1. Почему в большинстве ситуаций стоит использовать flexbox-позиционирование?

flexbox-позиционирование является самым простым и удобным вариантом в современном мире и именно оно является официально рекомендованным вариантом от World Wide Web Consortium (W3C)

1. Самостоятельно изучите способ позиционирования через `display: table` и ответьте на вопрос, для каких ситуаций оно лучше всего подходит?

display: table; лучше всего подходит для больших контейнеров. Например, боковая панель статической ширины рядом с "гибким" контейнером основного контента. Также таблицы можно использовать, когда вы хотите получить ряд одинаковых по высоте элементов. Тем не менее, неспособность хорошо работать с шириной, выраженной в процентах, представляет собой серьезное ограничение.

1. Какими способами можно сделать горизонтальное выравнивание по центру? Минимум 3 варианта, можно больше

1) Свойство text-align

Добавим класс center со свойством text-align: center к родительскому блоку и отобразим дочерний блок, как inline-block.

<div class="wrap center">

</div>

.center {

text-align: center;

}

.block {

...

display: inline-block;

}

2) Свойство margin: auto

Данное свойство мы прописываем дочернему блоку. В сокращенной записи верхнему и нижнему отступу мы можем оставить нулевое значение или указать нужные размеры согласно макету для верстки. Но нас интересует значение auto, как раз оно и задает автоматический отступ с левой и с правой стороны от вложенного блока. Это значит, что пространство с обеих сторон должно рассчитываться поровну

<div class="wrap">

<div class="block m-auto"></div>

</div>

.m-auto {

margin: 0 auto;

}

3) Свойство flexbox

Для родителя добавляются два флексовых свойства.

<div class="wrap flexbox">

<div class="block"></div>

</div>

.flexbox { display: flex;

justify-content: center;

}

1. Какие есть оси во флекс-верстке и как задается их направление?

В CSS Flexbox имеются две оси. Первая ось называется главной (по умолчанию она направлена слева направо). Вторая - поперечная (по умолчанию направлена сверху вниз), она всегда перпендикулярно главной. Главная ось задаёт основное направление flex-элементов во flex-контейнере, а поперечная ось определяет их направление при переносе на новую линию.

1. Разберитесь, как работает свойство `margin: auto` во флекс-верстке, приведите пример использования

значение auto получило премию журнала Форбс в номинации «Самое влиятельное значение CSS-свойства внутри флекс-контейнера».

Всё дело в механизме распределения свободного места. Если внутри флекс-контейнера есть свободное место, то оно перераспределяется так:

находятся элементы, у которых есть внешние отступы со значением auto;

всё свободное место в соответствующих направлениях отдаётся таким отступам (то есть задаётся определённый размер отступа в пикселях);

если элементов с автоматическими отступами на одном направлении несколько, то место между ними перераспределяется поровну;

только после этого запускаются механизмы выравнивания.

Поэтому margin: auto; влияет на положение флекс-элементов на обеих осях, а также «ломает» механизмы выравнивания, ведь выравнивание происходит, когда есть свободное место. Но если всё свободное место «перетекло» в автоматический отступ, то и выравнивать нечего.

.parent {

display: flex;

height: 400px;

background-color: #222;

}

.child {

background-color: red;

width: 100px;

height: 100px;

margin: auto;

}

1. В чем преимущества `box-sizing`?

box-sizing, которое позволяет нам точно менять, как блочная модель работает и как вычисляются размеры элемента.

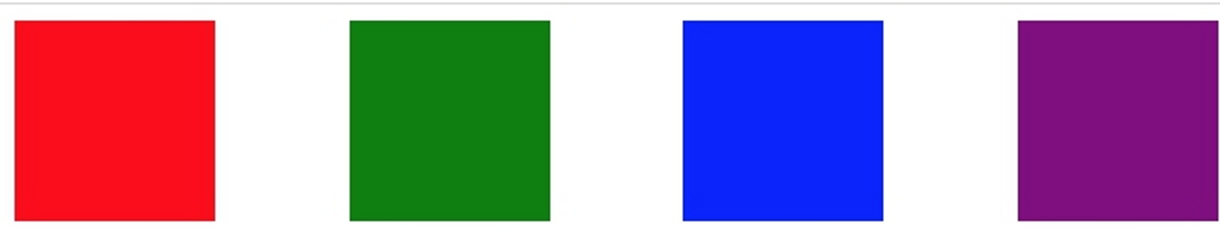
1. Чем отличается `flex-grow` от `flex-shrink`?

Это свойство указывает, может ли вырастать флекс-элемент при наличии свободного места, и насколько. По умолчанию значение равно 0. Значением может быть любое положительное целое число (включая 0).

Свойство flex-shrink полностью противоположно свойству flex-grow. Если в контейнере не хватает места для расположения всех элементов без изменения размеров, то свойство flex-shrink указывает, в каких пропорциях элемент будет уменьшаться. Чем больше значение у этого свойства, тем быстрее элемент будет сжиматься по сравнению с соседями, имеющими меньшее значение

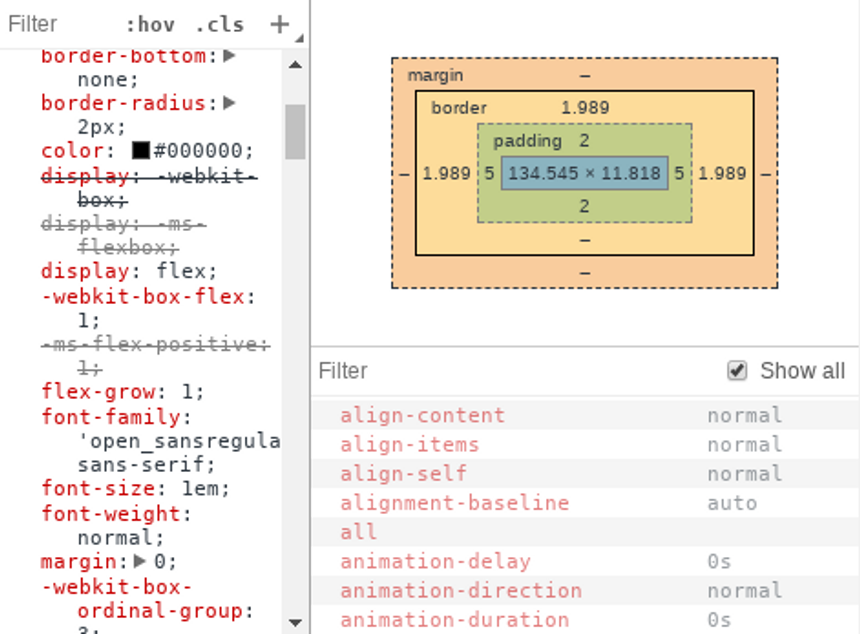
1. Как можно добиться следующего позиционирования элементов:

justify-content: space-between



1. Какой в итоге будет размер у элемента (можно округлить)?

141.53 x 15.81



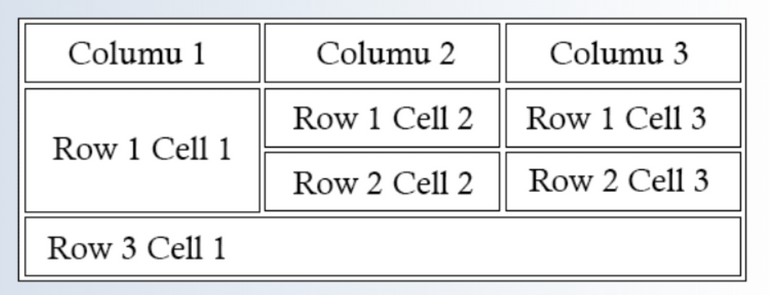
10. Самостоятельно разберитесь, зачем нужно свойство `order`?

При помощи свойства order можно менять порядок отображения флекс-элементов внутри флекс-контейнера.

По умолчанию элементы отображаются в том порядке, в котором они расположены в разметке, а значение свойства order равно 0.

Значение задаётся в виде целого отрицательного или положительного числа. Элементы встают по возрастающей.

11. Каким кодом можно сделать такую таблицу?



<body>

<table border="1" style="border-collapse:collapse;">

<tr>

<td>Columnu 1</th>

<td>Columnu 2</th>

<td>Columnu 3</th>

</tr>

<tr>

<td rowspan=”2”>Row1 Cell 1</td>

<td>Row1 Cell 2</td>

<td>Row1 Cell 2</td>

<td>Row2 Cell 3</td>

<td>Row2 Cell 3</td>

</tr>

<tr>

<td colspan “2”> Row3 Cell 1</td>

</tr>

</table> </body>

12. Изучите материалы и найдите, для каких ситуаций подходит `position: fixed`?

Элемент с position: fixed; фиксируется в рамках области просмотра страницы, будь то в окне браузера или в iframe, расстояние до края которой указано в свойствах top, right, bottom, left. После того, как свойство position примет значение fixed, размер элемента может измениться, а соседние элементы сдвинутся на освободившееся пространство. «Приклеить» элемент, чтобы он не менял своего положения при прокрутке страницы. Элемент с position: fixed; не влияет на размер родителя.

13. Каким способом лучше всего верстать большие блоки текста?

Лучше комбинировать способы.

14. Как рассчитывается размер flex-контейнера?

Размеры флекс-элементов рассчитываются как в обычной блочной модели: рамки +

отступы + размеры содержания