

Flexbox

Цели занятия

- подробно разобрать что такое Flexbox;
- закрепить на практике работу с Flex при помощи [игры](#);

Полезные материалы

- [Справочник CSS](#)
- [Правила оформления CSS-кода](#)
- [Онлайн-игра по Flexbox](#)

Flexbox

Модуль Flexbox Layout (Flexible Box) направлен на обеспечение более эффективного способа размещения, выравнивания и распределения пространства между элементами в контейнере, даже если их размер неизвестен и / или динамичен (Flex значит «гибкий»).

Основная идея flex layout состоит в том, чтобы дать контейнеру возможность изменять ширину / высоту его элементов (и порядок), чтобы наилучшим образом заполнить доступное пространство (главным образом, для отображения на всех типах устройств с любым размером экрана). Flex контейнер расширяет элементы, чтобы заполнить доступное свободное пространство, или сжимает их, чтобы предотвратить переполнение.

Свойства для контейнера

display — определяет flex контейнер; inline или block в зависимости от заданного значения. Включает flex контекст для всех потомков первого уровня.

```
.container {  
  display: flex;  
}
```

flex-direction — устанавливает основную ось, таким образом определяя направление flex элементов, помещаемых в flex контейнер. Flexbox — это (помимо дополнительной упаковки) концепция однонаправленного макета. Думайте о flex элементах, как о первичных раскладках в горизонтальных рядах или вертикальных столбцах.

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

- **row** (по умолчанию): слева направо;
- **row-reverse** справа налево;
- **column**: так же, как и row но сверху вниз
- **column-reverse**: то же самое, row-reverse но снизу вверх

flex-wrap — по умолчанию гибкие элементы будут пытаться уместиться на одной строке. Вы можете изменить это и позволить элементам переходить на новую строку по мере необходимости с помощью этого свойства.

```
.container {  
  flex-wrap: nowrap | wrap | wrap-reverse;  
}
```

- **nowrap** (по умолчанию): все flex элементы будут в одной строке
- **wrap**: flex-элементы будут перенесены на несколько строк сверху вниз.
- **wrap-reverse**: flex-элементы будут перенесены на несколько строк снизу вверх.

justify-content — это свойство определяет выравнивание вдоль главной оси. Оно помогает распределить дополнительный остаток свободного пространства, когда-либо все flex элементы в строке негибкие, либо гибкие, но достигли своего максимального размера. Это также обеспечивает некоторый контроль над выравниванием элементов, когда они переполняют линию.

```
.container {  
  justify-content: flex-start | flex-end | center |  
  space-between | space-around | space-evenly;  
}
```

- **flex-start** (по умолчанию): элементы сдвинуты в начало flex-direction направления.
- **flex-end**: элементы сдвинуты ближе к концу flex направления.
- **center**: элементы центрированы вдоль линии
- **space-between**: элементы равномерно распределены по линии; первый элемент находится в начале строки, последний элемент в конце строки
- **space-around**: элементы равномерно распределены по линии с одинаковым пространством вокруг них. Обратите внимание,

что визуально пространства не равны, так как все элементы имеют одинаковое пространство с обеих сторон. Первый элемент будет иметь одну единицу пространства напротив края контейнера, но две единицы пространства между следующим элементом, потому что у следующего элемента есть свой собственный интервал, который применяется.

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

align-items — это свойство определяет поведение по умолчанию того, как flex элементы располагаются вдоль поперечной оси на текущей линии. Думайте об этом как о justify-content версии для поперечной оси (перпендикулярной главной оси).

```
.container {  
  align-items: stretch | flex-start | flex-end | center |  
  baseline;  
}
```

- **stretch** (по умолчанию): растягивать, чтобы заполнить контейнер (все еще соблюдаются min-width / max-width)
- **flex-start**: элементы размещаются в начале поперечной оси.
- **flex-end**: элементы располагаются в конце поперечной оси.
- **center**: элементы центрированы по поперечной оси
- **baseline**: элементы выровнены, по их базовой линии

Свойства для первых дочерних элементов(flex элементы)

order — по умолчанию flex элементы располагаются в исходном порядке. Однако свойство order управляет порядком их появления в контейнере flex.

```
.item {  
  order: 3; /* default is 0 */  
}
```

flex-grow — это свойство определяет способность flex элемента растягиваться в случае необходимости. Оно принимает значение от нуля, которое служит пропорцией. Это свойство, какое количество доступного пространства внутри гибкого контейнера должен занимать элемент.

Если для всех элементов flex-grow установлено значение 1, оставшееся пространство в контейнере будет равномерно распределено между всеми дочерними элементами. Если один из дочерних элементов имеет значение 2, этот элемент займет в два раза больше места, чем остальные (или попытается, по крайней мере).

```
.item {  
  flex-grow: 1; /* default 0 */  
}
```

flex-shrink — это свойство определяет способность гибкого элемента сжиматься при необходимости.

```
.item {  
  flex-shrink: 0; /* default 1 */  
}
```