

Теперь рассмотрим одну из самых полезных технологий в веб-разработке — позиционирование элементов с помощью flexbox. Flexbox — это мощный инструмент для создания гибкой и адаптивной раскладки на странице. Он используется для позиционирования элементов в одном измерении (в строке или в колонке) и управления их размерами и выравниванием.

Есть два основных метода создания гибких макетов: CSS Grid (сеточная система) и Flexbox (гибкая система). В этом уроке мы сосредоточимся на flexbox.

Зачем использовать Flexbox?

Когда вы строите веб-страницу, у вас есть элементы, такие как блочные элементы (`<div>`) и строчные элементы (``). Если использовать только их, вы сможете построить базовую страницу с текстом и простым позиционированием, но при создании сложных макетов появляется потребность в более гибких и управляемых решениях. Flexbox — это как раз то, что нужно для более продвинутой компоновки элементов.

Основные особенности Flexbox:

Позволяет выстраивать элементы в одну строку или колонку.

Может автоматически распределять пространство между элементами.

Решает задачи вертикального и горизонтального выравнивания.

Создает одномерные макеты (например, навигационные панели).

Умеет переносить элементы на новые строки, если они не помещаются в одну строку.

Легко управляет порядком отображения элементов.

Упрощает создание адаптивных макетов.

Основные свойства Flexbox

1. Контейнер и элементы Flexbox

Flexbox состоит из двух основных понятий:

Flex-контейнер — это родительский элемент, который получает свойство `display: flex;`.

Flex-элементы — это дочерние элементы внутри контейнера, которые автоматически подстраиваются под гибкие правила.

2. Основные оси в Flexbox

Flexbox работает на основе осей:

Главная ось (main axis) — это ось, вдоль которой располагаются flex-элементы. Она может быть горизонтальной или вертикальной в зависимости от значения свойства flex-direction.

Поперечная ось (cross axis) — это ось, перпендикулярная главной оси.

Основные свойства Flexbox

Это базовое свойство для превращения контейнера в гибкий контейнер, что позволяет использовать все возможности flexbox.

Css

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

2. flex-direction — направление элементов

Свойство flex-direction определяет направление, в котором выстраиваются элементы вдоль главной оси.

Значения:

row (по умолчанию) — элементы выстраиваются слева направо.

row-reverse — элементы выстраиваются справа налево.

column — элементы выстраиваются сверху вниз.

column-reverse — элементы выстраиваются снизу вверх.

css

```
.container {  
    display: flex;  
    flex-direction: row;  
}
```

3. flex-wrap — управление переносом строк

Если flex-элементы не помещаются в одну строку, можно управлять их переносом на новую строку с помощью свойства flex-wrap.

Значения:

nowrap (по умолчанию) — все элементы остаются в одной строке.

wrap — элементы переносятся на новую строку, если не помещаются.

— перенос элементов, но с обратным направлением.

Пример:

css

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

4. justify-content — выравнивание по главной оси

Свойство justify-content выравнивает элементы по главной оси (обычно горизонтально).

Значения:

flex-start (по умолчанию) — элементы выравниваются к началу контейнера.

flex-end — элементы выравниваются к концу контейнера.

center — элементы выравниваются по центру.

space-between — элементы распределяются с равным пространством между ними.

space-around — элементы распределяются с одинаковым пространством вокруг каждого элемента.

css

```
.container {  
    display: flex;  
    justify-content: space-between;
```

5. align-items — выравнивание по поперечной оси

Свойство align-items управляет выравниванием flex-элементов по поперечной оси (обычно вертикально).

Значения:

stretch (по умолчанию) — элементы растягиваются на всю высоту контейнера.

flex-start — элементы выравниваются по началу поперечной оси.

flex-end — элементы выравниваются по концу поперечной оси.

center — элементы выравниваются по центру контейнера.

css

```
.container {  
    display: flex;  
    align-items: center;
```

6. flex-grow — контроль размера элементов

Свойство flex-grow управляет тем, насколько элемент может увеличиваться относительно других элементов.

Значение по умолчанию — 0, что означает, что элемент не будет увеличиваться.

Пример использования Flexbox: Создание макета страницы

html

```
<!DOCTYPE html>
```

```
<html lang="ru">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Flexbox </title>
```

```
<style>
```

```
  body {
```

```
    font-family: Arial, sans-serif;
```

```
    margin: 0;
```

```
    padding: 0;
```

```
    display: flex;
```

```
    flex-direction: column;
```

```
    align-items: center;
```

```
  }
```

```
  .container {
```

```
    display: flex;
```

```
    flex-direction: column;
```

```
    width: 1000px;
```

```
    max-width: 100%;
```

```
  }
```

```
  header, footer {
```

```
    background-color: #3498db;
```

```
    color: white;
```

```
    padding: 20px;
```

```
    text-align: center;
```

```
}
```

```
nav {  
    display: flex;  
    justify-content: space-around;  
    background-color: #2ecc71;  
    padding: 15px;  
    margin: 10px 0;  
}
```

```
nav a {  
    color: white;  
    text-decoration: none;  
}
```

```
.content {  
    display: flex;  
    justify-content: space-between;  
}
```

```
.main, .sidebar {  
    background-color: #ecf0f1;  
    padding: 20px;  
}
```

```
.main {  
    flex-grow: 2;  
}
```

```
.sidebar {
```

```
flex-grow: 1;
max-width: 300px;
}
```

```
.gallery {
  display: flex;
  justify-content: space-between;
  margin-top: 20px;
}
```

```
.gallery div {
  background-color: #bdc3c7;
  width: 30%;
  padding: 50px;
  text-align: center;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
<header>
```

```
<h1>Сайт Визитка</h1>
```

```
</header>
```

```
<nav>
```

```
<a href="#">Главная</a>
```

```
<a href="#">О нас</a>
```

```
<a href="#">Контакты</a>
```

```
</nav>
```

```
<div class="content">
  <div class="main">
    <h2>Основной контент</h2>
    <p>Здесь располагается основной текст.</p>

    <h3>Боковая панель</h3>
    <p>Дополнительная информация.</p>
  </div>
</div>

<div class="gallery">
  <div>Картинка 1</div>
  <div>Картинка 2</div>
  <div>Картинка 3</div>

  <p>Все права защищены © 2024</p>
```

Flexbox — это мощный инструмент, который упрощает создание адаптивных макетов, облегчает управление выравниванием, пространством и порядком элементов на странице. Он значительно упрощает работу с одномерными макетами.