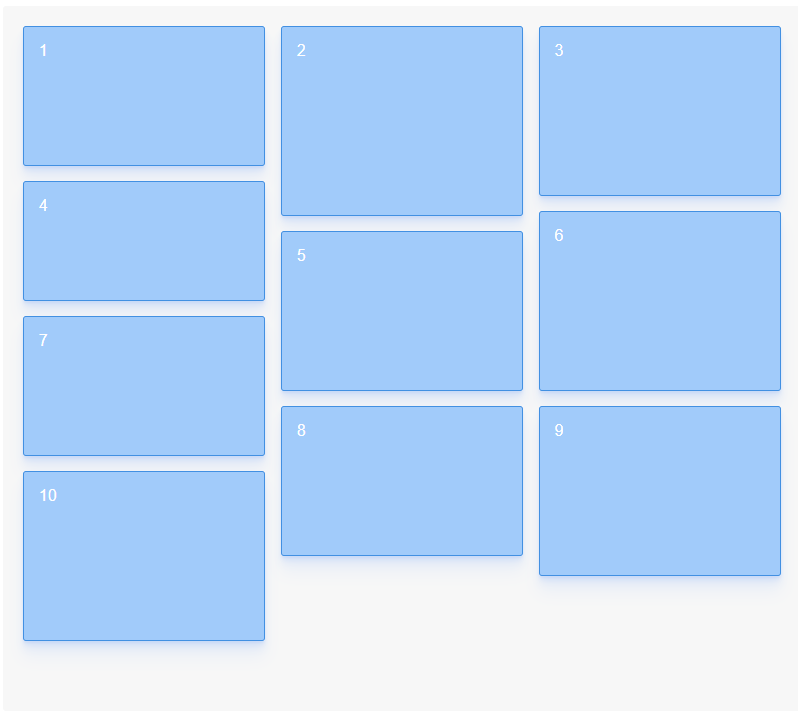
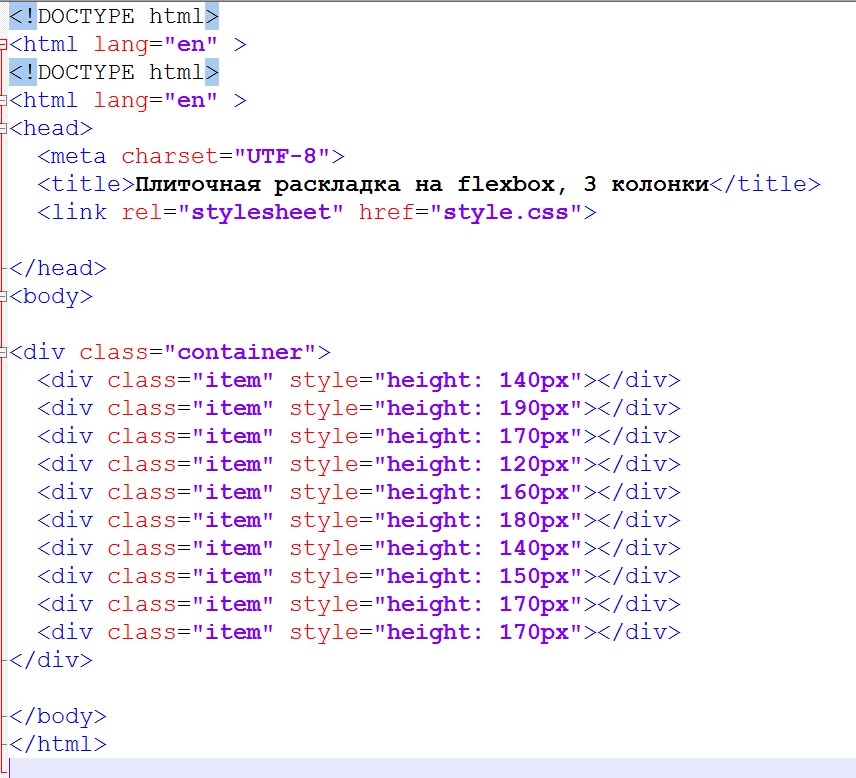
«Плиточная» раскладка блоков c помощью flexbox

1. **Задание**

**Создать следующий макет страницы страницы с помощью flexbox**

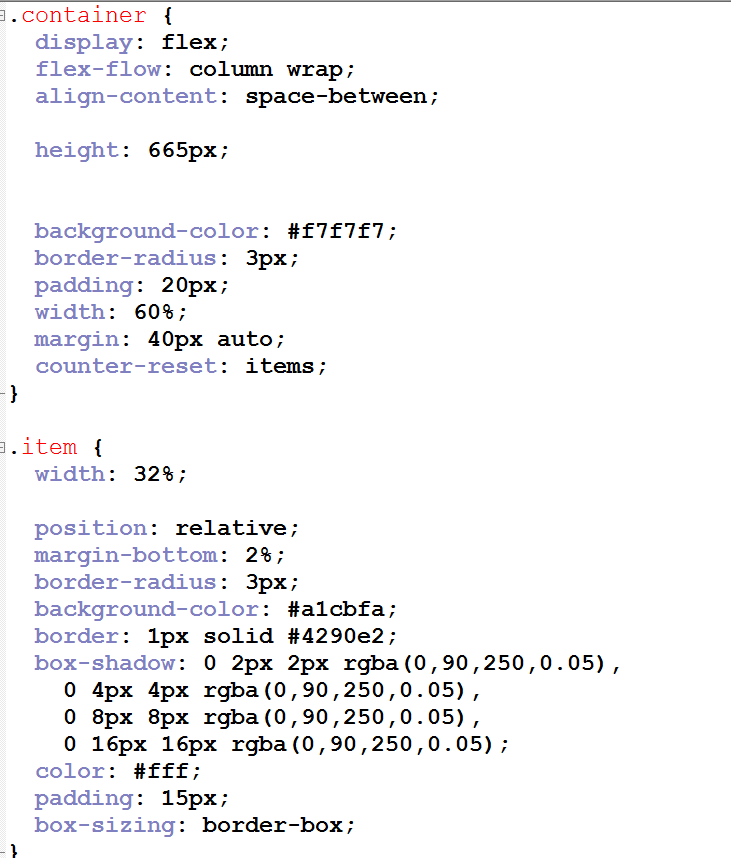


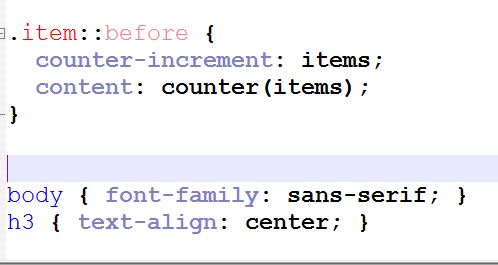
НТML- страница:



С первого взгляда это кажется довольно простым, взять и создать плиточную раскладку на флексах. Всё, что нужно, это всего лишь выставить flex-flow на column wrap

Стиль:





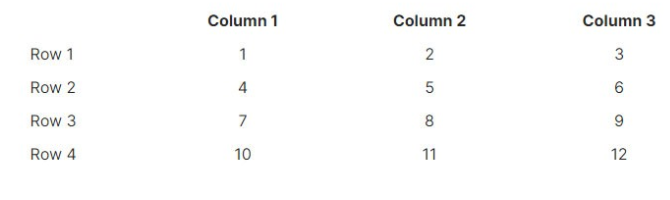
Проблема состоит в том, что нужно получить  элементы как колонки, но имеющие строчный порядок.

**Меняем порядок элементов с помощью order и nth-child()**

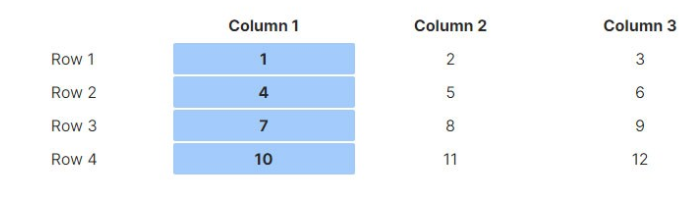
Свойство order влияет на порядок элементов в CSS Flexbox или гриде, и мы можем смело использовать его для смены порядка элементов в нашей плиточной раскладке. Свойство order довольно простое в использовании: если у вас два элемента и у одного стоит order: 1, а у другого order: 2, то элемент с order: 1 будет показан перед другим элементом, вне зависимости от их порядка в HTML исходнике.

В нашем случае, решение зависит от тонкостей спецификации order: что случится, если два или более элементов будут с одинаковым order? Какой из них будет первым? В этом случае, flexbox опирается на исходный код: тот элемент, который идет первым в исходном HTML коде, будет показан перед другим элементом с таким же значением order. Это даёт нам возможность легко перегруппировать элементы в сетке таким образом, что мы сможем сменить порядок расстановки элементов с колоночного на строчный, всё ещё показывая эти строки, как колонки, используя   
nth-child().

Посмотрите на табличку ниже. Чтобы получить рациональный порядок с использованием flex-direction: row, нам просто надо показывать элементы в стандартном порядке: 1, 2, 3, 4, 5, 6, и т.д.



Но если нам надо получить тот же порядок с использованием flex-direction: column, то тогда уже нам понадобится поменять сам порядок элементов, таким образом, чтобы он соответствовал порядку каждой колонки в таблице (а не каждой строки):



То есть первыми элементами в нашей флексбокс раскладке должны 1, 4, 7, 10. Эти элементы заполнят первую колонку, далее 2, 5, 8, 11 для второй колонки и 3, 6, 9, 12 для третьей и последней колонки.

Тут нам на помощь приходит селектор nth-child(). Мы можем его использовать для того, чтобы выделить каждый третий элемент (3n), начиная с первого элемента (3n + 1) и выставить всем этим элементам одинаковое значение order:

.item:nth-child(3n+1) { order: 1; }

Этот селектор выставит order: 1 для элементов 1, 4, 7, 10, то есть всем первым колонкам. Другими словами, мы используем комбинацию nth-child() и order, чтобы изменить порядок элементов на их изначальном положении. Чтобы создать 2 и 3 колонки, мы изменим порядок у других элементов:

**.item:nth-child(**3n**+**1**)** { **order**: 1; }  
**.item:nth-child(**3n**+**2**)** { **order**: 2; }  
**.item:nth-child(**3n**)** { **order**: 3; }

Тут мы делаем три группы элементов: 1, 4, 7, 10 (3n+1) c order: 1, далее 2, 5, 8, 11 (3n+2) с order: 2 и 3, 6, 9, 12 (3). Всё вместе выдаёт нам такой порядок элементов 1, 4, 7, 10, 2, 5, 8, 11, 3, 6, 9, 12.

Такой подход создаст иллюзию того, что элементы вернулись к своему изначальному порядку, как если бы вы рассматривали их в порядке слева направо. Если бы мы визуально оценивали эту сетку по строкам, то первая строка включала бы по первому элементу с каждой группы (1, 2, 3), вторая строка содержала бы каждый второй элемент из каждой группы (4, 5, 6) и так далее и т.п. Используя эту технику мы можем создать плиточную раскладку с элементами, показанными как колонки, но имеющими порядок как было бы в строчном варианте.

Этот селектор выставит order: 1 для элементов 1, 4, 7, 10, то есть всем первым колонкам. Другими словами, мы используем комбинацию nth-child() и order, чтобы изменить порядок элементов на их изначальном положении. Чтобы создать 2 и 3 колонки, мы изменим порядок у других элементов:

**.item:nth-child(**3n**+**1**)** { **order**: 1; }  
**.item:nth-child(**3n**+**2**)** { **order**: 2; }  
**.item:nth-child(**3n**)** { **order**: 3; }

Тут мы делаем три группы элементов: 1, 4, 7, 10 (3n+1) c order: 1, далее 2, 5, 8, 11 (3n+2) с order: 2 и 3, 6, 9, 12 (3). Всё вместе выдаёт нам такой порядок элементов 1, 4, 7, 10, 2, 5, 8, 11, 3, 6, 9, 12.

Такой подход создаст иллюзию того, что элементы вернулись к своему изначальному порядку, как если бы вы рассматривали их в порядке слева направо. Если бы мы визуально оценивали эту сетку по строкам, то первая строка включала бы по первому элементу с каждой группы (1, 2, 3), вторая строка содержала бы каждый второй элемент из каждой группы (4, 5, 6) и так далее и т.п. Используя эту технику мы можем создать плиточную раскладку с элементами, показанными как колонки, но имеющими порядок как было бы в строчном варианте.

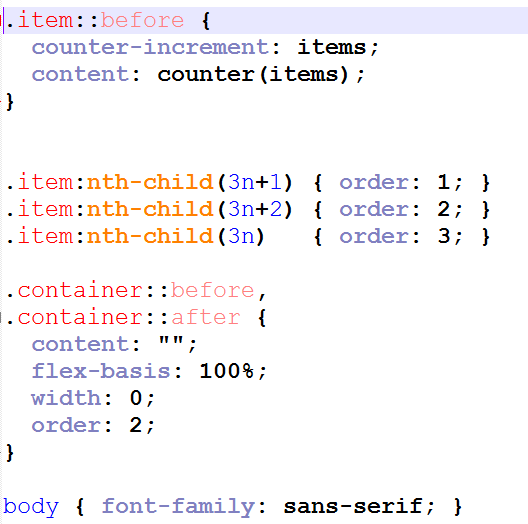
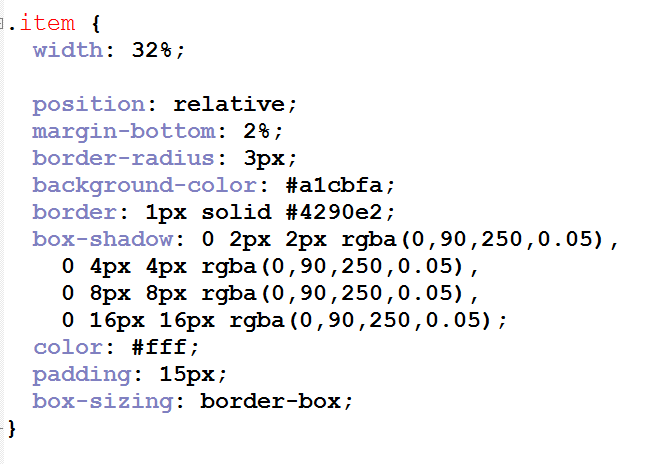
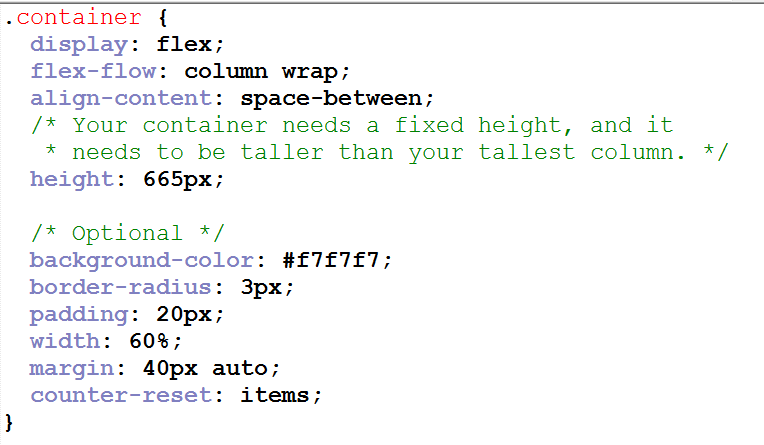
# Предотвращаем слияние колонок

Если у вас довольно много элементов в раскладке, то этот подход 100% приведёт к тому, что что-то пойдет не так. Мы делаем расчёт на то, что каждая “группа”, которую мы создали, будет показана ровно как одна колонка в контейнере, но в реальности у элементов могут быть разные высоты и колонки с лихвой могут сливаться друг с другом.

Вообще, с флексбоксами нет простого способа для указания “этот элемент должен перейти на новую строку”, но мы можем достигнуть этого эффекта, добавив невидимые элементы, которые возьмут 100% высоты контейнера. Так как они требуют 100% высоты родителя, то они не поместятся в колонку вместе с любым другим элементом, создавая тем самым, переходы на новые строки.

Нам нужно вставить такие переходы в сетку и массив элементов, таким образом, чтобы у нас получилась такая последовательность: 1, 4, 7, 10,<переход>, 2, 5, 8, 11,<переход>, 3, 6, 9, 12. Мы можем использовать псевдоэлементы на контейнере, чтобы добавить такие переходы и мы можем выставить order на 2 обоим. Добавление псевдоэлемента :before, сделает его первым потомком контейнера, а добавление псевдоэлемента :after сделает его последним потомком того же контейнера, так что если мы выставим order: 2 каждому из них, то они станут первым и последним элементом “группы” с order: 2 (так как они находятся до и после других элементов): :before, 2, 5, 8, 11, :after.

Итого получаем в стилях:

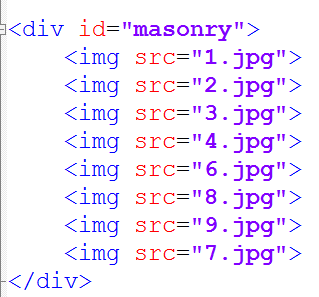


1. **С помощью flexbox построить страницу по макету**

Макет



НТМL- шаблон



Для стиля: при наведении мыши

