STEP IT ACADEMY

Занятие №7

Flexible Box Layout Module

Flexbox

Это новая технология, которая уже имеет достаточно широкую поддержку браузеров. Flexbox предоставляет инструменты для быстрого создания сложных, гибких макетов, и функции, которые были сложны в традиционных методах CSS.

Для начала нам нужно выбрать, какие элементы следует выкладывать в виде flex блоков. Для этого мы устанавливаем специальное значение display

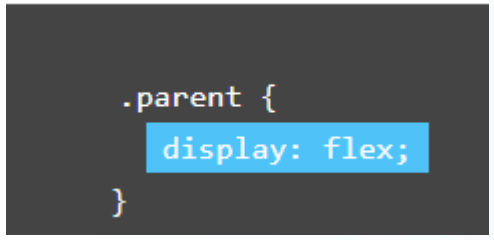
Display

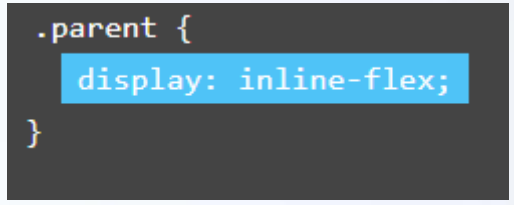
Многоцелевое свойство, которое определяет, как элемент должен быть показан в документе.

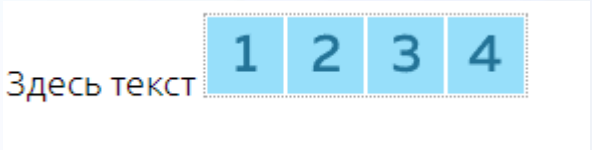
Применяется ко всем элементам.

flex - Элемент ведёт себя как блочный и выкладывает содержимое согласно флекс-модели.

inline-flex - Элемент ведёт себя как строчный и выкладывает содержимое согласно флекс-модели.

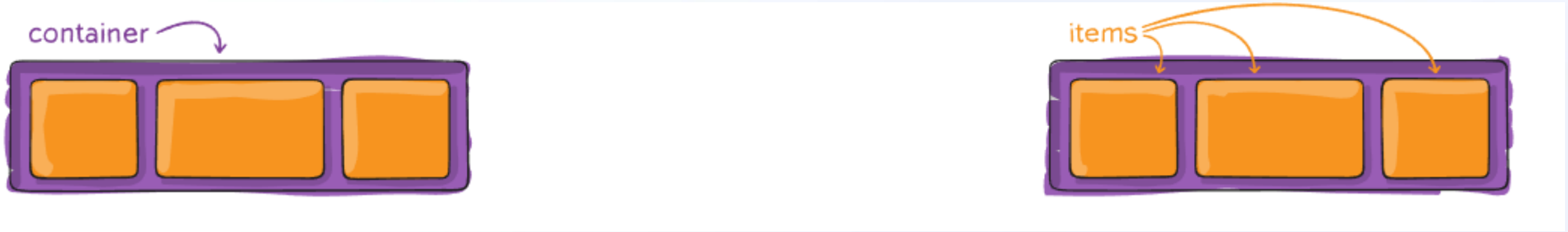






Свойства Flexbox

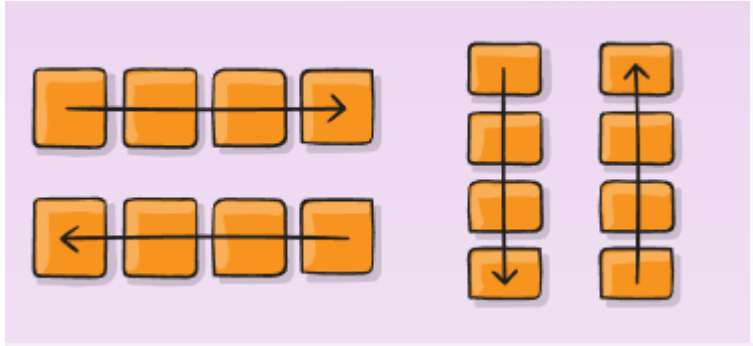
При работе с Гибкими контейнерами, всегда должен быть родительский контейнер, которому и задается основные свойства. Родительский контейнер внутри себя содержит элементы(items), которые и выравниваются.



Давайте разберем, какие есть способы работы с гибкими контейнерами.

flex-direction

Это устанавливает главную ось, тем самым определяя направление размещения гибких элементов в контейнере flex. Flexbox — это (помимо опциональной обертки) концепция макета с одним направлением. Думайте о гибких элементах как о расположении в первую очередь либо в горизонтальных рядах, либо в вертикальных столбцах.



```css
.container {
  flex-direction: row | row-reverse | column | column-revers
}
```

Row - Значение по умолчанию, слева направо.Flex-элементывыкладываются в строку

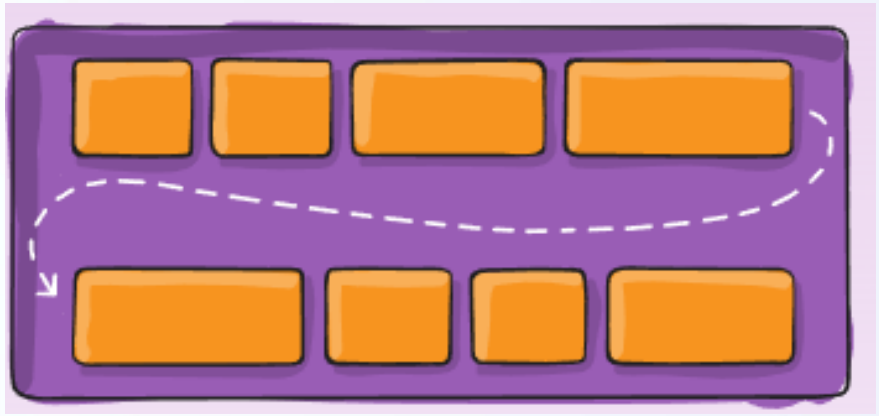
row-reverse - Направление справа налево. Flex-элементы выкладываются встроку относительноправого края контейнера.

Column - Направление сверху вниз. Flex-элементы выкладываются вколонку.

column-reverse - Колонка с элементами в обратном порядке, снизу вверх

flex-wrap

По умолчанию все flex-элементы будут пытаться разместиться на одной строке. Вы можете изменить это и разрешить элементам переноситься по мере необходимости с помощью этого свойства.



nowrap (по умолчанию): все гибкие элементы будут в одной строке

wrap: flex-элементы будут переноситься на несколько строк сверху вниз.

wrap-reverse: flex-элементы будут переноситься на несколько строк снизу вверх.

flex-flow

Это сокращение для свойств flex-direction и flex-wrap, которые вместе определяют главную и поперечную оси flex-контейнера. Значение по умолчанию — row nowrap.

1 значение – flex-direction

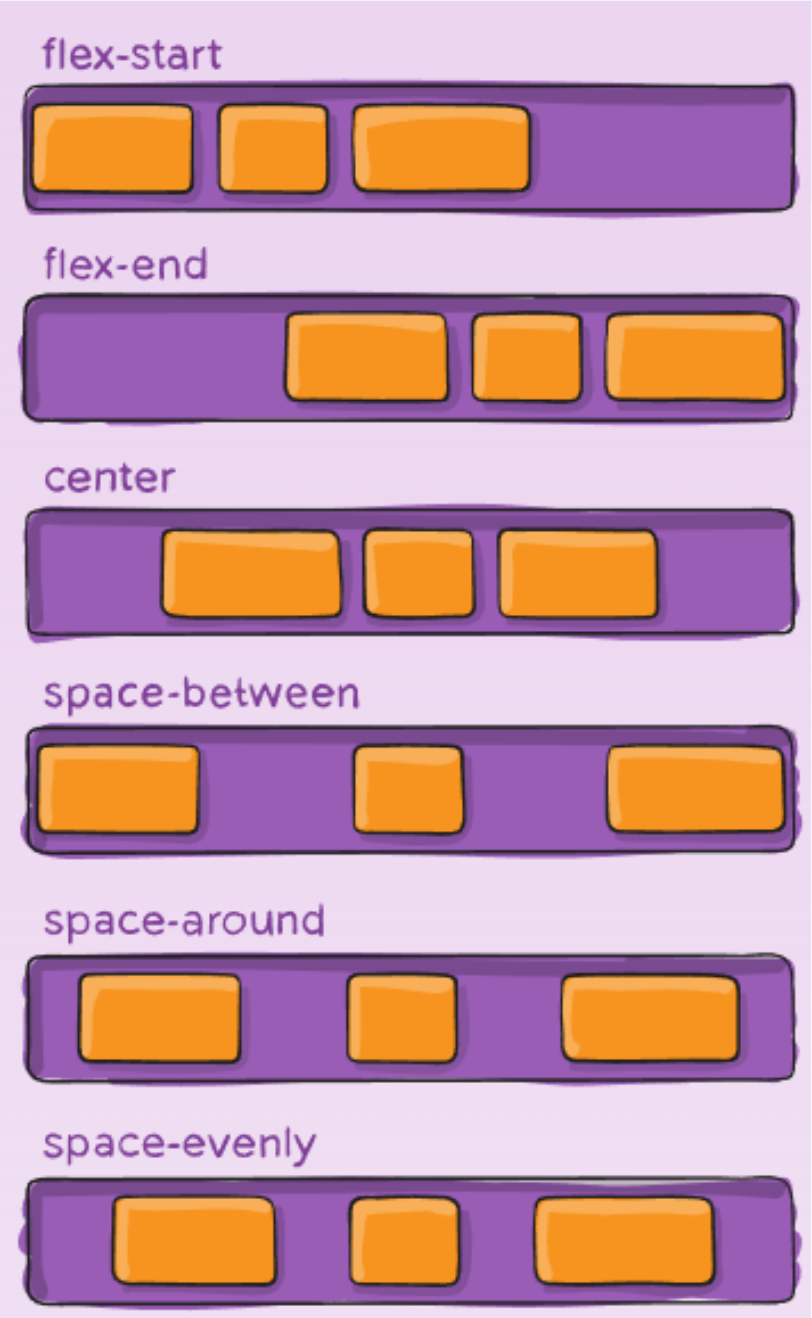
2 значение – flex wrap

```
.container {
  flex-flow: column wrap;
}
```

justify-content

Это определяет выравнивание вдоль главной оси. Это помогает распределять дополнительное свободное пространство, когда либо все гибкие элементы в строке негибкие, либо гибкие, но достигли своего максимального размера. Он также осуществляет некоторый контроль над выравниванием элементов, когда они выходят за пределы строки.

```
.container {
  justify-content:
}
```



flex-start (по умолчанию): элементы упаковываются в направлении начала flex-направления

flex-end: элементы упаковываются ближе к концу flex-направления

center: элементы располагаются по центру строки

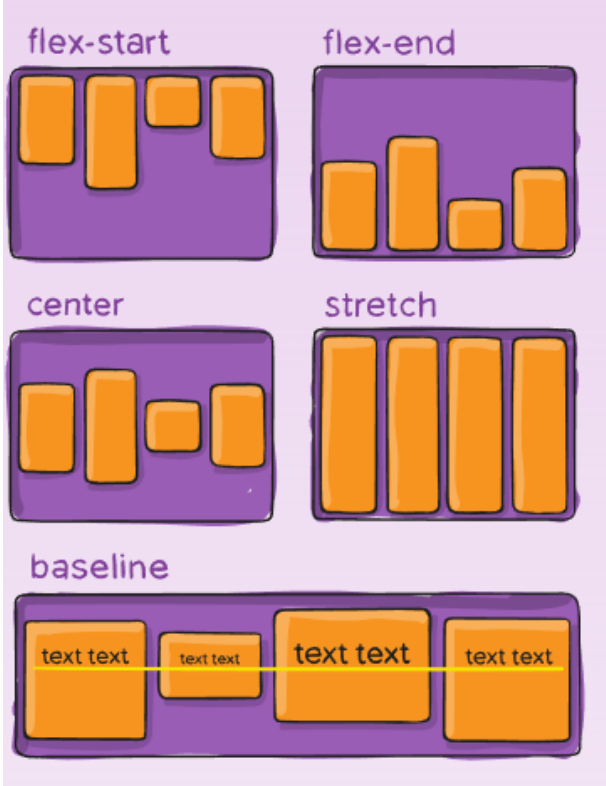
space-between: предметы равномерно распределены в очереди; первый элемент находится в начальной строке, последний элемент в конечной строке

space-around: элементы равномерно распределены в строке с равным пространством вокруг них.

space-evenly: элементы распределяются так, чтобы расстояние между любыми двумя элементами (и расстояние до краев) было одинаковым.

align-items

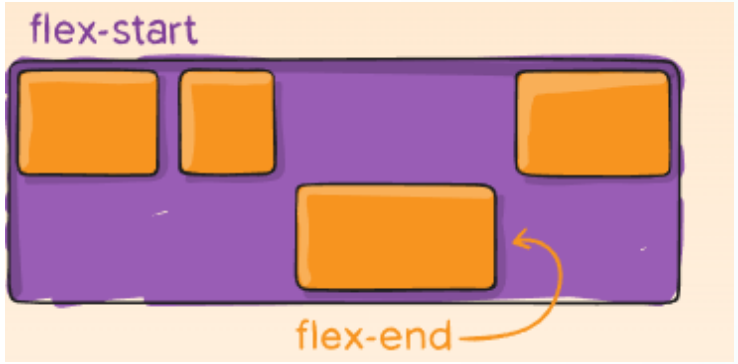
Это определяет поведение по умолчанию для того, как flex-элементы располагаются вдоль поперечной оси на текущей строке. Думайте об этом как о версии с выравниванием содержимого для поперечной оси (перпендикулярной главной оси).



flex-start - флексы выравниваются в начале поперечной оси контейнера.
flex-end - флексы выравниваются в конце поперечной оси контейнера.
center - флексы выравниваются по линии поперечной оси.
baseline - флексы выравниваются по их базовой линии.
Stretch - флексы растягиваются таким образом, чтобы занять всё доступное пространство контейнера.

align-self

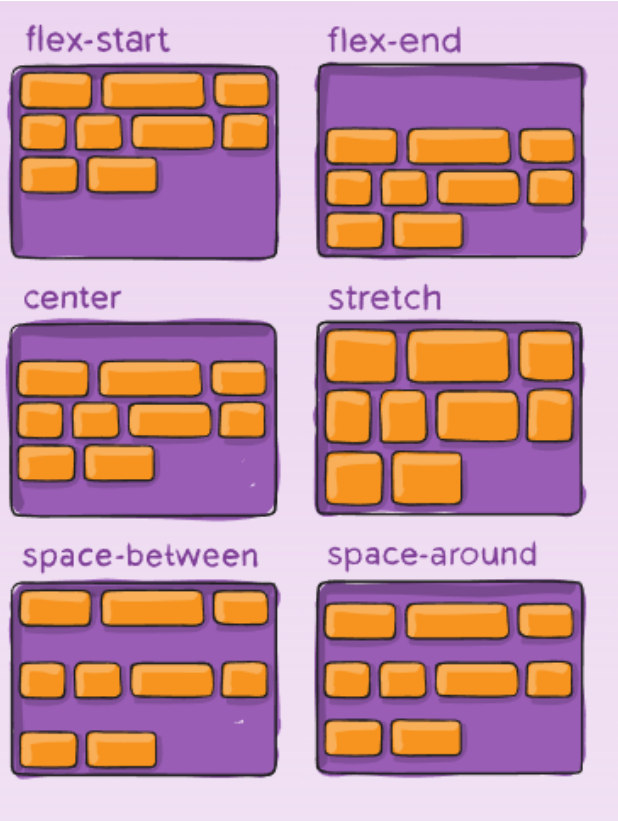
Свойство align-self выравнивает флекс-элементы текущей строки, переписывая значение align-items.



flex-start Элемент выравнивается в начале поперечной оси контейнера.
flex-end Элемент выравнивается в конце поперечной оси контейнера.
center Элемент выравнивается по центральной линии на поперечной оси.
baseline Элемент выравнивается по базовой линии текста.
Stretch Элемент растягивается таким образом, чтобы занять всё свободное пространство контейнера по поперечной оси.

align-content

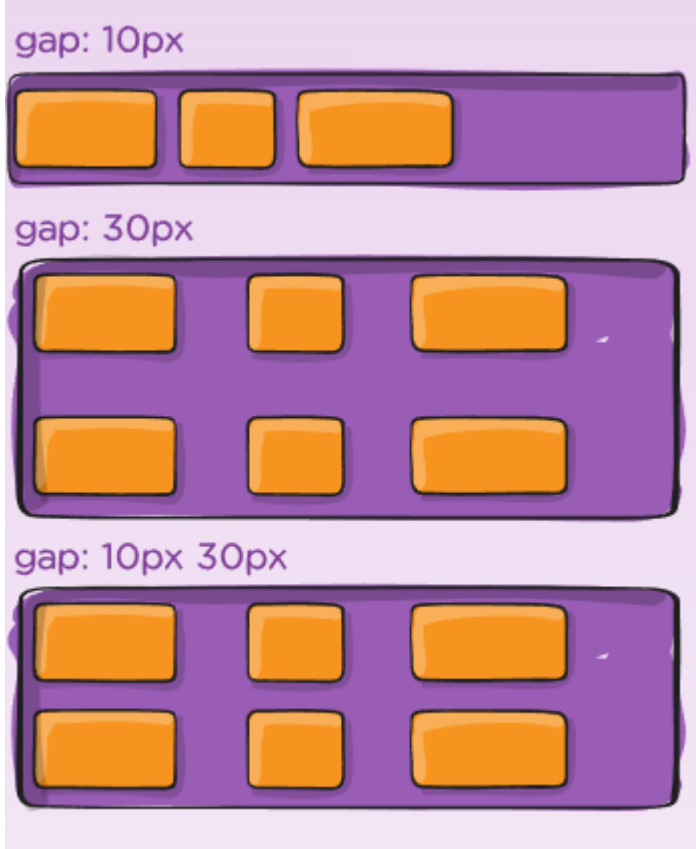
Это выравнивает строки гибкого контейнера внутри, когда на поперечной оси есть дополнительное пространство, аналогично тому, как justify-content выравнивает отдельные элементы внутри



flex-start Строки располагаются в начале поперечной оси. Каждая следующая строка идёт вровень с предыдущей.
flex-end Строки располагаются начиная с конца поперечной оси. Каждая предыдущая строка идёт вровень со следующей.
center Строки располагаются по центру контейнера.
space-between Строки равномерно распределяются в контейнере и расстояние между ними одинаково.
space-around Строки равномерно распределяются таким образом, чтобы пространство между двумя соседними строками было одинаковым. Пустое пространство перед первой строкой и после последней строки равно половине пространства между двумя соседними строками.
stretch Строки равномерно растягиваются, заполняя свободное пространство.

gap, row-gap, column-gap

Свойство gap управляет пространством между flex-элементами. Это расстояние применяется только между элементами, а не на внешних краях.



Поведение можно рассматривать как минимальный отступ, как если бы отступ каким-то образом стал больше (из-за чего-то вроде justify-content: space-between;), тогда **gap** вступит в силу только в том случае, если это пространство в конечном итоге станет меньше.

```
gap: 10px;

gap: 10px 20px; /* row-gap column gap */

row-gap: 10px;
```

Давайте подведем итоги урока.