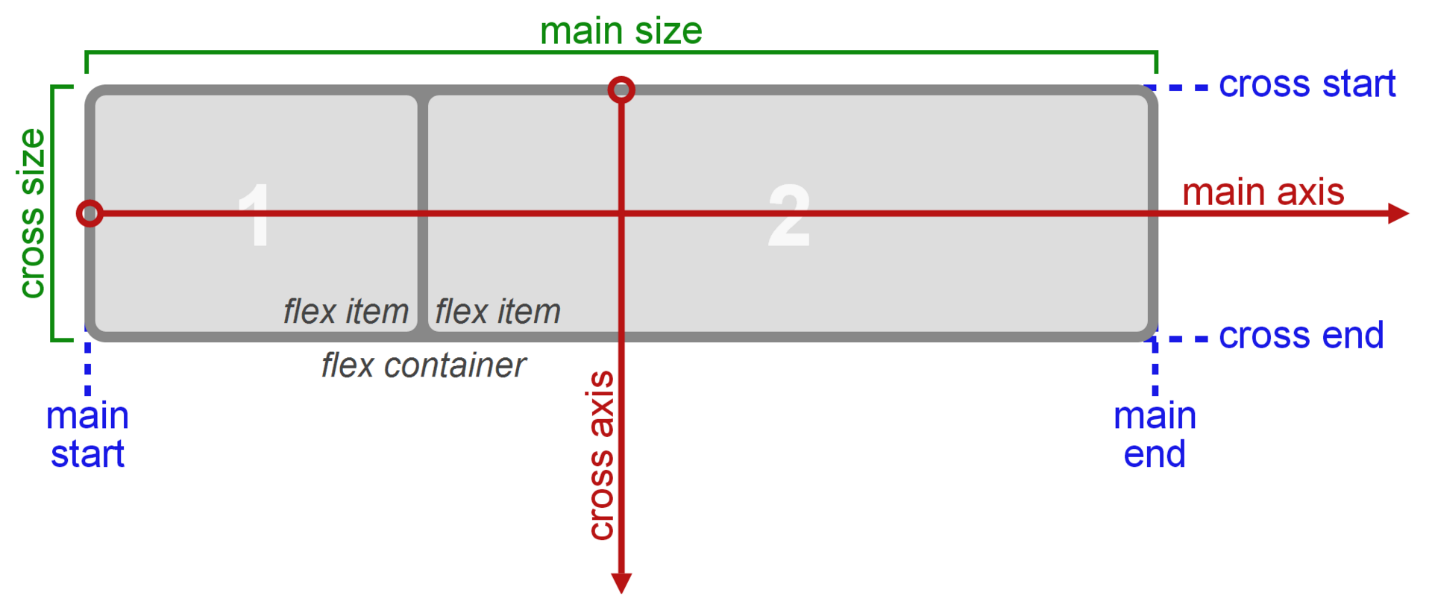
**Что такое flexbox**

* Содержание:
* 1. Основные понятия
* 2. Flex-контейнер
* 3. Flex-элементы
* 4. Порядок отображения flex-элементов и ориентация
* 4.1. Направление главной оси: flex-direction
* 4.2. Управление многострочностью flex-контейнера: flex-wrap
* 4.3. Краткая запись направления и многострочности: flex-flow
* 4.4. Порядок отображения flex-элементов: order
* 5. Гибкость flex-элементов
* 5.1. Задание гибких размеров одним свойством: flex
* 5.2. Коэффициент роста: flex-grow
* 5.3. Коэффициент сжатия: flex-shrink
* 5.4. Базовый размер: flex-basis
* 6. Выравнивание
* 6.1. Выравнивание по главной оси: justify-content
* 6.2. Выравнивание по поперечной оси: align-items и align-self
* 6.3. Выравнивание строк flex-контейнера: align-content

**Поддержка браузерами**

***IE:****11.0, 10.0 -ms-****Firefox:****28.0, 18.0 -moz-****Chrome:****29.0, 21.0 -webkit-****Safari:****6.1 -webkit-****Opera:****12.1 -webkit-****iOS Safari:****7.0 -webkit-****Opera Mini:****8****Android Browser:****4.4, 4.1 -webkit-****Chrome for Android:****44*

**1. Основные понятия**

Рис. 1. Модель flexbox

Для описания модуля Flexbox используется определенный набор терминов. Значение flex-flow и режим записи определяют соответствие этих терминов физическим направлениям: верх / право / низ / лево, осям: вертикальная / горизонтальная и размерам: ширина / высота.

**Главная ось (main axis)** — ось, вдоль которой выкладываются flex-элементы. Она простирается в основном измерении.

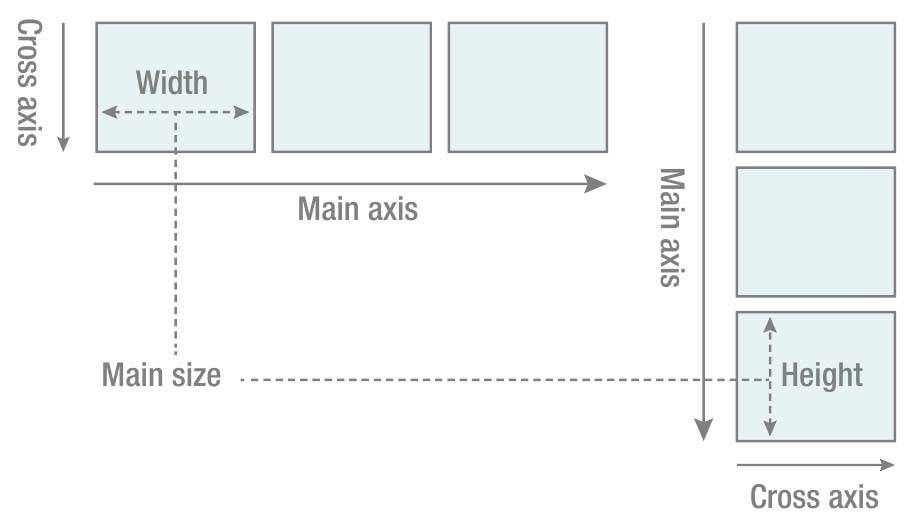
**Main start и main end** — линии, которые определяют начальную и конечную стороны flex-контейнера, относительно которых выкладываются flex-элементы (начиная с main start по направлению к main end).

**Основной размер (main size**) — ширина или высота flex-контейнера или flex-элементов, в зависимости от того, что из них находится в основном измерении, определяют основной размер flex-контейнера или flex-элемента.

**Поперечная ось (cross axis)** — ось, перпендикулярная главной оси. Она простирается в поперечном измерении.

**Cross start и cross end** — линии, которые определяют начальную и конечную стороны поперечной оси, относительно которых выкладываются flex-элементы.

**Поперечный размер (cross size)** — ширина или высота flex-контейнера или flex-элементов, в зависимости от того, что находится в поперечном измерении, являются их поперечным размером.

Рис. 2. Режим строки и колонки

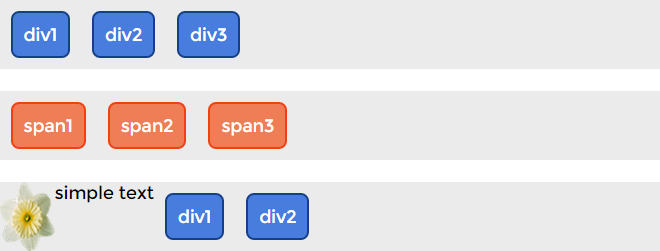
**2. Flex-контейнер**

Flex-контейнер устанавливает новый гибкий контекст форматирования для его содержимого. Flex-контейнер не является блочным контейнером, поэтому для дочерних элементов не работают такие CSS-свойства, как float, clear, vertical-align. Также, на flex-контейнер не оказывают влияние свойства column-\*, создающие колонки в тексте и псевдоэлементы ::first-line и ::first-letter.

Модель flexbox-разметки связана с определенным значением CSS-свойства display родительского html-элемента, содержащего внутри себя дочерние блоки. Для управления элементами с помощью этой модели нужно установить свойство display следующим образом:

.flex-container { /\*генерирует flex-контейнер уровня блока\*/ display: -webkit-flex; display: flex; } .flex-container { /\*генерирует flex-контейнер уровня строки\*/ display: -webkit-inline-flex; display: inline-flex; }

После установки данных значений свойства каждый дочерний элемент автоматически становится flex-элементом, выстраиваясь в один ряд (вдоль главной оси). При этом блочные и строчные дочерние элементы ведут себя одинаково, т.е. ширина блоков равна ширине их содержимого с учетом внутренних полей и рамок элемента.

Рис. 3. Выравнивание элементов в модели flexbox

Если родительский блок содержит текст или изображения без оберток, они становятся анонимными flex-элементами. Текст выравнивается по верхнему краю блока-контейнера, а высота изображения становится равной высоте блока, т.е. оно деформируется.

**3. Flex-элементы**

Flex-элементы — блоки, представляющие содержимое flex-контейнера в потоке. Flex-контейнер устанавливает новый контекст форматирования для своего содержимого, который обуславливает следующие особенности:

* Для flex-элементов блокируется их значение свойства display. Значение display: inline-block; и display: table-cell; вычисляется в display: block;.
* Пустое пространство между элементами исчезает: оно не становится своим собственным flex-элементом, даже если межэлементный текст обернут в анонимный flex-элемент. Для содержимого анонимного flex-элемента невозможно задать собственные стили, но оно будет наследовать их (например, параметры шрифта) от flex-контейнера.
* Абсолютно позиционированный flex-элемент не участвует в компоновке гибкого макета.
* Поля margin соседних flex-элементов не схлопываются.
* Процентные значения margin и padding вычисляются от внутреннего размера содержащего их блока.
* margin: auto; расширяются, поглощая дополнительное пространство в соответствующем измерении. Их можно использовать для выравнивания или раздвигания смежных flex-элементов.
* Автоматический минимальный размер flex-элементов по умолчанию является минимальным размером его содержимого, то есть min-width: auto;. Для контейнеров с прокруткой автоматический минимальный размер обычно равен нулю.

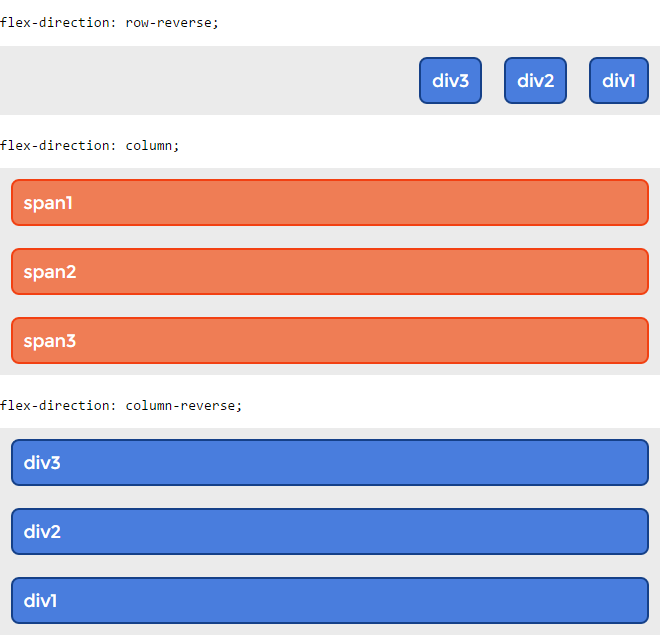
**4. Порядок отображения flex-элементов и ориентация**

Содержимое flex-контейнера можно разложить в любом направлении и в любом порядке (переупорядочение flex-элементов внутри контейнера влияет только на визуальный рендеринг).

**4.1. Направление главной оси: flex-direction**

Свойство относится к flex-контейнеру. Управляет направлением главной оси, вдоль которой укладываются flex-элементы, в соответствии с текущим режимом записи. Не наследуется.

|  |  |
| --- | --- |
| **flex-direction** | |
| Значения: |  |
| row | Значение по умолчанию, слева направо (в rtl справа налево). Flex-элементы выкладываются в строку. Начало (main-start) и конец (main-end) направления главной оси соответствуют началу (inline-start) и концу (inline-end) инлайн-оси (inline-axis). |
| row-reverse | Направление справа налево (в rtl слева направо). Flex-элементы выкладываются в строку относительно правого края контейнера (в rtl — левого). |
| column | Направление сверху вниз. Flex-элементы выкладываются в колонку. |
| column-reverse | Колонка с элементами в обратном порядке, снизу вверх. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

Рис. 4. Свойство flex-direction для left-to-right языков

**Синтаксис**

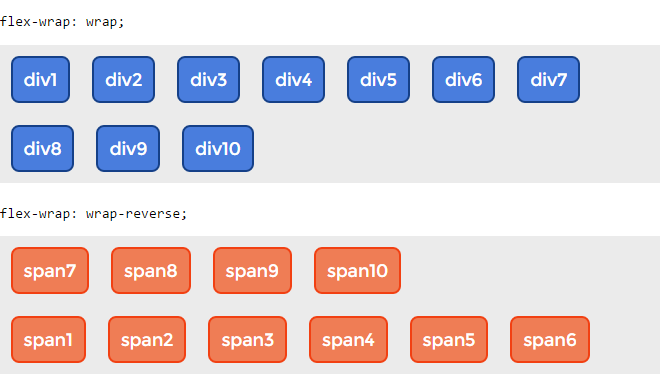
.flex-container { display: -webkit-flex; -webkit-flex-direction: row-reverse; display: flex; flex-direction: row-reverse; }

**4.2. Управление многострочностью flex-контейнера: flex-wrap**

Свойство определяет, будет ли flex-контейнер однострочным или многострочным, а также задает направление поперечной оси, определяющее направление укладки новых линий flex-контейнера.

По умолчанию flex-элементы укладываются в одну строку, вдоль главной оси. При переполнении они будут выходить за пределы ограничивающей рамки flex-контейнера. Свойство не наследуется.

|  |  |
| --- | --- |
| **flex-wrap** | |
| Значения: |  |
| nowrap | Значение по умолчанию. Flex-элементы не переносятся, а располагаются в одну линию слева направо (в rtl справа налево). |
| wrap | Flex-элементы переносятся, располагаясь в несколько горизонтальных рядов (если не помещаются в один ряд) в направлении слева направо (в rtl справа налево). |
| wrap-reverse | Flex-элементы переносятся на новые линии, располагаясь в обратном порядке слева-направо, при этом перенос происходит снизу вверх. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

Рис. 5. Управление многострочностью с помощью свойства flex-wrap для LTR-языков

**Синтаксис**

.flex-container { display: -webkit-flex; -webkit-flex-wrap: wrap; display: flex; flex-wrap: wrap; }

**4.3. Краткая запись направления и многострочности: flex-flow**

Свойство позволяет определить направления главной и поперечной осей, а также возможность переноса flex-элементов при необходимости на несколько строк. Представляет собой сокращённую запись свойств flex-direction и flex-wrap. Значение по умолчанию flex-flow: row nowrap;. свойство не наследуется.

|  |  |
| --- | --- |
| **flex-flow** | |
| Значения: |  |
| направление | Указывает направление главной оси. Значение по умолчанию row. |
| многострочность | Задаёт многострочность поперечной оси. Значение по умолчанию nowrap. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

**Синтаксис**

.flex-container { display: -webkit-flex; -webkit-flex-flow: row wrap; display: flex; flex-flow: row wrap; }

**4.4. Порядок отображения flex-элементов: order**

Свойство определяет порядок, в котором flex-элементы отображаются и располагаются внутри flex-контейнера. Применяется к flex-элементам. Свойство не наследуется.

Первоначально все flex-элементы имеют order: 0;. При указании значения от -1 для элемента он перемещается в начало сроки, значение 1 — в конец. Если несколько flex-элементов имеют одинаковое значение order, они будут отображаться в соответствии с исходным порядком.

|  |  |
| --- | --- |
| **order** | |
| Значения: |  |
| число | Свойство задается целым числом, отвечающим за порядок отображения flex-элементов. Значение по умолчанию 0. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

**Синтаксис**

.flex-container { display: -webkit-flex; display: flex; } .flex-item { -webkit-order: 1; order: 1; }

Рис. 6. Порядок отображения flex-элементов

**5. Гибкость flex-элементов**

Определяющим аспектом гибкого макета является возможность «сгибать» flex-элементы, изменяя их ширину / высоту (в зависимости от того, какой размер находится на главной оси), чтобы заполнить доступное пространство в основном измерении. Это делается с помощью свойства flex. Flex-контейнер распределяет свободное пространство между своими дочерними элементами (пропорционально их коэффициенту flex-grow) для заполнения контейнера или сжимает их (пропорционально их коэффициенту flex-shrink), чтобы предотвратить переполнение.

Flex-элемент будет полностью «негибок», если его значения flex-grow и flex-shrink равны нулю, и «гибкий» в противном случае.

**5.1. Задание гибких размеров одним свойством: flex**

Свойство является сокращённой записью свойств flex-grow, flex-shrink и flex-basis. Значение по умолчанию: flex: 0 1 auto;. Можно указывать как одно, так и все три значения свойств. Свойство не наследуется.

W3C рекомендует использовать сокращённую запись, так как она правильно сбрасывает любые неуказанные компоненты, чтобы подстроиться под типичное использование.

|  |  |
| --- | --- |
| **flex** | |
| Значения: |  |
| коэффициент растяжения | Коэффициент увеличения ширины flex-элемента относительно других flex-элементов. |
| коэффициент сужения | Коэффициент уменьшения ширины flex-элемента относительно других flex-элементов. |
| базовая ширина | Базовая ширина flex-элемента. |
| auto | Эквивалентно flex: 1 1 auto;. |
| none | Эквивалентно flex: 0 0 auto;. |
| initial | Устанавливает значение свойства в значение по умолчанию. Эквивалентно flex: 0 1 auto;. |
| inherit | Наследует значение свойства от родительского элемента. |

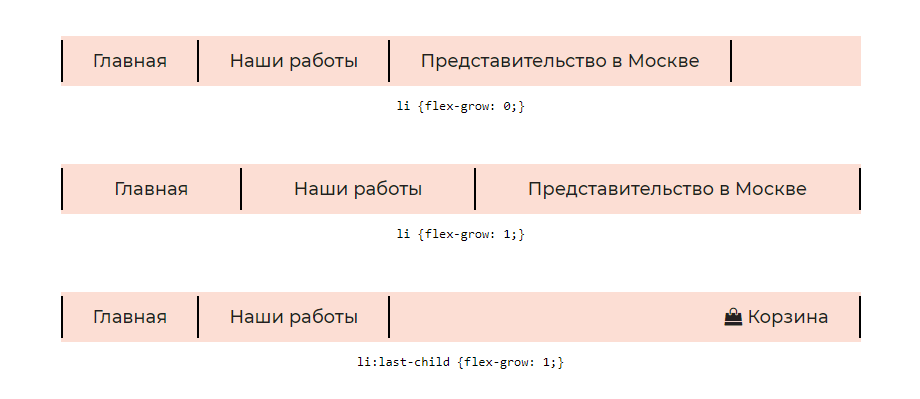
**Синтаксис**

.flex-container { display: -webkit-flex; display: flex; } .flex-item { -webkit-flex: 3 1 100px; -ms-flex: 3 1 100px; flex: 3 1 100px; }

**5.2. Коэффициент роста: flex-grow**

Свойство определяет коэффициент роста одного flex-элемента относительно других flex-элементов в flex-контейнере при распределении положительного свободного пространства. Если сумма значений flex-grow flex-элементов в строке меньше 1, они занимают менее 100% свободного пространства. Свойство не наследуется.

|  |  |
| --- | --- |
| **flex-grow** | |
| Значения: |  |
| число | Положительное целое или дробное число, устанавливающее коэффициент роста flex-элемента. Значение по умолчанию 0. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

Рис. 7. Управление свободным пространством в панели навигации с помощью flex-grow

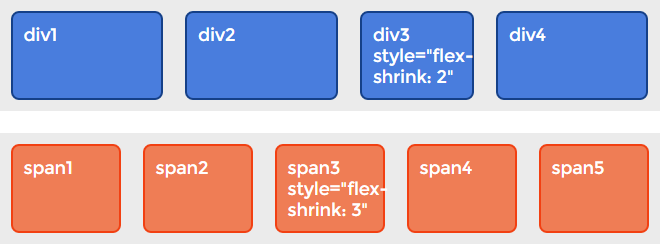
**Синтаксис**

.flex-container { display: -webkit-flex; display: flex; } .flex-item { -webkit-flex-grow: 3; flex-grow: 3; }

**5.3. Коэффициент сжатия: flex-shrink**

Свойство указывает коэффициент сжатия flex-элемента относительно других flex-элементов при распределении отрицательного свободного пространства. Умножается на базовый размер flex-basis. Отрицательное пространство распределяется пропорционально тому, насколько элемент может сжаться, поэтому, например, маленький flex-элемент не уменьшится до нуля, пока не будет заметно уменьшен flex-элемент большего размера. Свойство не наследуется.

|  |  |
| --- | --- |
| **flex-shrink** | |
| Значения: |  |
| число | Положительное целое или дробное число, устанавливающее коэффициент уменьшения flex-элемента. Значение по умолчанию 1. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

Рис. 8. Сужение flex-элементов в строке

**Синтаксис**

.flex-container { display: -webkit-flex; display: flex; } .flex-item { -webkit-flex-shrink: 3; flex-shrink: 3; }

**5.4. Базовый размер: flex-basis**

Свойство устанавливает начальный основной размер flex-элемента до распределения свободного пространства в соответствии с коэффициентами гибкости. Для всех значений, кроме auto и content, базовый гибкий размер определяется так же, как width в горизонтальных режимах записи. Процентные значения определяются относительно размера flex-контейнера, а если размер не задан, используемым значением для flex-basis являются размеры его содержимого. Не наследуется.

|  |  |
| --- | --- |
| **flex-basis** | |
| Значения: |  |
| auto | Значение по умолчанию. Элемент получает базовый размер, соответствующий размеру его содержимого (если он не задан явно). |
| content | Определяет базовый размер в зависимости от содержимого flex-элемента. |
| длина | Базовый размер определяется так же, как для ширины и высоты. Задается в единицах длины. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

**Синтаксис**

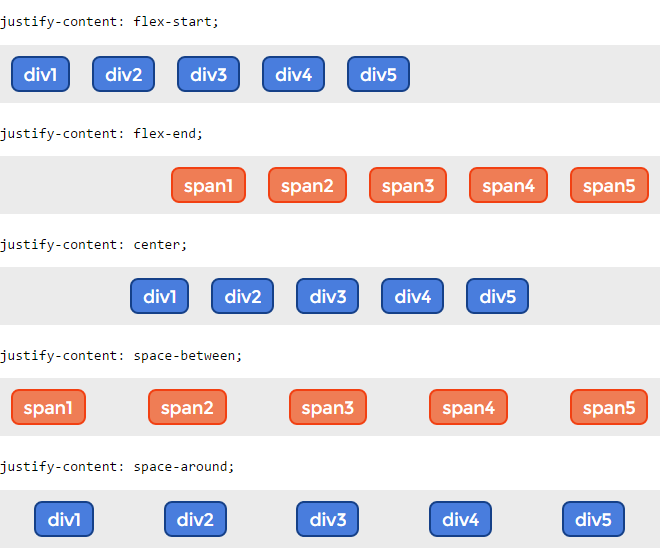
.flex-container { display: -webkit-flex; display: flex; } .flex-item { -webkit-flex-basis: 100px; flex-basis: 100px; }

**6. Выравнивание**

**6.1. Выравнивание по главной оси: justify-content**

Свойство выравнивает flex-элементы по главной оси flex-контейнера, распределяя свободное пространство, незанятое flex-элементами. Когда элемент преобразуется в flex-контейнер, flex-элементы по умолчанию сгруппированы вместе (если для них не заданы поля margin). Промежутки добавляются после расчета значений margin и flex-grow. Если какие-либо элементы имеют ненулевое значение flex-grow или margin: auto;, свойство не будет оказывать влияния. Свойство не наследуется.

|  |  |
| --- | --- |
| **justify-content** | |
| Значения: |  |
| flex-start | Значение по умолчанию. Flex-элементы выкладываются в направлении, идущем от начальной линии flex-контейнера. |
| flex-end | Flex-элементы выкладываются в направлении, идущем от конечной линии flex-контейнера. |
| center | Flex-элементы выравниваются по центру flex-контейнера. |
| space-between | Flex-элементы равномерно распределяются по линии. Первый flex-элемент помещается вровень с краем начальной линии, последний flex-элемент — вровень с краем конечной линии, а остальные flex-элементы на линии распределяются так, чтобы расстояние между любыми двумя соседними элементами было одинаковым. Если оставшееся свободное пространство отрицательно или в строке присутствует только один flex-элемент, это значение идентично параметру flex-start. |
| space-around | Flex-элементы на линии распределяются так, чтобы расстояние между любыми двумя смежными flex-элементами было одинаковым, а расстояние между первым / последним flex-элементами и краями flex-контейнера составляло половину от расстояния между flex-элементами. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

Рис. 9. Выравнивание элементов и распределение свободного пространства с помощью свойства justify-content

**Синтаксис**

.flex-container { display: -webkit-flex; -webkit-justify-content: flex-start; display: flex; justify-content: flex-start; }

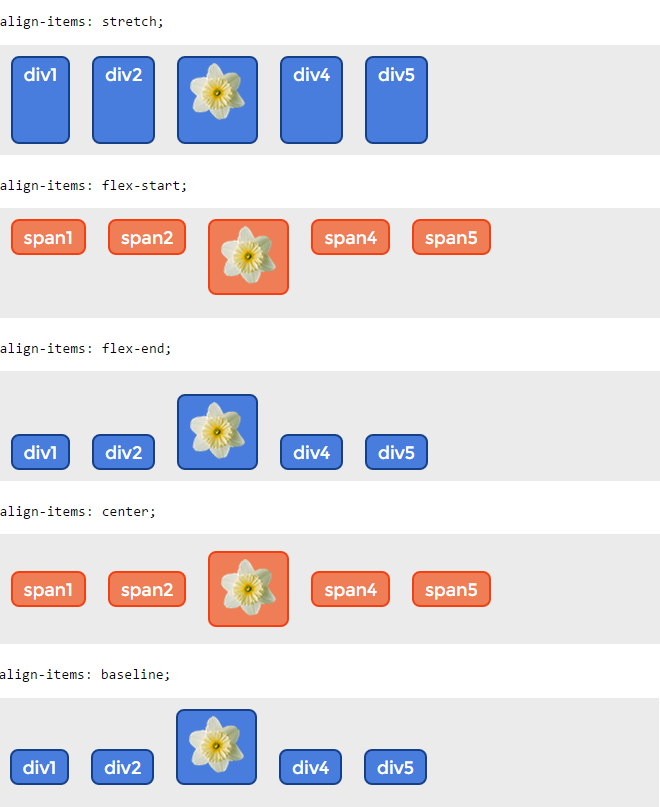
**6.2. Выравнивание по поперечной оси: align-items и align-self**

Flex-элементы можно выравнивать по поперечной оси текущей строки flex-контейнера. align-items устанавливает выравнивание для всех элементов flex-контейнера, включая анонимные flex-элементы. align-self позволяет переопределить это выравнивание для отдельных flex-элементов. Если любое из поперечных margin flex-элемента имеет значение auto, align-self не имеет никакого влияния.

**6.2.1. Align-items**

Свойство выравнивает flex-элементы, в том числе и анонимные flex-элементы по поперечной оси. Не наследуется.

|  |  |
| --- | --- |
| **align-items** | |
| Значения: |  |
| flex-start | Верхний край flex-элемента помещается вплотную с flex-линией (или на расстоянии, с учетом заданных полей margin и рамок border элемента), проходящей через начало поперечной оси. |
| flex-end | Нижний край flex-элемента помещается вплотную с flex-линией (или на расстоянии, с учетом заданных полей margin и рамок border элемента), проходящей через конец поперечной оси. |
| center | Поля flex-элемента центрируется по поперечной оси в пределах flex-линии. |
| baseline | Базовые линии всех flex-элементов, участвующих в выравнивании, совпадают. |
| stretch | Если поперечный размер flex-элемента вычисляется как auto и ни одно из поперечных значений margin не равно auto, элемент растягивается. Значение по умолчанию. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

Рис. 10. Выравнивание элементов в контейнере по вертикали

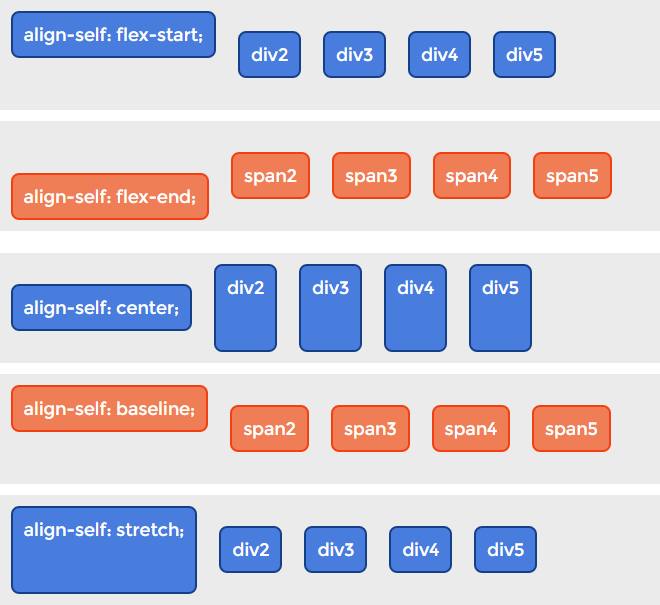
**Синтаксис**

.flex-container { display: -webkit-flex; -webkit-align-items: flex-start; display: flex; align-items: flex-start; }

**6.2.2. Align-self**

Свойство отвечает за выравнивание отдельно взятого flex-элемента по высоте flex-контейнера. Переопределяет выравнивание, заданное align-items. Не наследуется.

|  |  |
| --- | --- |
| **align-self** | |
| Значения: |  |
| auto | Значение по умолчанию. Flex-элемент использует выравнивание, указанное в свойстве align-items flex-контейнера. |
| flex-start | Верхний край flex-элемента помещается вплотную с flex-линией (или на расстоянии, с учетом заданных полей margin и рамок border элемента), проходящей через начало поперечной оси. |
| flex-end | Нижний край flex-элемента помещается вплотную с flex-линией (или на расстоянии, с учетом заданных полей margin и рамок border элемента), проходящей через конец поперечной оси. |
| center | Поля flex-элемента центрируется по поперечной оси в пределах flex-линии. |
| baseline | Flex-элемент выравнивается по базовой линии. |
| stretch | Если поперечный размер flex-элемента вычисляется как auto и ни одно из поперечных значений margin не равно auto, элемент растягивается. Значение по умолчанию. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

Рис. 11. Выравнивание отдельных flex-элементов

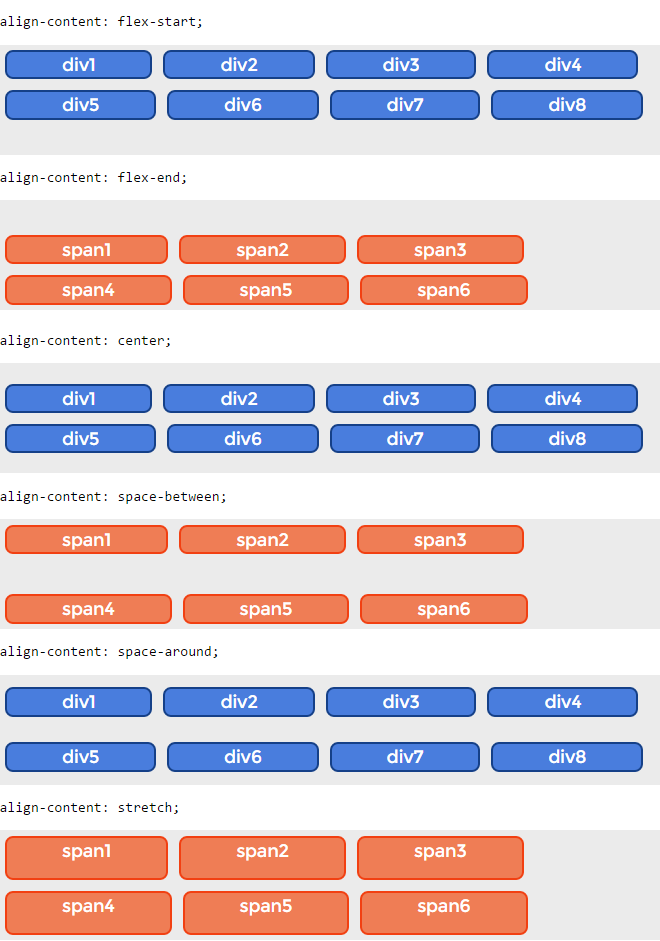
**Синтаксис**

.flex-container { display: -webkit-flex; display: flex; } .flex-item { -webkit-align-self: center; align-self: center; }

**6.3. Выравнивание строк flex-контейнера: align-content**

Свойство выравнивает строки в flex-контейнере при наличии дополнительного пространства на поперечной оси, аналогично выравниванию отдельных элементов на главной оси с помощью свойства justify-content. Свойство не влияет на однострочный flex-контейнер. Не наследуется.

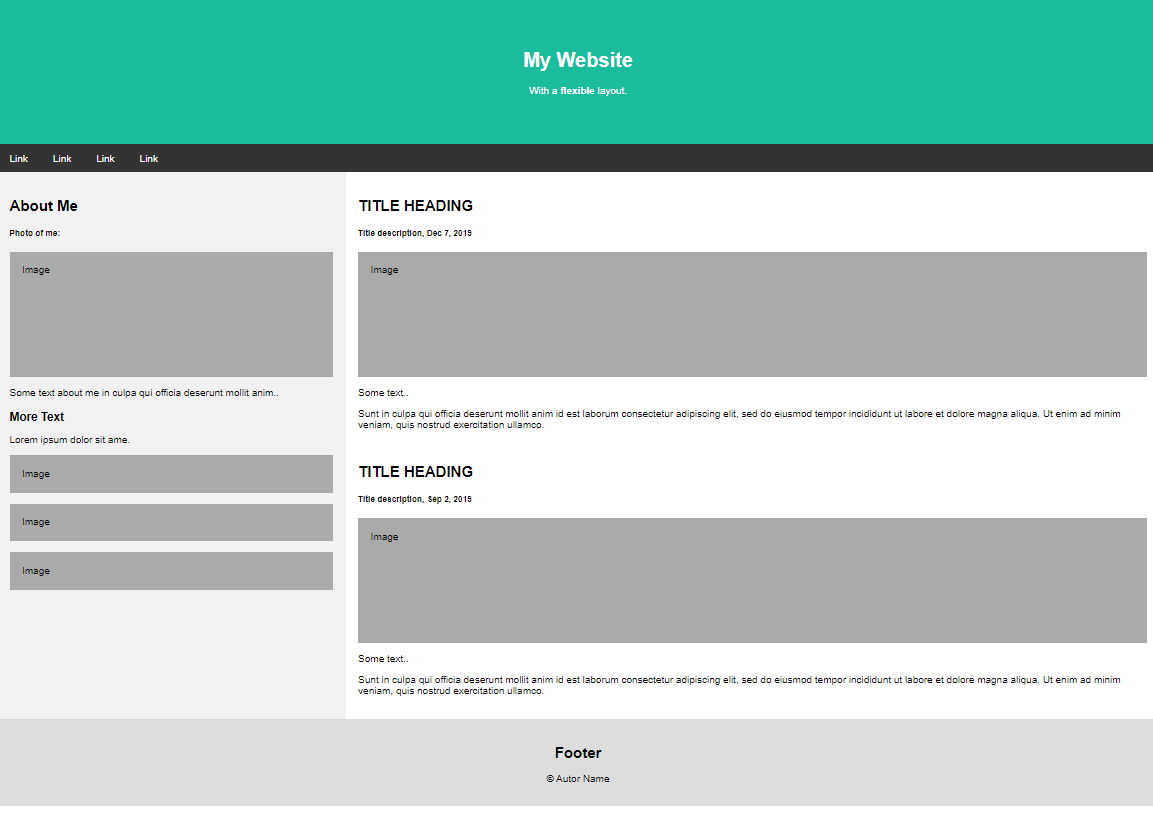
|  |  |
| --- | --- |
| **align-content** | |
| Значения: |  |
| flex-start | Строки укладываются по направлению к началу flex-контейнера. Край первой строки помещается вплотную к краю flex-контейнера, каждая последующая — вплотную к предыдущей строке. |
| flex-end | Строки укладываются по направлению к концу flex-контейнера. Край последней строки помещается вплотную к краю flex-контейнера, каждая предыдущая — вплотную к последующей строке. |
| center | Строки укладываются по направлению к центру flex-контейнера. Строки расположены вплотную друг к другу и выровнены по центру flex-контейнера с равным расстоянием между начальным краем содержимого flex-контейнера и первой строкой и между конечным краем содержимого flex-контейнера и последней строкой. |
| space-between | Строки равномерно распределены в flex-контейнере. Если оставшееся свободное пространство отрицательно или в flex-контейнере имеется только одна flex-линия, это значение идентично flex-start. В противном случае край первой строки помещается вплотную к начальному краю содержимого flex-контейнера, край последней строки — вплотную к конечному краю содержимого flex-контейнера. Остальные строки распределены так, чтобы расстояние между любыми двумя соседними строками было одинаковым. |
| space-around | Строки равномерно распределены в flex-контейнере с половинным пробелом на обоих концах. Если оставшееся свободное пространство отрицательно, это значение идентично центcenter. В противном случае строки распределяются таким образом, чтобы расстояние между любыми двумя соседними строками было одинаковым, а расстояние между первой / последней строками и краями содержимого flex-контейнера составляло половину от расстояния между строками. |
| stretch | Значение по умолчанию. Строки flex-элементов равномерно растягиваются, заполняя все доступное пространство. Если оставшееся свободное пространство отрицательно, это значение идентично flex-start. В противном случае свободное пространство будет разделено поровну между всеми строками, увеличивая их поперечный размер. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

Рис. 12. Многострочное выравнивание flex-элементов

**Синтаксис**

.flex-container { display: -webkit-flex; -webkit-flex-flow: row wrap; -webkit-align-content: flex-end; display: flex; flex-flow: row wrap; align-content: flex-end; height: 100px; }

*Задание. Создать адаптивную страницу*



**Шапка сайта**

Для шапки сайта создадим элемент <header></header>. Внутри блока разместим заголовок первого уровня и абзац с текстом.

**html код**

<header>

<h1>My Website</h1>

<p>With a <b>flexible</b> layout.</p>

<header>

Данному блоку установим свойства CSS, обеспечивающие отступ на **60 px**, выравнивания текста по **центру**, установим цвет заднего фона **#1abc9c** и сделаем цвет текста **белым**. Используйте свойства background, color, padding, text-align

**Горизонтальный блок навигации**

Для создания блока навигации воспользуемся тегом <nav></nav>. Внутри элемента расположим пустые ссылки.

**html код**

<nav>

<a href="#">Link</a>

<a href="#">Link</a>

<a href="#">Link</a>

<a href="#">Link</a>

</nav>

Для данного элемента установим метод отображения flex, цвет заливки #333.

Для ссылок, расположенных внутри элемента <nav></nav> необходимо установить свойства CSS, обеспечивающие следующие параметры:

Цвет текста – белый,

Отступы по вертикали – 14px

Отступы по горизонтали – 20px

Выравнивание текста внутри элемента по центру

Линия под ссылкой отсутствует

Цвет заднего фона при наведении - #ddd

Цвет текста при наведении – черный

Используем свойства

color, padding, text-decoration, text-align, background-color и псевдо класс hover

**Блок основных данных**

Блоки с основным содержимым поместим внутри элемента <section></section>. Данный элемент должен быть flex элементом с возможность переноса вложенных элементов по строкам (display, flex-wrap).

Внутри элемента <section></section> разместим два блока – блок с боковой панелью <aside><aside> и основной блок <div class=main></div>.

**Боковая панель**

Боковую панель создадим с помощью тегов <aside></aside>. Внутри элемента разместите информацию о себе.

**Примерный html код**

<aside>

<h2>About Me</h2>

<h5>Photo of me:</h5>

<div class="fakeimg" style="height:200px;">Image</div>

<p>Some text about me...</p>

<h3>More Text</h3>

<p>Text text text text.</p>

<div class="fakeimg" style="height:60px;">Image</div><br>

<div class="fakeimg" style="height:60px;">Image</div><br>

<div class="fakeimg" style="height:60px;">Image</div>

</aside>

С помощью свойств CSS сделайте так, что бы данный блок занимал 30% экрана, установите отступы со всех сторон 20px. Цвет фона установите #f1f1f1.

(flex, background-color, padding)

**Основной блок**

Так же как и блок <aside></aside>, данный блок должен находится внутри блока <section></section>. Примерное содержимое блока указано ниже.

**Примерный html код**

<div class="main">

<h2>TITLE HEADING</h2>

<h5>Title description, Dec 7, 2019</h5>

<div class="fakeimg" style="height:200px;">Image</div>

<p>Some text..</p>

<p>Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.</p>

<br>

<h2>TITLE HEADING</h2>

<h5>Title description, Sep 2, 2019</h5>

<div class="fakeimg" style="height:200px;">Image</div>

<p>Some text..</p>

<p>Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.</p>

</div>

Для данного блока установите следующие параметры:

Цвет фона – белый, отступы – 20px, ширина – 70%.

**Подвал**

Подвал сайта разместите сразу после элемента <section></section>.

Установите цвет фона #ddd, тексты отцентрируйте и установите отступы 20px.

**Примерный html код**

<h2>Footer </h2>

<p>&copy Autor Name</p>

</footer>

**Адаптивность**

Чтобы добавить адаптивность страницы в CSS коде необходимо добавить медиазапрос, устанавливающий правила для экрана с максимальной шириной 700px. Данное правило должно поменять направление главной оси для элементов section и nav на вертикальное – сверху-вниз.

@media screen and (max-width: 700px) {

section, nav {

flex-direction: column;

}

}