1. Почему в большинстве ситуаций стоит использовать flexbox-позиционирование?
2. Способ позиционирования Flexbox поддерживается практически всеми современными браузерами.
3. Для большинства случаев достаточно использование трех свойств: display: flex; justify-content, align-items.
4. Быстрое выравнивание элементов по горизонтали и вертикали.
5. Позволяет создать адаптивный дизайн намного проще.
6. Простое создание сложных макетов
7. В верстке можно использовать HTML таблицы и CSS таблицы. Для создания HTML таблицы используется тег <table>, а CSS таблицы – имитация того же действия, но с помощью CSS свойств. В CSS для этого используется свойство display: table
8. Динамическое вертикальное выравнивание по центру

element {

display: table;

}

Затем для элемента, который дочерний в HTML-разметке:

element {

display: table-cell;

text-align: center;

vertical-align: middle;

}

Ячейка растягивается во всю ширину страницы, а содержимое центрируется.

1. Динамическое горизонтальное выравнивание по центру

element {

display: table;

margin: auto;

}

1. Адаптивная верстка: переключив у элемента свойство display c table-cell на block, сможем расположить элементы в столбик.
2. Динамический прилипающий подвал

display {

table-row;

}

Использование display: table хорошо для семантической верстки и позволяет избавиться от лишних тегов.

1. Во флекс-верстке есть главная ось и поперечная. По умолчанию главная ось направлена слева направо. А поперечная ось сверху вниз. Поперечная ось всегда перпендикулярна главной.

Направление главной оси можно менять с помощью свойства flex-direction, которое можно принимать следующие значения: row-слева направо (по умолчанию), row-reverse (справа налево), column (сверху вниз), column-reverse (снизу вверх). При использовании значения column поперечная ось также меняет свое направление и становится слева направо (аналогично с column-reverse).

Направление поперечной оси задается с помощью свойства flex-wrap со значениями wrap и wrap-reverse. По умолчанию flex-элементы не переносят на новую строку. Для переноса используют свойство flex-wrap.

Свойства flex-wrap и flex-direction можно записать с помощью универсального свойства flex-flow, где первое значение будет от свойства flex-direction, а второе от flex-wrap.

1. Margin: auto используют, чтобы отцентрировать блочный элемент по горизонтали. При этом может быть два варианта: либо браузер выделяет под отступы с обеих сторон все свободное пространство, либо 0. Первый вариант – наиболее распространённый способ использования margin: auto.

И margin: auto не будет работать для строчных элементов и элементов с абсолютным позиционированием. Следовательно, будет выставлять значение 0. Также не будет работать свойство для стандартных блочных элементов, у которых не указана ширина. Если ширина элемента задана auto или 100%, то есть он занимает всю ширину контейнера и значение margin: auto будет 0.

#square {

margin: auto;

width: 150px;

height: 150px;

color: red:

} /\* красный квадрат со сторой 150px посередине блока\*/

Для вертикальных отступов margin: auto всегда будет 0.

Можно использовать margin-left: auto и margin-right: auto, чтобы прижать элемент к правой или левой границе блока.

1. При помощи свойства box-sizing можно изменить то, как браузер будет рассчитывать размеры элемента. Если задать box-sizing: border-box , то padding и borderбудут включены в размеры блока width и heidht. Это свойство работает и при способе позиционирования flexbox.
2. flex-grow определяет, может ли начальная ширина расти. Увеличение ширины элемента осуществляется за счет свободного пространства линии.

flex-shrink определяет, может ли начальная ширина элемента уменьшаться. Уменьшение будет только в том случае, если ширины линий будет недостаточно для отображения всех элементов, расположенных на ней.

1. Justify-content: space-between;
2. width 148,5; height 17,8
3. Свойство order нужно для изменения порядка следования flex-элементов. По умолчанию располагаются в том же порядке, что и в HTML-коде. В качестве значения может быть 0 или любое целое отрицательно или положительное число.
4. Файл table.html