

Урок 2.10. Flex, Grid

Материалы

- [Статья по Flex на css-tricks \(англ\)](#).
- [Статья по Grid на css-tricks \(англ\)](#).

Summary

Общее

- Рекомендую к прочтению две статьи из материалов: они интересные и с картинками. Будут прекрасным дополнением и продолжением урока.
- Как во Flex, так и в Grid участвуют два типа элементов: контейнер и его дочерние элементы первого уровня.

Flex

- Для включения flex в контейнере ему необходимо задать `display: flex`
- Выравнивание элементов по ширине:

```
.container {
  display: flex;

  /* По левому краю */
  justify-content: flex-start;

  /* По правому краю */
  justify-content: flex-end;

  /* По центру */
  justify-content: center;

  /* Всё свободное расстояние между элементами заполняется равномерно, расталкивая элементы по краям */
  justify-content: space-between;

  /* Справа и слева от каждого элемента добавляется расстояние (равномерно вычисленное равными частями от всего свободного места) */
  justify-content: space-around;
```

```
    /* Одинаковое расстояние (также как и space-between, но добавляются расстояния справа и слева контейнера) */  
    justify-content: space-evenly;  
}
```

- Выравнивание элементов по вертикали:

```
.container {  
    /* По верхнему краю */  
    align-items: flex-start;  
  
    /* По нижнему краю */  
    align-items: flex-end;  
  
    /* По центру */  
    align-items: center;  
  
    /* Растянуть */  
    align-items: stretch;  
}
```

- Направление флекса

```
.container {  
    /* В строку слева направо (по умолчанию) */  
    flex-direction: row;  
  
    /* В колонку сверху вниз */  
    flex-direction: column;  
  
    /* В строку справа налево */  
    flex-direction: row-reverse;  
  
    /* В колонку снизу вверх */  
    flex-direction: column-reverse;  
}
```

- Размеры у элемента

```
.flex-item {  
    /* Первое значение - flex-grow. */  
    /* Сколько места будет занимать элемент, если места больше, чем нужно */  
  
    /* Второе значение - flex-shrink */
```

```

/* Сколько места будет занимать элемент, если места меньше, чем нужно */

/* Третье значение - flex-basis */
/* Сколько места будет занимать элемент по умолчанию */
flex: 1 1 20%;
}

```

- Если указать `flex-grow: 0`, то элемент не будет растягиваться и будет занимать столько места, сколько у него ширина или `flex-basis`
- Значение `flex-basis` может быть в любых единицах измерения

Grid

- Для включения Grid в контейнере необходимо указать `display: grid`
- Определяем колонки и строки:

```

.grid-container {
    display: grid;

    /* Три равные по ширине колонки. Ширина каждой - 1/3 от ширины контейнера*/
    grid-template-columns: 1fr 1fr 1fr;
    grid-template-rows: auto;
}

.grid-container {
    display: grid;

    /* Две колонки. Ширина первой - 2/3 ширины контейнера, второй - 1/3 */
    grid-template-columns: 2fr 1fr;
    grid-template-rows: auto;
}

```

- Допустимо задавать размер колонок в любых других единицах измерения. Обычно используются пиксели или `fr`.
- Отступы между ячейками (строками и колонками)

```

.grid-container {
    row-gap: 10px;
    column-gap: 10px;
}

```