**Создание гибкого макета страницы с помощью Flexbox**

**9.1 Понятие Flexbox. Конейнер Flex**

Flexbox - это общее название модуля Flexible Box Layout. Этот модуль определяет специальный режим верстки пользовательского интерфейса, который называется Flex Layout, отличающийся от макета таблицы или блока.

Благодаря Flexbox стало проще создавать сложные интерфейсы, в которых можно переопределять направление и выравнивание элементов, а также создавать адаптивные табличные представления. Все современные браузеры, включая Microsoft Edge, Opera, Яндекс, Safari, Firefox имеют полную поддержку этого модуля.

Основными компонентами макета Flexbox являются контейнер Flex (flex container) и элементы Flex (flex items).

**9.1.1 Основные понятия**

Одним из ключевых понятий является main axis или центральная ось. Это условная ось в гибком контейнере, вдоль которой располагаются гибкие элементы (рисунок 9.1).

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 9.1 - Main axis и cross axiss в flexbox и CSS 3

Элементы в контейнере можно располагать горизонтально в ряд и вертикально в столбец. В зависимости от типа локации будет меняться и центральная ось. Термины main start и main end описывают начало и конец центральной оси соответственно, а расстояние между ними называется основным размером (main size).

Помимо главной оси, существует еще cross axis или поперечная ось. Она перпендикулярна основной. При расположении элементов в ряд поперечная ось направлена ​​сверху вниз, а при расположении в столбик - слева направо. Начало поперечной оси обозначалось как поперечное начало, а конец - как поперечный конец. Расстояние между ними называется cross size.

То есть, если элементы расположены в линию, основной размер представляет собой ширину контейнера или элементов, а поперечный размер представляет их высоту. С другой стороны, если элементы расположены в столбце, основной размер представляет собой высота контейнера и элементов, а поперечный размер представляет их ширину.

**9.1.2 Создание flex-контейнера**

Для создания flex-контейнера необходимо присвоить его стилевому свойству display одно из двух значений: flex или inline-flex.

Далее представлен пример создания простейшей веб-страницы, которая применяет flexbox. Для контейнера flex-container установлено свойство display:flex. В нем располагается три flex-элемента.

**<head> <style>**

**.flex-container { display: flex; }**

**.flex-item {text-align:center; font-size: 1.2em; padding: 1.6em;color: snow; }**

**.color1 {background-color: RosyBrown;}**

**.color2 {background-color: SandyBrown ;}**

**.color3 {background-color: SaddleBrown;} </style> </head>**

**<body><div class="flex-container"> <h3>Виды кофе</h3>**

**<div class="flex-item color1">Капучино. Капучино - кофейный**[**напиток**](https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D0%BF%D0%B8%D1%82%D0%BE%D0%BA)[**итальянской кухни**](https://ru.wikipedia.org/wiki/%D0%98%D1%82%D0%B0%D0%BB%D1%8C%D1%8F%D0%BD%D1%81%D0%BA%D0%B0%D1%8F_%D0%BA%D1%83%D1%85%D0%BD%D1%8F)**на основе**[**эспрессо**](https://ru.wikipedia.org/wiki/%D0%AD%D1%81%D0%BF%D1%80%D0%B5%D1%81%D1%81%D0%BE)**с добавлением в него подогретого до 65 градусов вспененного**[**молока**](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%BB%D0%BE%D0%BA%D0%BE). **</div>**

**<div class="flex-item color2">Латте. Латте -** [**кофейный напиток**](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D1%84%D0%B5%D0%B9%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%BA%D1%82%D0%B5%D0%B9%D0%BB%D1%8C)**на основе**[**молока**](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%BB%D0%BE%D0%BA%D0%BE)**, представляющий собой трёхслойную смесь из молочной пены,**[**молока**](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%BB%D0%BE%D0%BA%D0%BE) **и кофе**[**эспрессо**](https://ru.wikipedia.org/wiki/%D0%AD%D1%81%D0%BF%D1%80%D0%B5%D1%81%D1%81%D0%BE)**. </div> <div class="flex-item color3">Американо. Американо  - кофе по-американски, способ приготовления**[**кофе**](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D1%84%D0%B5)**, заключающийся в соединении определённого количества горячей**[**воды**](https://ru.wikipedia.org/wiki/%D0%92%D0%BE%D0%B4%D0%B0)**и**[**эспрессо**](https://ru.wikipedia.org/wiki/%D0%AD%D1%81%D0%BF%D1%80%D0%B5%D1%81%D1%81%D0%BE)**. </div> </div></body>**

Реализация данного кода представлена на рисунке 9.2.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Веб-сайт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 9.2 - Flex Container в CSS3

Если значение flex определяет контейнер как блочный элемент, то значение inline-flex определяет элемент как строчный (inline). Далее представлено оба способа на примере веб-страницы о чае.

**<head> <style>  
 .flex-container { display: flex; border:3px solid black; }  
 .inline-flex-container { display: inline-flex; border:3px solid black; margin-top:8px; }  
 .flex-item { text-align:center; font-size: 1.2em; padding: 1.6em; }  
 .color1 {background-color: BurlyWood}  
 .color2 {background-color: YellowGreen;}  
 .color3 {background-color: AntiqueWhite;} </style></head>  
<body> <div> <div class="flex-container">  
 <div class="flex-item color1">Чёрный чай</div>  
 <div class="flex-item color2">Зеленый чай</div>  
 <div class="flex-item color3">Белый чай</div> </div>  
 <div class="inline-flex-container">  
 <div class="flex-item color1">Чёрный чай</div>  
 <div class="flex-item color2"> Зеленый чай</div>  
 <div class="flex-item color3">Белый чай</div> </div> </div> </body>**Реализация данного кода представлена на рисунке 9.3.

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 9.3 - Inline-flex и flexbox в CSS3

В частности, в первом случае гибкий контейнер растягивается на всю ширину страницы, а во втором случае занимает ровно то пространство, которое необходимо для гибких элементов.

**9.2 Направление flex-direction**

Flex-элементы во flex-контейнере могут иметь определенное направление, а именно они могут располагаться в виде строк или в виде столбцов. Для управления направлением элементов CSS3 предоставляет свойство flex-direction. Оно определяет направление элементов и может принимать следующие значения:

* row: значение по умолчанию, при котором элементы располагаются в виде строки слева направо;
* row-reverse: элементы также располагаются в виде стоки только в обратном порядке справа налево;
* column: элементы располагаются в столбик сверху вниз;
* column-reverse: элементы располагаются в столбик в обратном порядке.

Далее приведен пример использования свойства flex-direction.

**<head> <style>**

**.flex-container { display: flex;border:2px solid black;}**

**.row {flex-direction: row;}**

**.row-reverse {flex-direction: row-reverse; }**

**.flex-item {text-align:center;font-size: 1.2em;padding: 1.6em; }**

**.color1 {background-color: Plum;}**

**.color2 {background-color: LightPink;}**

**.color3 {background-color: LightGreen;}**

**</style></head>**

**<body> <h3>Строка в прямом направлении </h3>**

**<div class="flex-container row">**

**<div class="flex-item color1">Ирис -**[**род**](https://ru.wikipedia.org/wiki/%D0%A0%D0%BE%D0%B4_(%D0%B1%D0%B8%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%8F))**многолетних**[**корневищных**](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D0%BD%D0%B5%D0%B2%D0%B8%D1%89%D0%B5)**растений семейства**[**Ирисовые, или Касатиковые**](https://ru.wikipedia.org/wiki/%D0%98%D1%80%D0%B8%D1%81%D0%BE%D0%B2%D1%8B%D0%B5)**. Род насчитывает около 800**[**видов**](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B2%D0%B8%D0%B4)**с богатейшим разнообразием форм и оттенков, за что и получил своё название (от**[**др. греч.**](https://ru.wikipedia.org/wiki/%D0%94%D1%80%D0%B5%D0%B2%D0%BD%D0%B5%D0%B3%D1%80%D0%B5%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA)**радуга).</div>**

**<div class="flex-item color2">Пион - род травянистых многолетников и листопадных кустарников. Единственный род семейства пионовые, пионы цветут в конце весны, ценятся садоводами за пышную листву, эффектные цветы и декоративные плоды. </div>**

**<div class="flex-item color3">Ландыш - монотипный либо олиготипный род однодольных растений семейства Спаржевые. Как правило, считается монотипным родом с единственным видом Ландыш майский. </div>**

**</div> <h3>Реверсивная строка</h3> <div class="flex-container row-reverse">**

**<div class="flex-item color1">Ирис -**[**род**](https://ru.wikipedia.org/wiki/%D0%A0%D0%BE%D0%B4_(%D0%B1%D0%B8%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%8F))**многолетних**[**корневищных**](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D0%BD%D0%B5%D0%B2%D0%B8%D1%89%D0%B5)**растений семейства**[**Ирисовые, или Касатиковые**](https://ru.wikipedia.org/wiki/%D0%98%D1%80%D0%B8%D1%81%D0%BE%D0%B2%D1%8B%D0%B5)**. Род насчитывает около 800**[**видов**](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B2%D0%B8%D0%B4)**с богатейшим разнообразием форм и оттенков, за что и получил своё название (от**[**др. греч.**](https://ru.wikipedia.org/wiki/%D0%94%D1%80%D0%B5%D0%B2%D0%BD%D0%B5%D0%B3%D1%80%D0%B5%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA)**радуга).</div>**

**<div class="flex-item color2">Пион - род травянистых многолетников и листопадных кустарников. Единственный род семейства пионовые, пионы цветут в конце весны, ценятся садоводами за пышную листву, эффектные цветы и декоративные плоды. </div>**

**<div class="flex-item color3">Ландыш - монотипный либо олиготипный род однодольных растений семейства Спаржевые. Как правило, считается монотипным родом с единственным видом Ландыш майский. </div> </div> </body>**

Реализация данного кода представлена на рисунке 9.3.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 9.4 - Flex-direction во flexbox и CSS3

Аналогично работает расположение в виде столбца.

**<head> <style> .flex-container { display: flex; border:2px solid black; }  
 . column { flex-direction: column;}  
 .column-reverse { flex-direction: column-reverse; }  
 .flex-item { text-align:center; font-size: 0.9em;padding: 0.9em; color: black;}  
 .color1 {background-color: Gold;}  
 .color2 {background-color: Tomato;}  
 .color3 {background-color: Chartreuse;} </style> </head> <body>**

**<h3>Столбец в прямом направлении</h3> <div class="flex-container column">  
 <div class="flex-item color1"> Банан - название съедобных плодов культивируемых растений рода Банан (Musa); обычно под таковыми понимают Musa acuminata. С ботанической точки зрения банан является ягодой, многосеменной и толстокожей. </div>  
 <div class="flex-item color2">Яблоко – сочный плод яблони, который употребляется в пищу в свежем и запеченном виде, служит сырьём в кулинарии и для приготовления напитков. Наибольшее распространение получила яблоня домашняя, реже выращивают яблоню сливолистную.</div>  
 <div class="flex-item color3"> Груша - род плодовых и декоративных деревьев и кустарников семейства розовые, а также их плод. Плоды груш отличаются характерной зернистой мякотью из-за наличия каменистых клеток. </div> </div>  
<h3>Реверсивный столбец</h3> <div class="flex-container column-reverse">  
 <div class="flex-item color1"> Банан - название съедобных плодов культивируемых растений рода Банан (Musa); обычно под таковыми понимают Musa acuminata. С ботанической точки зрения банан является ягодой, многосеменной и толстокожей. </div>  
 <div class="flex-item color2"> Яблоко – сочный плод яблони, который употребляется в пищу в свежем и запеченном виде, служит сырьём в кулинарии и для приготовления напитков. Наибольшее распространение получила яблоня домашняя, реже выращивают яблоню сливолистную. </div>  
 <div class="flex-item color3">Груша - род плодовых и декоративных деревьев и кустарников семейства розовые, а также их плод. Плоды груш отличаются характерной зернистой мякотью из-за наличия каменистых клеток. </div> </div> </body>**

Реализация данного кода представлена на рисунке 9.5.

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 9.5 – Использование column во Flexbox CSS3

**9.3 Flex-flow. Порядок элементов**

**9.3.1 Flex-flow**

Свойство flex-flow является сокращенной формой записи свойств flex-direction и flex-wrap и позволяет за одну декларацию задать направление главной оси и возможность переноса флекс-элементов на новую строку.

Свойство flex-wrap определяет, должны ли элементы переноситься на новую строку или оставаться на одной. Имеет три значения - nowrap, wrap, wrap-reverse. nowrap - значение по умолчанию, указывает, что элементы располагаются в контейнере в одну строку или столбец.

**<head> <style>**

**.flex-container {display: flex;**

**border: solid 0.24em black;  
 height:24.24em;**

**flex-flow: row wrap;}  
 .flex-item {text-align:center;**

**font-size: 0.9em;**

**padding: 1.4em; }  
 .col1 {background-color: PaleTurquoise;}  
 .col2 {background-color: LightSkyBlue;}  
 .col3 {background-color: LightBlue;}   
 .col4 {background-color: DeepSkyBlue;}  
 .col5 {background-color: CornflowerBlue;}  
 </style></head>  
<body><div class="flex-container">  
 <div class="flex-item col1">Яхта - судно, оснащенное парусами или оборудованное механическим двигателем, предназначенное для прогулок, туризма или спорта.</div>  
 <div class="flex-item col2">Теплоход - обобщающее понятие, описывающее класс самоходных судов, преобразующих тепловую энергию сжигания топлива в механическую.</div>  
 <div class="flex-item col3">Катер – открытая модель судна, с мотором стационарного типа или с подвесным движком, обустроенные каютами.</div>  
 <div class="flex-item col4"> Пароход – судно, движение которого обеспечивается паровой машиной или турбиной вращающей гребные колеса или винты.</div>  
 <div class="flex-item col5"> Корабль – в современном понимании, военное или большое парусное судно с боевыми орудиями. </div> </div> </body>**

Реализация данного кода представлена на рисунке 9.6.

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 9.6 – Применение свойства Flex-flow

**9.3.2 Свойство order**

Свойство order позволяет установить группу для flex-элемента, позволяя тем самым переопределить его позицию внутри flex-контейнера. В качестве значения свойство принимает числовой порядок группы. К одной группе может принадлежать несколько элементов.

Например, элементы в группе 0 располагаются перед элементами с группой 1, а элементы с группой 1 располагаются перед элементами с группой 2 и так далее.

**<head> <style>.flex-container { display: flex; flex-flow: row wrap;}  
 .flex-item {text-align:center; font-size: 1em; padding: 1.5em; }  
 .gr1{order:-1; } .gr2{ order:1;}  
 .col1 {background-color: Tan;}  
 .col2 {background-color: BurlyWood;} .col3 {background-color: Khaki;}  
 .col4 {background-color: PeachPuff;} .col5 {background-color: DarkKhaki ;}  
 </style></head> <body> <h3>Породы обезьян</h3>  
<div class="flex-container">  
 <div class="flex-item col1">Мартышка</div>  
 <div class="flex-item col2 gr2">Шимпанзе</div>  
 <div class="flex-item col3 gr2">Орангутан</div>  
 <div class="flex-item col4">Макака</div>  
 <div class="flex-item col5 gr1">Горилла</div> </div></body>**

Реализация данного кода представлена на рисунке 9.7

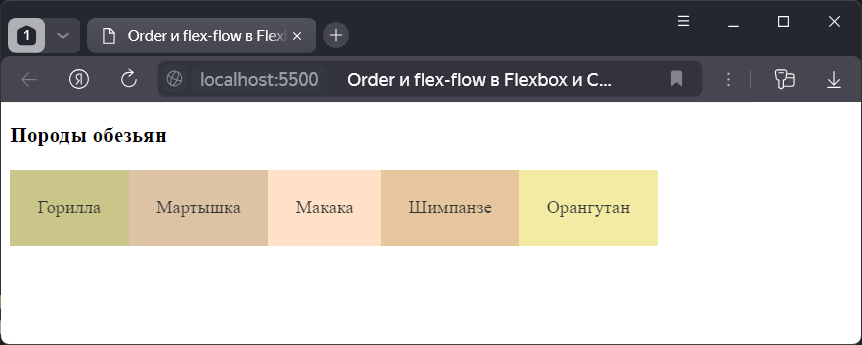


Рисунок 9.7 - Order и flex-flow в Flexbox и CSS3

В данном случае определены 3 группы. Первый отображается последний элемент, так как он имеет группу -1. По умолчанию если у элементов явным образом не указано свойство order, то оно имеет значение 0. И последними в данном случае отображаются второй и третий элемент, так как у них свойство order равно 1.

**9.4 Выравнивание элементов. Justify-content**

Иногда можно сталкиваться с ситуацией, когда пространство flex-контейнеров по размеру отличается от пространства, необходимого для flex-элементов. Например, когда flex-элементы не используют все пространство flex-контейнера; либо flex-элементам требуется большее пространство, чем доступно во flex-контейнере. В этом случае элементы выходят за пределы контейнера.

Для управления подобными ситуациями можно применять свойство justify-content. Оно выравнивает элементы вдоль основной оси - main axis и может принимает следующие значения:

* flex-start: значение по умолчанию, при котором первый элемент выравнивается по левому краю контейнера (при расположении в виде строки) или по верху (при расположении в виде столбца), за ним располагается второй элемент и так далее;
* flex-end: последний элемент выравнивается по правому краю (при расположении в виде строки) или по низу (при расположении в виде столбца) контейнера, за ним выравнивается предпоследний элемент и так далее;
* center: элементы выравниваются по центру;
* space-between: если в строке только один элемент или элементы выходят за границы flex-контейнера, то данное значение аналогично flex-start. В остальных случаях первый элемент выравнивается по левому краю (при расположении в виде строки) или по верху (при расположении в виде столбца), а последний элемент - по правому краю контейнера (при расположении в виде строки) или по низу (при расположении в виде столбца).
* space-around: если в строке только один элемент или элементы выходят за пределы контейнера, то его действие аналогично значению center. В ином случае элементы равным образом распределяют пространство между левым и правым краем контейнера, а расстояние между первым и последним элементом и границами контейнера составляет половину расстояния между элементами.

Далее приведен пример выравнивания элементов строки.

**<head> <style>**

**.flex-container { display: flex; border:2px black solid;}**

**.end{justify-content: flex-end; }**

**.center{ justify-content: center; }**

**.between{ justify-content: space-between;}**

**.around{ justify-content: space-around; }**

**.flex-item {text-align:center;font-size: 0.9em; padding: 1.6em; }**

.**col1 {background-color: LightCoral;} .col4 {background-color: MediumTurquoise;}**

**.col2 {background-color: Orange ;} .col5 {background-color: Lime;}**

**.col3 {background-color: Cyan;} .col6 {background-color: DeepSkyBlue;}**

**</style> </head>**

**<body> <h2> Материки планеты Земля </h2>**

**<h3>Использование flex-end</h3>**

**<div class="flex-container end">**

**<div class="flex-item col1">Азия</div>**

**<div class="flex-item col2">Африка</div>**

**<div class="flex-item col3">Европа </div>**

**<div class="flex-item col4">Австралия </div>**

**<div class="flex-item col5"> Америка </div>**

**<div class="flex-item col6">Антарктида</div>**

**</div><h3>Использование center</h3>**

**<div class="flex-container center">**

**<div class="flex-item col1">Азия </div>**

**<div class="flex-item col2">Африка</div>**

**<div class="flex-item col3">Европа</div>**

**<div class="flex-item col4">Австралия</div>**

**<div class="flex-item col5">Америка</div>**

**<div class="flex-item col6">Антарктида</div>**

**<h3>Использование space-between</h3>**

**<div class="flex-container between">**

**<div class="flex-item col1">Азия</div>**

**<div class="flex-item col2">Африка</div>**

**<div class="flex-item col3">Европа</div>**

**<div class="flex-item col4">Австралия</div>**

**<div class="flex-item col5">Америка</div>**

**<div class="flex-item col6">Антарктида</div>**

**<h3>Использование space-around</h3>**

**<div class="flex-container around">**

**<div class="flex-item col1">Азия</div>**

**<div class="flex-item col2">Африка</div>**

**<div class="flex-item col3">Европа</div>**

**<div class="flex-item col4">Австралия</div>**

**<div class="flex-item col4">Америка</div>**

**<div class="flex-item col4">Антарктида</div> </div> </body>**

Реализация программного кода представлена на рисунке 9.8.

Изображение выглядит как текст, снимок экрана, Красочность, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 9.8 - Justify-content в Flexbox и CSS3

Выравнивание в виде столбцов представлено на рисунке 9.9.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 9.9 - Justify-content и column в Flexbox и CSS3

**9.5 Выравнивание элементов. Свойство align-items**

Свойство align-items также выравнивает элементы, но по поперечной оси (вертикально, если они организованы в виде строки, и горизонтально, если они организованы в виде столбца). Это свойство может принимать следующие значения:

* stretch: значение по умолчанию, при котором гибкие элементы растягиваются по всей высоте или по всей ширине гибкого контейнера;
* flex-start: элементы выравниваются по верхнему краю или по левому краю гибкого контейнера;
* flex-end: элементы выравниваются по нижнему краю или по правому краю гибкого контейнера;
* center: элементы выравниваются по центру гибкого контейнера;
* baseline: элементы выравниваются по их базовой линии.

Далее приведен пример выравнивания при расположении в строку.

**<head> <style>**

**.container { display: flex;border:2px black solid;height:4.9em;}**

**.start{align-items: flex-start;} .end{align-items: flex-end;}**

**.center{align-items: center; } .baseline{ align-items: baseline; }**

**.item { text-align:center; font-size: 0.9em; padding: 1.1em; color: white; }**

**.largest{ padding-top:1.9em; }**

**.col1 {background-color: LimeGreen;} .col2 {background-color: MediumSeaGreen;}**

**.col3 {background-color: ForestGreen;}. col4 {background-color: DarkOliveGreen;} </style> </head>**

**<body> <h3>Использование flex-start</h3> <div class="container start">**

**<div class="item col1">Береза</div>**

**<div class="item col2">Тополь</div>**

**<div class="item col3">Дуб</div>**

**<div class="item col4">Осина</div>**

**</div> <h3>Использование flex-end</h3> <div class="container end">**

**<div class="item col1">Береза</div>**

**<div class="item col2">Тополь</div>**

**<div class="item col3">Дуб</div>**

**<div class="item col4">Осина</div>**

**</div> <h3>Использование center</h3> <div class="container center">**

**<div class="item col1">Береза</div>**

**<div class="item col2">Тополь</div>**

**<div class="item col3">Дуб</div>**

**<div class="item col4">Осина</div>**

**</div> <h3>Использование baseline</h3><div class="container baseline">**

**<div class="item col1">Береза</div>**

**<div class="item col2 largest">Тополь</div>**

**<div class="item col3">Дуб</div>**

**<div class="item col4">Осина</div>**

**</div></body>**

Реализация данного кода представлена на рисунке 9.10.

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 9.10 - Align-items во Flexbox и CSS3

Аналогично свойство работает при расположении в столбец. Например, изменим стили flex-контейнера следующим образом (рисунок 9.11):

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 9.11 - Align-items и column в Flexbox и CSS3

**9.6 Многоколоночный дизайн на Flexbox**

Основная идея мультиколонок заключается в том, что можете взять фрагмент содержимого и поместить его в несколько колонок, как в газете. Вы делаете это с помощью одного или двух свойств. Далее рассматривается, как можно сделать простейшие многоколонночные макеты страницы с помощью Flexbox.

**9.6.1 Двуколоночный дизайн**

Далее представлен пример представления страницы, состоящей из двух колонок, заданных с помощью свойств flexbox.

**<head> <style>  
 \*{ box-sizing: border-box; }  
 html, body {padding: 1; margin: 1; font-family: verdana, arial, sans-serif;}  
 body { display: flex; padding: 1.1em; flex-direction: column;}  
 .item {background-color: #9400D3; font-size: 1.1em; padding: 1em;}  
 .item:nth-child(even) {background-color: #DDA0DD; }  
 @media screen and (min-width: 600px) {body { flex-direction: row;}}  
 </style></head> <body><div class="item">  
 <h2>Певица Люся Чеботина</h2>  
 <p> Родилась 26 апреля 1997 года в Петропавловске-Камчатском. Окончила Государственный музыкальный колледж эстрадного и джазового искусства. С 2016-го девушка записывала композиции собственного сочинения. 29 октября 2021 года выпустила второй студийный альбом “The End”. Ведущий сингл с альбома “Солнце Монако” уже к середине ноября занял 6 строчку в Apple Music и чарте ВКонтакте, затем добрался до 2-го места в “Яндекс Музыке”. В 2023-м поп-певица записала совместный трек “Королева” с Филиппом Киркоровым. Также долгое время в российских чартах первые строчки занимали такие композиции Люси, как “Аэроэкспресс”, “Плакал Голливуд”. </p> </div>  
<div class="item"> <h2> Песня “Солнце Монако”</h2> <p>Зачем мне солнце Монако. Для чего, скажи мне, луна Сен-Тропе. Когда твой взгляд светит ярко. В этом смысла ноль, если тебя рядом нет. </p></div> </body>**

Реализация данного кода представлена на рисунке 9.12.

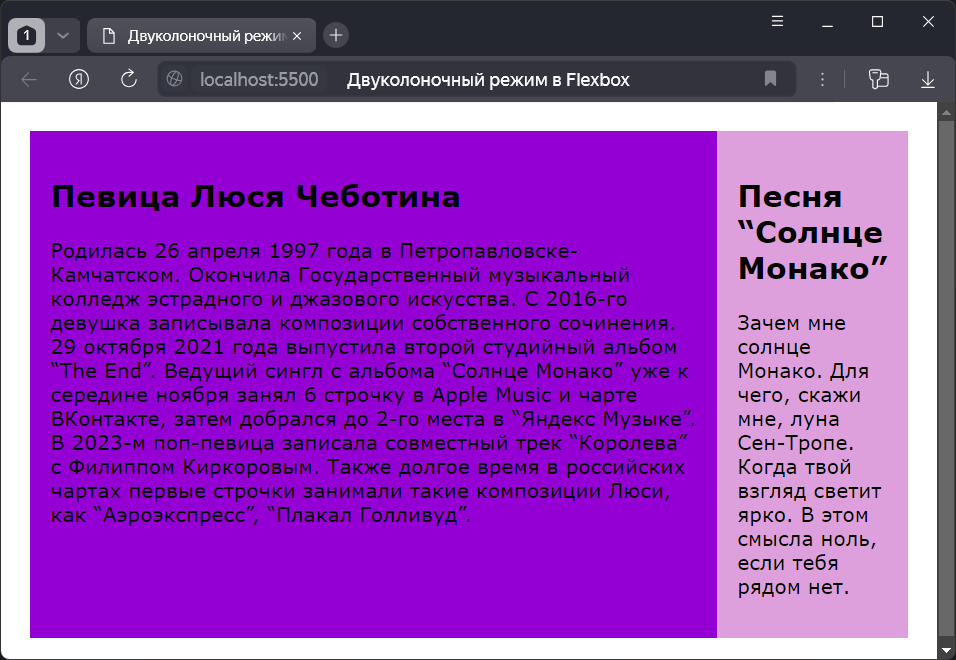


Рисунок 9.12 - Двуколоночный режим в Flexbox Здесь flex-контейнером является элемент body. Так как на мобильных устройствах (особенно смартфонах) размер экрана не такой большой, поэтому по умолчанию устанавливается расположение элементов в столбик. Однако для устройств с экраном от 600рх и выше действует правило media-query, которое устанавливает расположение в виде строки.

**9.6.2 Трехколоночный режим**

Далее представлен пример создания гибкого макета веб-приложения, в котором заданы три колонки.

**<head> <style>**

**\*{ box-sizing: border-box; }**

**html, body { padding: 1; margin: 1; font-family: verdana, arial, sans-serif;}**

**body {display: flex; padding: 1.1em; flex-direction: column; }**

**.item {background-color: PaleTurquoise; font-size: 1.1em; padding: 1em; flex: 1;}**

**.item:nth-child(1) { background-color: DarkTurquoise;}**

**@media screen and (min-width: 600px) { body {flex-direction: row;}**

**.item:nth-child(2) {order: -1;}}**

**</style> </head>**

**<body> <div class="item">**

**<h1>Певец Дима Билан</h1>**

**<p>Сверхпопулярный поп-исполнитель, первый российский певец, победивший в конкурсе “Евровидение” - Дима Билан – родился в 1981-м году в Карачаево-Черкессии. С ранних лет мальчик тянулся к творчеству, он с большим удовольствием учился в музыкальной школе, участвовал в городских конкурсах и фестивалях.**  **</p> </div>**

**<div class="item">**

**<h3>Награды певца</h3>**

**<p>Дима Билан является многократным лауреатом различных музыкальных наград, в числе которых премии “**[**МУЗ-ТВ**](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%BC%D0%B8%D1%8F_%D0%9C%D1%83%D0%B7-%D0%A2%D0%92)**”,**[**RU.TV**](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%BC%D0%B8%D1%8F_RU.TV)**,** [**ZD Awards**](https://ru.wikipedia.org/wiki/%D0%97%D0%B2%D1%83%D0%BA%D0%BE%D0%B2%D0%B0%D1%8F_%D0%B4%D0%BE%D1%80%D0%BE%D0%B6%D0%BA%D0%B0_%D0%9C%D0%9A)**,**[**MTV EMA**](https://ru.wikipedia.org/wiki/MTV_Europe_Music_Awards)**, телевизионный фестиваль “**[**Песня года**](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%81%D0%BD%D1%8F_%D0%B3%D0%BE%D0%B4%D0%B0)**” и другие. 14-кратныйобладатель национальной музыкальной премии “**[**Золотой граммофон**](https://ru.wikipedia.org/wiki/%D0%97%D0%BE%D0%BB%D0%BE%D1%82%D0%BE%D0%B9_%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BE%D1%84%D0%BE%D0%BD)**” радиостанции “**[**Русское Радио**](https://ru.wikipedia.org/wiki/%D0%A0%D1%83%D1%81%D1%81%D0%BA%D0%BE%D0%B5_%D1%80%D0%B0%D0%B4%D0%B8%D0%BE)**”. Рекордсмен по количеству наград премий**[**MTV RMA**](https://ru.wikipedia.org/wiki/MTV_Russia_Music_Awards)**и “МУЗ-ТВ”. </p> </div>**

**<div class="item">**

**<h3>Топ-10 лучших песен артиста</h3>**

**<ul> <li>“**[**Ночной хулиган**](https://ru.wikipedia.org/wiki/%D0%AF_%D0%BD%D0%BE%D1%87%D0%BD%D0%BE%D0%B9_%D1%85%D1%83%D0%BB%D0%B8%D0%B3%D0%B0%D0%BD)**”</li>**

**<li>“**[**Believe**](https://ru.wikipedia.org/wiki/Believe_(%D0%B0%D0%BB%D1%8C%D0%B1%D0%BE%D0%BC_%D0%94%D0%B8%D0%BC%D1%8B_%D0%91%D0%B8%D0%BB%D0%B0%D0%BD%D0%B0))**”</li>**

**<li>“**[**Я тебя помню**](https://ru.wikipedia.org/wiki/%D0%92%D1%80%D0%B5%D0%BC%D1%8F-%D1%80%D0%B5%D0%BA%D0%B0)**”</li>**

**<li>“Невозможное возможно”</li>**

**<li>“Это была любовь” </li>**

**<li>“На берегу неба” </li>**

**<li>“Неделимые” </li>**

**<li>“Малыш"” </li>**

**<li>“Молния” </li>**

**<li>“Ты должна рядом быть” </li></ul> </p> </div>**

**</body>**

Реализация данного кода представлена на рисунке 9.13.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 9.13 - Трехколоночный режим с равными столбцами во Flexbox

В отличие от предыдущего примера здесь добавлен еще один элемент. Особенностью этого примера является то, что столбцы имеют одинаковые размеры. Для этого у них установлено свойство flex: 1, то есть при растяжении или уменьшении границ контейнера все элементы будут масштабироваться на равную величину.

И кроме того, при ширине экрана больше 600px у второго элемента устанавливается свойство order: -1, благодаря чему этот элемент помещается первым.

Подобным образом можно добавить и большее количество столбцов. Но в данном случае по умолчанию столбцы имеют одинаковую ширину. В том случае, если один из столбцов должен иметь ширину больше, чем у остальных добавляется в стили страницы следующее правило:

**.item:first-child { flex: 0 0 50%;}**

В этом случае первый элемент всегда будет занимать 50% пространства контейнера (рисунок 9.14):

Изображение выглядит как текст, электроника, компьютер, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 9.14 - Трехколоночный дизайн в Flexbox и CSS3

**9.7 Пример макета страницы на Flexbox**

Далее приведен приме создания гибкого макета страницы, состоящего из шапки, футера и центральной части, в которой задано три столбца: основное содержимое и два сайдбара. Код веб-страницы представлен ниже.

**<head> <style>**

**\* { box-sizing: border-box; }**

**html, body {padding: 1; margin: 1; font-family: verdana, arial, sans-serif; }**

**body {color:white;**

**font-size: 1.2em;**

**padding: 1.1em;**

**display: flex;**

**flex-direction: column;}**

**main {display: flex; flex-direction: column;}**

**article {background-color: LimeGreen; flex: 2 2 12em; padding: 1em; }**

**nav,aside {flex: 1; background-color: YellowGreen ;}**

**nav { order: -1;}**

**header, footer { flex: 0 0 5em; background-color: SeaGreen;}**

**@media screen and (min-width: 600px) { body {min-height: 100vh; }**

**main {flex-direction: row; min-height: 100%; flex: 1 1 auto;}}**

**</style></head>**

**<body> <header>**

**<h1>Золотой голос России - Муслим Магомаев</h1>**

**<h2> Веб-страница, посвященная творчеству Муслима Магомаева<h2> </header>**

**<main> <article>**

**<h1> Муслим Магометович Магомаев </h1>**

**<p> Советский, азербайджанский и российский эстрадный и оперный певец, композитор, киноактёр. Народный артист СССР. За время карьеры достиг популярности как в странах бывшего СССР, так и других странах Восточной Европы и Центральной Азии. </p></article>**

**<nav> <h2>Лучшие песни</h2> </nav>**

**<a href=azerb.html>“Азербайджан”</a>**

**<a href=verni.html>“Верни мне музыку”</a>**

**<a href=poimi.html>“Пойми меня” </a>**

**<a href=cvet.html>“Цветные сны”</a>**

**<a href=melod.html>“Мелодия”</a>**

**<a href=podm.html>“Подмосковные вечера”</a>**

**<a href=padaed.html>“Падает снег”</a>**

**<a href=luchshii.html>“Лучший город земли”</a>**

**<a href=koroleva.html>“Королева красоты”</a>**

**<a href=siniya.html>“Синяя вечность”</a>**

**<aside><h2>Семья артиста</h2> <p>Муслим Магомаев родился 17 августа 1942 года в Баку. Отец, Магомет Магомаев, погиб на фронте, не дожив до Победы 15 дней. Мама Муслима Айшет Магомаева - драматическая актриса. Муслим считал себя по национальности азербайджанцем, а Россию - матерью. В детстве Муслим Магомаев рос в семье дяди**[**Джамала Муслимовича**](https://24smi.org/celebrity/283091-dzhamal-eddin-magomaev.html)**. </p> </aside> </main>**

**<footer> <p>Последние годы жизни он прожил в Москве, отказываясь от концертных выступлений. Занимался живописью, вёл переписку с поклонниками через свой**[**веб-сайт**](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82)**.</p> </footer> </body>**

Реализация данного кода представлена на рисунке 9.15.

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 9.15 - Макет верстки на flexbox в HTML5 и CSS3

В данном примере, flex-контейнером верхнего уровня здесь является элемент body. Его flex-элементами являются header, main и footer. Body располагает все свои элементы сверху вниз в столбик. Здесь также стоит отметить, что при ширине от 600px и выше для заполнения всего пространства браузера у body устанавливается стиль height: 100vh;. Элементы header и footer аналогичны. Их свойство flex: 0 0 5em; указывают, что при любом изменении контейнера эти элементы будут иметь размер в 5em. То есть они имеют статический размер.

Более сложным является элемент main, который определяет основное содержимое. При этом, будучи flex-элементом, он также является flex-контейнером для вложенных элементов и управляет их позиционированием. При ширине браузера до 600px он располагает элементы в столбик, что очень удобно на мобильных устройствах. При ширине от 600px вложенные элементы nav, article и aside располагаются в виде строки. И поскольку при такой ширине браузера родительский элемент body заполняет по высоте все пространство браузера, то для заполнения всей высоты контейнера body при его изменении у элемента main устанавливается свойство flex: 1 1 auto;.

У вложенных в main flex-элементов стоит отметить, что элемент навигации nav и элемент сайдбара aside будут иметь одинаковые размеры при масштабировании контейнера. А элемент article, содержащий основное содержимое, будет соответственно больше. При этом хотя nav определен после элемента article, но благодаря установке свойства order: -1 блок навигации будет стоять до блока article.