# Лабораторная работа 3 Редактирование отступов и позиционирования элементов HTML-страниц

## Цель работы:

Изучить способы задания отступов и позиционирования элементов на HTML-странице с использованием CSS, для создания более гибкой и аккуратной верстки.

## Описание задания:

Создать веб-страницу с несколькими блоками контента и отредактировать отступы и позиционирование элементов с помощью CSS. В процессе выполнения работы вы освоите свойства margin, padding, position, float, и display.

## Инструменты:

- Текстовый редактор (например, Visual Studio Code)
- Браузер (Google Chrome, Firefox и т.д.)

## Ход работы:

### 1. Создание HTML-файла:

Создайте новый файл с расширением `.html`, например, `index.html`.
Добавьте основные HTML-теги:
```html
<!DOCTYPE html>
<html lang="ru">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Редактирование отступов и позиционирование</title>
 <link rel="stylesheet" href="styles.css">
</head>
<body>
 <!-- Контент страницы будет здесь -->
</body>
</html>
```

### 2. Добавление элементов на страницу:

Внутри `<body>`, добавьте несколько разделов, которые будут стилизованы.
```html
<div class="header">Заголовок страницы</div>
<div class="content">
 <p>Это основной контент страницы.</p>
 <img src="image.jpg" alt="Пример изображения" class="image">
</div>
<div class="sidebar">Боковая панель</div>
<div class="footer">Подвал страницы</div>
```

### 3. Создание CSS-стилей для отступов:

Создайте файл `styles.css` в той же папке, что и `index.html`.
Добавьте стили для задания внутренних и внешних отступов:
```css
body {
 font-family: Arial, sans-serif;
 margin: 0;
 padding: 0;
}
.header, .footer {
 background-color: #333;
 color: white;
 padding: 20px;
 text-align: center;
}
.content {
 margin: 20px;
 padding: 20px;
 background-color: #f2f2f2;
}
.sidebar {
 background-color: #ddd;
 padding: 15px;
 margin: 20px 0;
}
```

### 4. Изучение позиционирования элементов:

Добавьте различные типы позиционирования (`static`, `relative`, `absolute`, `fixed`) для отдельных блоков:
```css
.header {
 position: fixed;
 top: 0;
 width: 100%;
}
.footer {
 position: relative;
 bottom: 0;
}
.content {
 position: relative;
 top: 60px; /\* Смещение вниз для видимости фиксированной шапки \*/
}
```

### 5. Использование float и display для выравнивания:

Разместите боковую панель справа от основного контента с помощью `float`:
```css
.content {
 width: 70%;
 float: left;
}
.sidebar {
 width: 25%;
 float: right;
}
```
Добавьте свойство `clear` в подвал, чтобы очистить область после `float`-элементов:
```css
.footer {
 clear: both;
}
```

### 6. Проверка работы:

Откройте `index.html` в браузере и убедитесь, что все элементы находятся на своих местах, отступы работают корректно, а заголовок остается закрепленным вверху.

## Контрольные вопросы:

- В чем разница между внутренними (padding) и внешними (margin) отступами?

- Как работает свойство position и чем отличаются значения relative, absolute и fixed?

- Как свойство float помогает в выравнивании элементов?

- Что делает свойство clear и в каких случаях оно необходимо?

## Дополнительное задание:

Попробуйте использовать flexbox или grid для организации боковой панели и основного контента, а также поэкспериментируйте с различными значениями padding и margin для создания более привлекательной компоновки.