#4 - Flexbox

**Flexbox** — это новый способ располагать блоки на странице.  
Это технология, созданная именно для раскладки элементов, в отличие от float-ов.  
С помощью **Flexbox** можно легко выравнивать элементы по горизонтали и по вертикали, менять направление и порядок отображение элементов, растягивать блоки на всю высоту родителя или прибивать их к нижнему краю.

Для начала надо знать, что flex-элементы располагаются по осям. По умолчанию элементы располагаются по горизонтали — вдоль **main axis** — главной оси.

Chart

Description automatically generated

Приготовим полигон для экспериментов (html, css):

<div class="flex-container">

    <div class="flex-item"> Curabitur ac vestibulum mi </div>

    <div class="flex-item"> In viverra dapibus   </div>

    <div class="flex-item"> Fusce tincidunt diam et </div>

    <div class="flex-item"> Nulla in dui vel est  </div>

    <div class="flex-item"> at diam in lobortis  </div>

  </div>

.flex-container {

  padding: 10px;

  background: gold;

  }

.flex-item {

  margin: 10px;

  padding: 5px;

  background: tomato;

  border: 1px solid #FFF;

  }

У нас в браузере получается вот такая картинка: один родительский блок (желтый) и 5 дочерних.

У нас в браузере получается вот такая картинка: один родительский блок (желтый) и 5 дочерних.

Background pattern

Description automatically generated

**Display: flex**

И теперь родительскому элементу добавляем display: flex;.  
Внутренние div-ы выстраиваются в ряд (вдоль главной оси) колонками одинаковой высоты, независимо от содержимого.

Graphical user interface

Description automatically generated with medium confidence

display: flex; делает все дочерние элементы резиновыми — flex, а не блочными — block, как было изначально.

Если родительский блок содержит картинки или текст без оберток, они становятся анонимными flex-элементами.

Свойство display для **Flexbox** может принимать два значения:

flex — при рассчете ширины блоков приоритет у раскладки (при недостаточной ширине блоков контент может вылезать за границы);

Достаточно места для всех:

A picture containing graphical user interface

Description automatically generated

Ширина ограничена:

Diagram

Description automatically generated with low confidence

inline-flex — приоритет у содержимого (контент растопыривает блоки до необходимой ширины, чтобы строчки, по возможности, поместились). Кроме того, другие строчные элементы могут встать на одну строку с inline-flex родителем.

Text

Description automatically generated

**Flex-direction**

Направление раскладки блоков управляется свойством flex-direction.

Возможные значения:

row — строка (значение по умолчанию);

A picture containing graphical user interface

Description automatically generated

row-reverse — строка с элементами в обратном порядке;

Graphical user interface

Description automatically generated

column — колонка;  
column-reverse — колонка с элементами в обратном порядке.

Graphical user interface

Description automatically generated with medium confidence

**Flex-wrap**

В одной строке может быть много блоков. Переносятся они или нет определяет свойство flex-wrap.

Возможные значения:

nowrap — блоки не переносятся (значение по умолчанию);  
wrap — блоки переносятся;  
wrap-reverse — блоки переносятся и располагаются в обратном порядке.

nowrap, wrap, wrap-reverse

Graphical user interface, website

Description automatically generated

Для короткой записи свойств flex-direction и flex-wrap существует свойство: flex-flow.

Возможные значения: можно задавать оба свойства или только какое-то одно. Например:

flex-flow: column;  
flex-flow: wrap-reverse;  
flex-flow: column-reverse wrap;

row-reverse wrap-reverse

**Justify-content**

Для выравнивания элементов есть несколько свойств: justify-content, align-items и align-self. justify-content и align-items применяются к родительскому контейнеру, align-self — к дочерним. justify-content отвечает за выравнивание по главной оси.

Возможные значения justify-content:

flex-start — элементы выравниваются от начала главной оси (значение по умолчанию);  
flex-end — элементы выравниваются от конца главной оси;  
center — элементы выравниваются по центру главной оси;  
space-between — элементы выравниваются по главной оси, распределяя свободное место между собой;  
space-around — элементы выравниваются по главной оси, распределяя свободное место вокруг себя.

Graphical user interface, application, website

Description automatically generated

A picture containing graphical user interface

Description automatically generated

A picture containing graphical user interface

Description automatically generated

**Align-items**

align-items отвечает за выравнивание по перпендикулярной оси.

Возможные значения align-items:

flex-start — элементы выравниваются от начала перпендикулярной оси;  
flex-end — элементы выравниваются от конца перпендикулярной оси;  
center — элементы выравниваются по центру;  
baseline — элементы выравниваются по базовой линии;  
stretch — элементы растягиваются, занимая все пространство по перпендикулярной оси (значение по умолчанию).

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

**Align-self**

align-self также отвечает за выравнивание по перпендикулярной оси, но задается отдельным flex-элементам.

Возможные значения align-self:

auto — значение по умолчанию. Означает, что элемент использует align-items родительского элемента;  
flex-start — элемент выравнивается от начала перпендикулярной оси;  
flex-end — элемент выравнивается от конца перпендикулярной оси;  
center — элемент выравнивается по центру;  
baseline — элемент выравнивается по базовой линии;  
stretch — элемент растягивается, занимая все пространство по высоте.

**Align-content**

Для управления выравниванием внутри многострочного flex-контейнера есть свойство align-content.

Возможные значения:

flex-start — элементы выравниваются от начала главной оси;  
flex-end — элементы выравниваются от конца главной оси;  
center — элементы выравниваются по центру главной оси;  
space-between — элементы выравниваются по главной оси, распределяя свободное место между собой;  
space-around — элементы выравниваются по главной оси, распределяя свободное место вокруг себя;  
stretch — элементы растягиваются, заполняя всю высоту (значение по умолчанию).

**Flex-grow**  
Определяет, сколько как распределяется между флекс0детьми свободное пространство родителя. В качестве значения принимаются числа, они задают пропорции.   
  
К примеру, если у родителя есть 300 пикселей свободного пространства и 5 флекс-детей со значением flex-grow: 1, то все дети получат по равной доле ⅕ от 300 пикселей.   
300px \* ⅕ = 60px получит каждый элемент, “попросивший” 1 часть  
  
Если какой-то элемент получил значение 2, то он получит вдвое больше.  
300px \* ⅙ = 50px получит каждый элемент, “попросивший” 1 часть  
300px \* 2/6 = 100px получит элемент, “попросивший” 2 части

Для того, чтобы лучше освоить флексы, советую вам вот эти полезные ресурсы:

Шпаргалка по флексам: <http://yoksel.github.io/flex-cheatsheet/>

FlexBox Froggy, игра для взрослых верстальщиков: <https://flexboxfroggy.com/#ru>   
  
flex-grow странный. Так ли это? <https://css-live.ru/articles/flex-grow-strannyj-tak-li-eto.html>