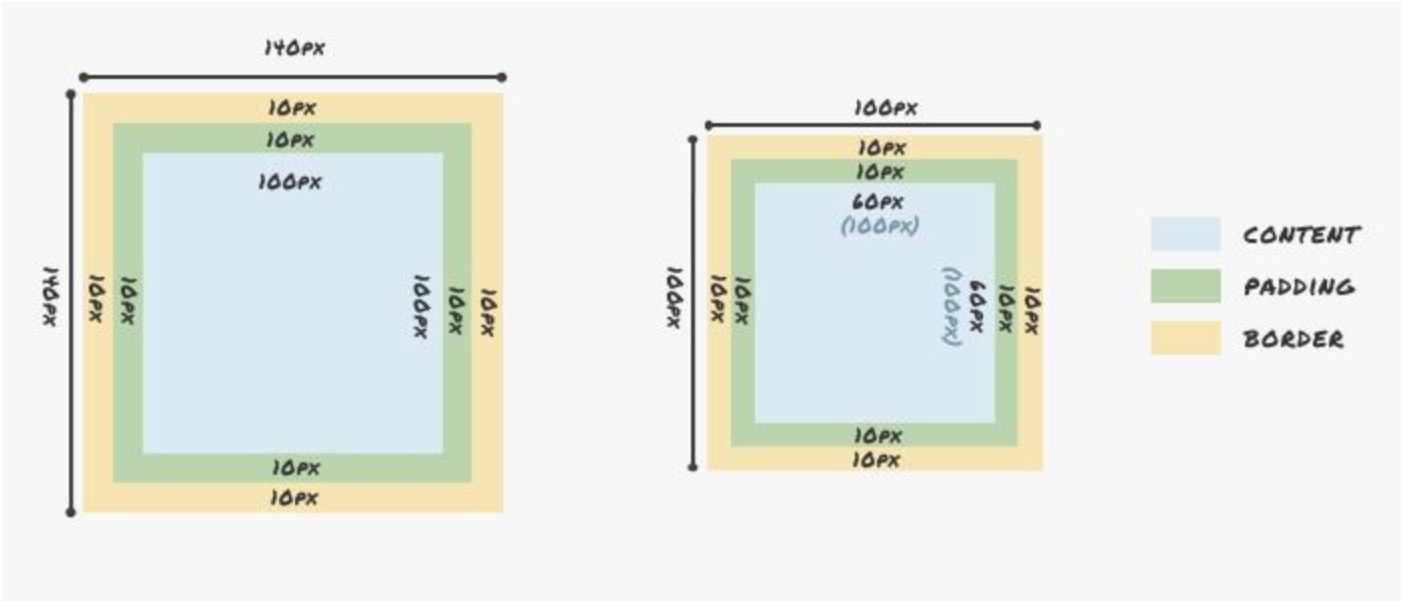
```
    <style>
        .block {
            margin: 0px;
            background: silver;
            padding: 10px
        }
     </style>
</head>
<body>
    <span class="block">Строчный элемент</span>
    <span class="block">Строчный элемент</span>
    <span class="block">Строчный элемент</span>
</body>
```



Блочные и строчные элементы

- Размер блока по умолчанию считается как совокупность размера контента, внутренних отступов и границы; (content-box)
- Когда мы используем width и height св-ва мы задаем размер контента, а не блока;
- Изменить формулу расчета можно с помощью св-ва box-sizing;
- Значения content-box и border-box;
- При значении border-box св-ва width и height задают не размер контента, а уже размер блока;

Блочные и строчные элементы



content-box

border-box

Блочные и строчные элементы

- overflow - св-ва для изменения поведения контента, которые не помещается в элемент;
- overflow-x - для изменения только по горизонтали;
- overflow-y - для изменения только по вертикали;

Расположение элементов

Расположение элементов

- display - св-во, задающее поведение для элемента: блочное, строчное или блочно-строчное;
- display: block; - блочное;
- display: inline; - строчное;
- display: inline-block; - блочно-строчное;

flexbox

Построения гибких макетов

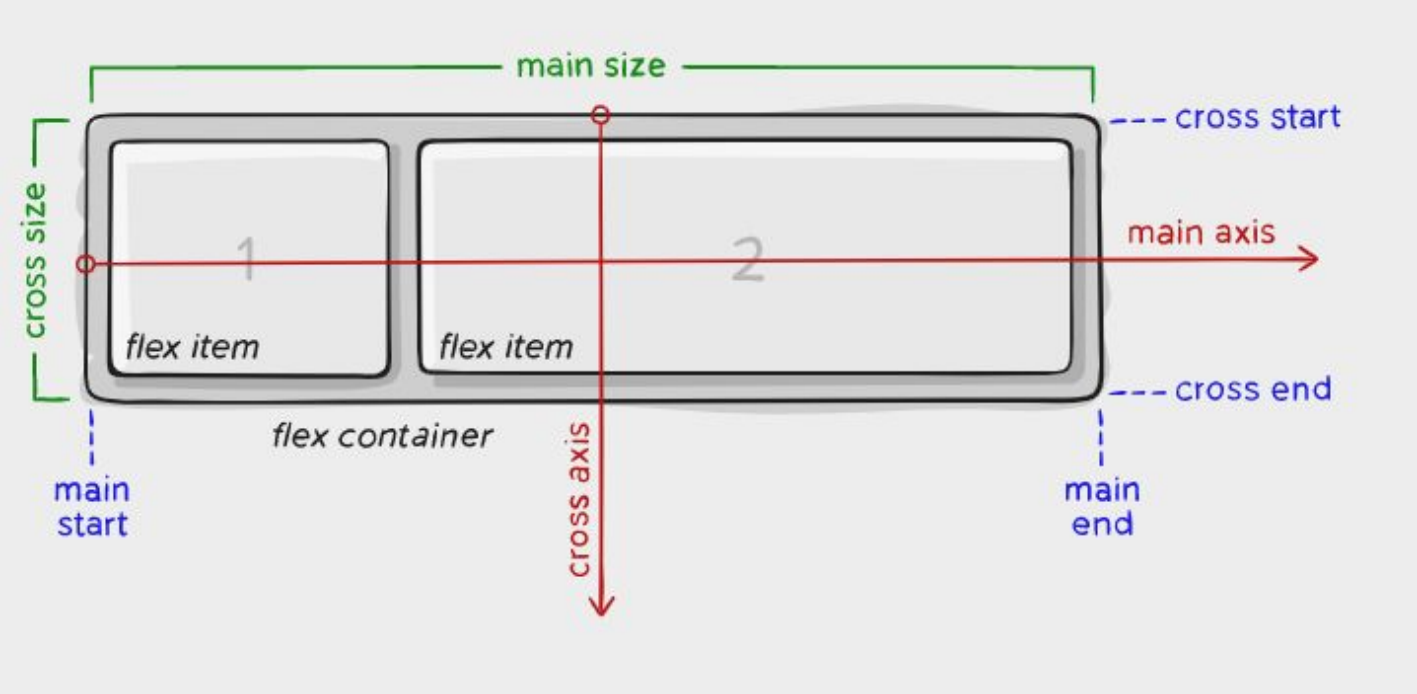
Что такое flexbox?

- Это мод для построения макетов, позволяющий упорядочивать элементы внутри контейнера
- Тянущиеся элементы
- Легко менять порядок и позиционирование элементов в контейнере

Контейнер и элементы

- Flex-контейнер – элемент, у которого задан display: flex
- Flex-элементы – непосредственные дочерние элементы такого контейнера

Основные концепты



- Main axis (основная ось) – ось, вдоль которой выстраиваются флекс-элементы
- Main start|main end (начало|конец основной оси) – флекс-элементы выстраиваются начиная от main start и заканчивая main end
- Cross axis – ось, перпендикулярная к основной оси. Направление зависит от направления главной оси
- Cross start|cross end – элементы размещаются вдоль поперечной оси начиная от cross start и заканчивая cross end

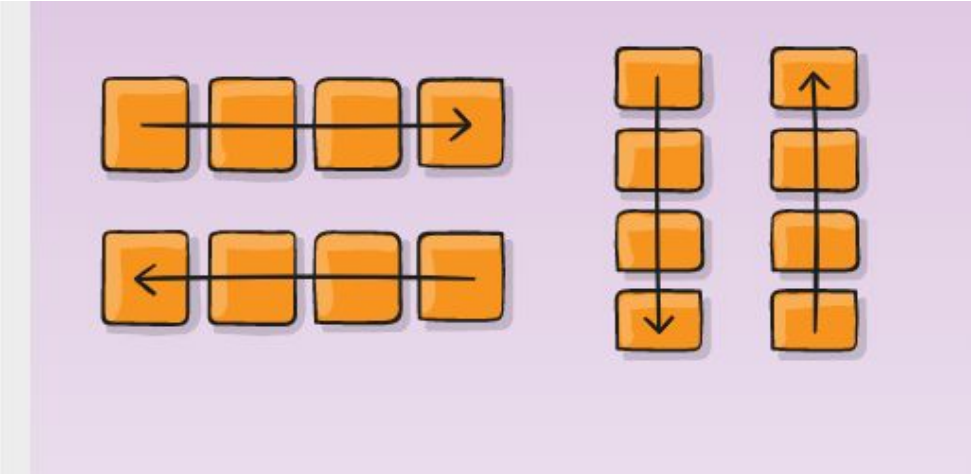
- Main size – ширина или высота флекс-элемента (зависит от направления главной оси)
- Cross size – ширина или высота флекс-элемента в направлении поперечной оси

Свойства

- https://css-tricks.com/snippets/css/a-guide-to-flexbox/

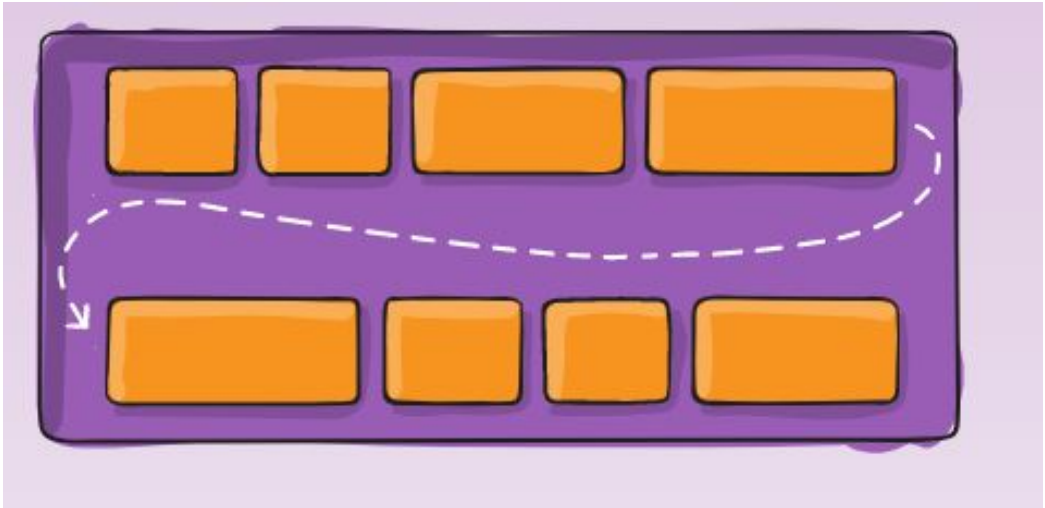
flex-direction: row | row-reverse | column | column-reverse

- Определяет направление главной оси



flex-wrap: nowrap | wrap | wrap-reverse

- По умолчанию флекс-элементы выстраиваются в одну линию. Для переноса flex-wrap: wrap

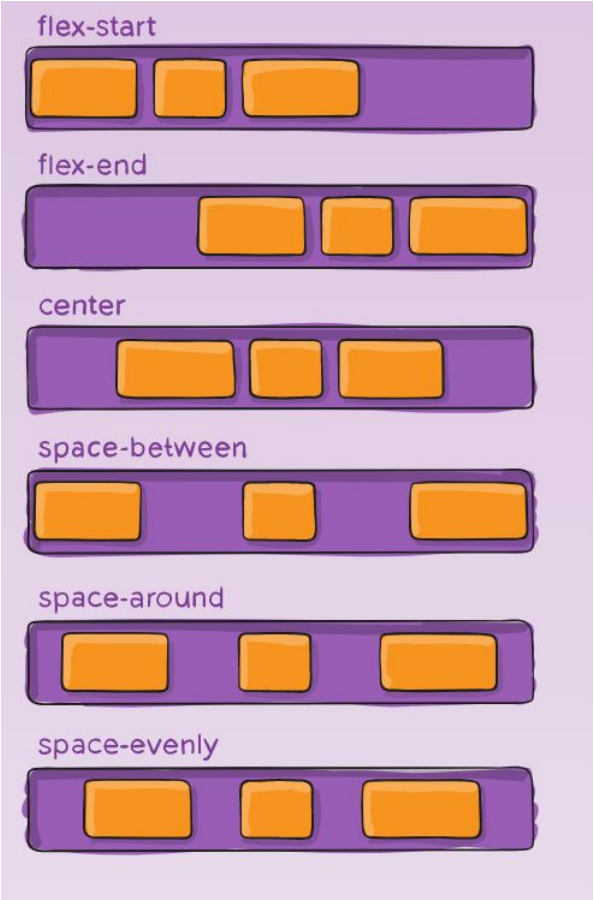


Flex-flow = flex-direction + flex-wrap

- flex-flow: column wrap;

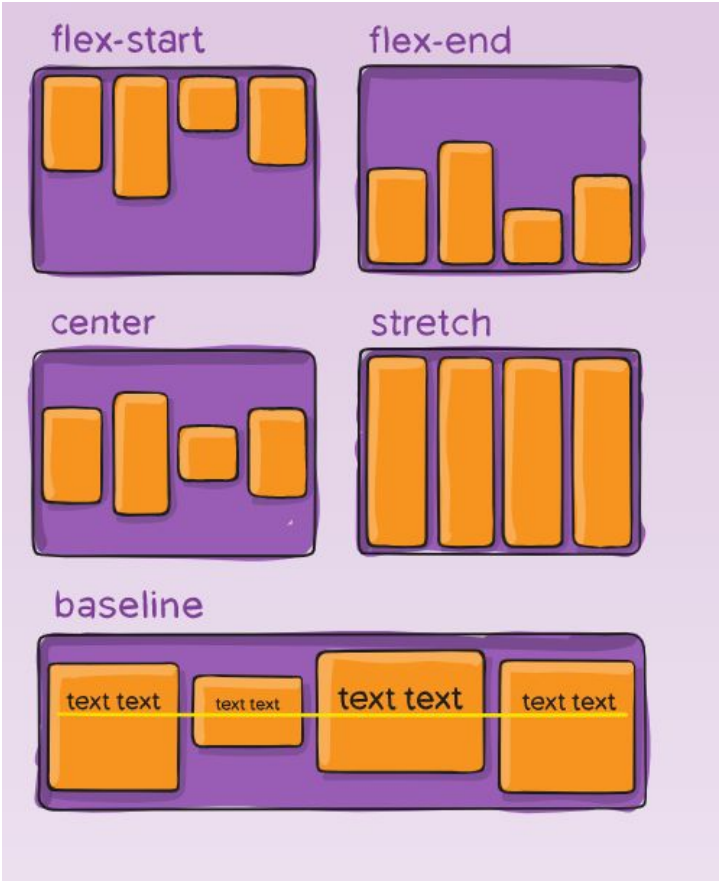
Justify-content

- Управляет распределением элементов по главной оси



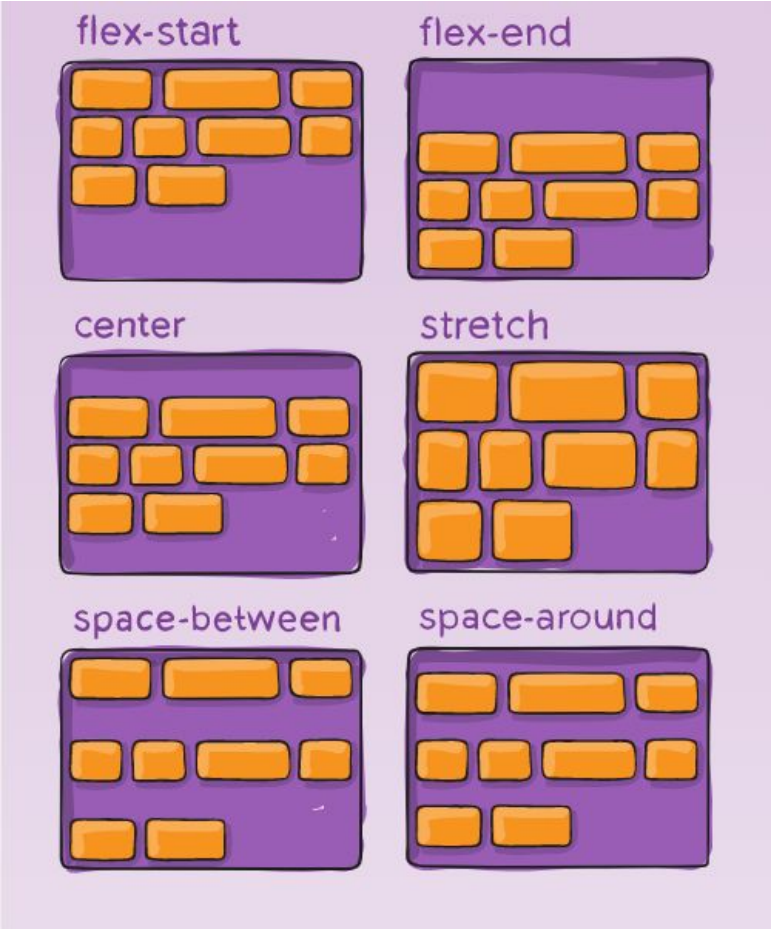
Align-items

- Управляет выравниванием элементов вдоль поперечной оси



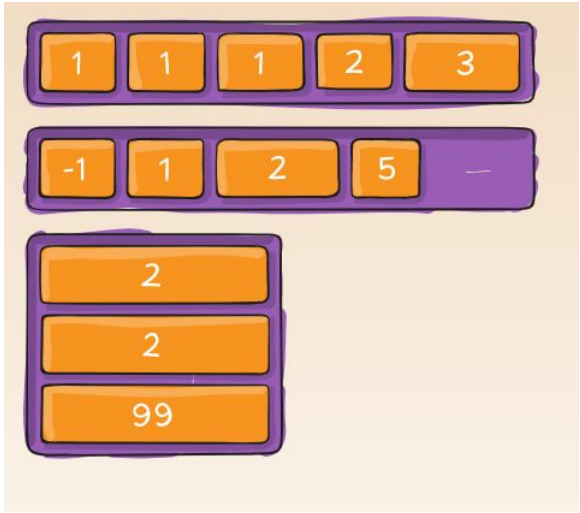
Align-content

- Управляет выравниванием линий флекс контейнера когда есть дополнительное место вдоль поперечной оси.
- Работает только вместе с flex-wrap: wrap | wrap-reverse



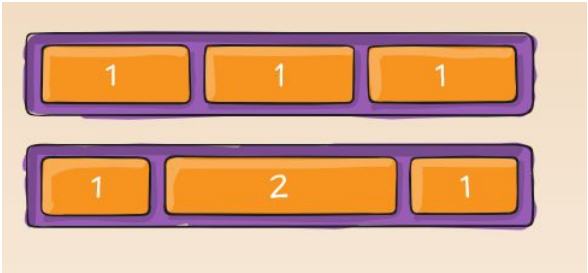
order

- По умолчанию флекс-элементы выстраиваются в соответствии с порядком в HTML документе.
- Свойство order позволяет управлять их порядком, не меняя HTML



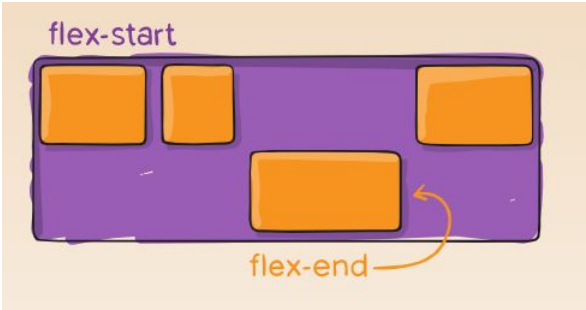
Flex-grow + flex-shrink

- Принимает безразмерное значение
- Управляет тем, какое количество места занимает флекс-элемент при наличии свободного пространства
- Если значение свойства у всех элементов одинаковое, они распределяются одинаково
- Flex-grow растягивает элемент
- Flex-shrink сжимает элемент



Align-self

- Позволяет переписать значение для индивидуальных элементов, заданных в align-items
- Принимает такие же значения как align-items



Flex-basis

- https://www.w3.org/TR/css-flexbox-1/images/rel-vs-abs-flex.svg
- https://css-tricks.com/almanac/properties/f/flex-basis/
- Определяет размер флекс-элементов до распределения места
- Flex-basis: auto – будет использовано заданное значение ширины
- Flex-basis: 0 – значение по умолчанию

Flex = flex-grow + flex-shrink + flex-basis

- flex: none | [<'flex-grow'> <'flex-shrink'>? || <'flex-basis'>]

Базовые примеры

- Создание карточек
- Выравнивание элемента по вертикали и горизонтали – создание базового лендинга https://tinder.com/
- Создание липкого подвала - https://css-tricks.com/couple-takes-sticky-footer/

Проект

- Разобрать сайт Postman https://www.postman.com/
- Сделать первую секцию без формы, верхнее меню с базовой навигацией (без кнопок логина и входа), вторую и третью секции и базовый подвал

Ресурсы

- https://css-tricks.com/snippets/css/a-guide-to-flexbox/ - гайд по флексбоксу
- https://flexboxfroggy.com/#ru – flexbox froggy