



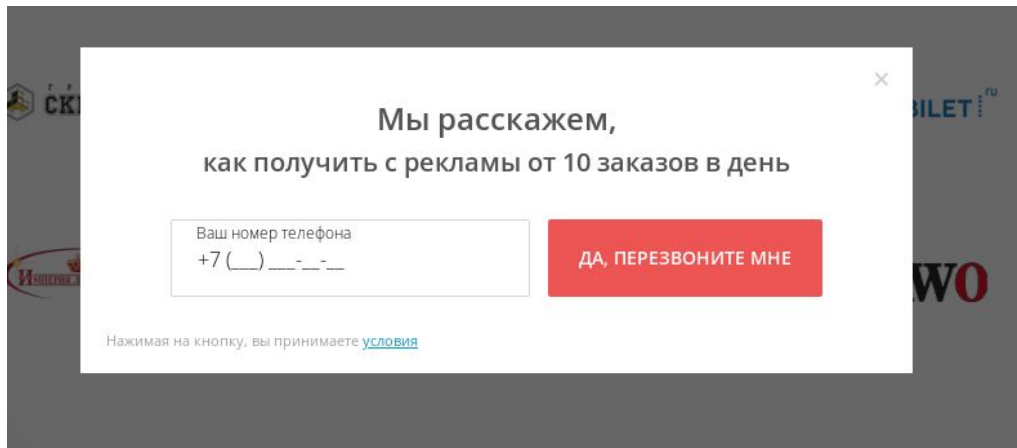
# Лекция 8.1.2

Клиент-серверная структура. Структура DOM. События. Изменение элементов



# Применение JavaScript

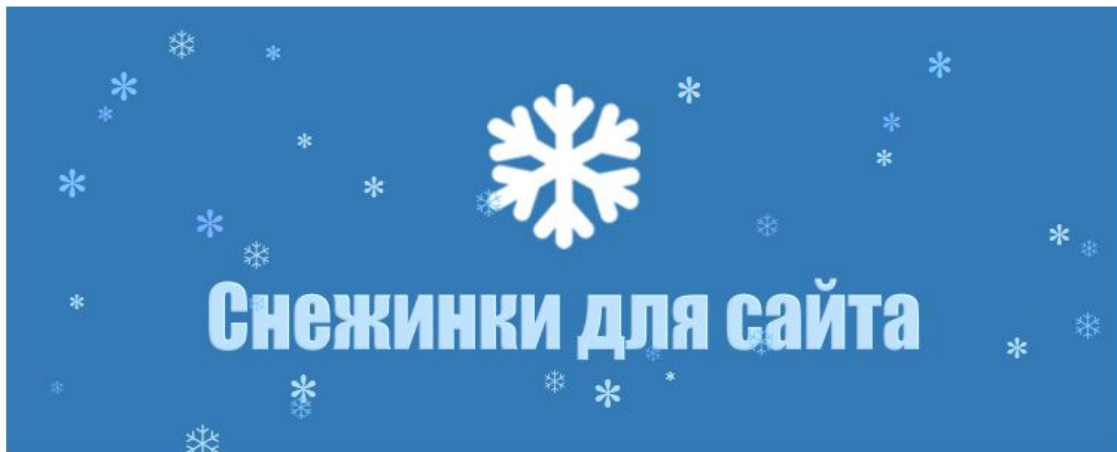
Всплывающие подсказки, движущиеся картинки, падающие снежинки и прочие анимации — за всем этим можно идти к JS.



# СНЕЖИНКИ ДЛЯ САЙТА

Главная — Блог — Создание и оформление сайтов — Весь сайт

Очень простой и быстрый способ добавить на свой сайт падающие снежинки



С наступлением зимы и Нового Года можно немного украсить свой сайт, добавив на него падающие снежинки.



## Разделы:

### > Создание и оформление сайтов

- > Тексты, ссылки и кнопки
- > Вкладки, блоки и списки
- > Секции
- > Меню и аккордеоны
- > Модальные окна
- > Слайдеры и карусели
- > Формы
- > Фото, аудио и видео



## Вкладки

Вкладка №1

Вкладка №2

Вкладка №3

**Текст вкладки №1** Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus. Curabitur nec arcu. Donec sollicitudin mi sit amet mauris. Nam elementum quam ullamcorper ante. Etiam aliquet massa et lorem. Mauris dapibus lacus auctor risus. Aenean tempor ullamcorper leo. Vivamus sed magna quis ligula eleifend.



## Спойлеры, раскрывающиеся списки и т.д.

Заголовок #1

Заголовок #2

Заголовок #3

Заголовок #1

Не следует, однако забывать, что реализация намеченных плановых заданий позволяет выполнять важные задания по разработке позиций, занимаемых участниками в отношении поставленных задач..

Заголовок #2

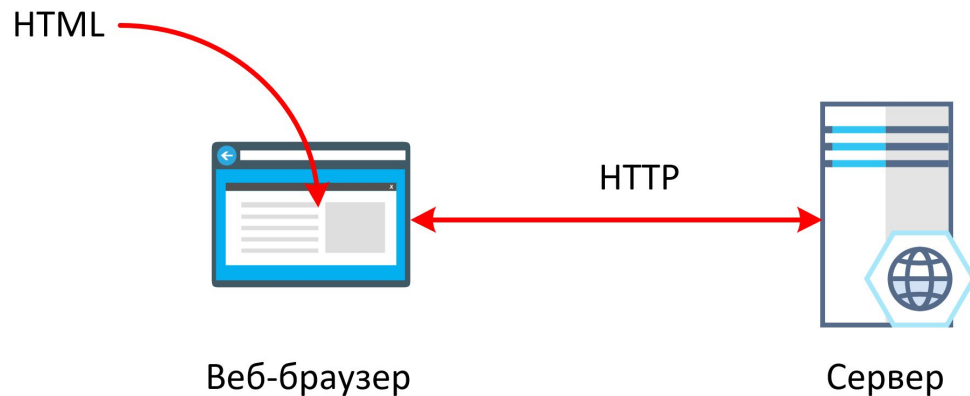
Заголовок #3

# Браузерные/Мобильные игры



# Клиент-серверная архитектура

1. Браузер запрашивает страницу по определенному адресу.
2. Браузер получает HTML-страницу. И запускает активные элементы.
3. Эти активные элементы могут **работать самостоятельно** (например снежинки) или **могут отправить запрос** после ввода данных (форма обратной связи).

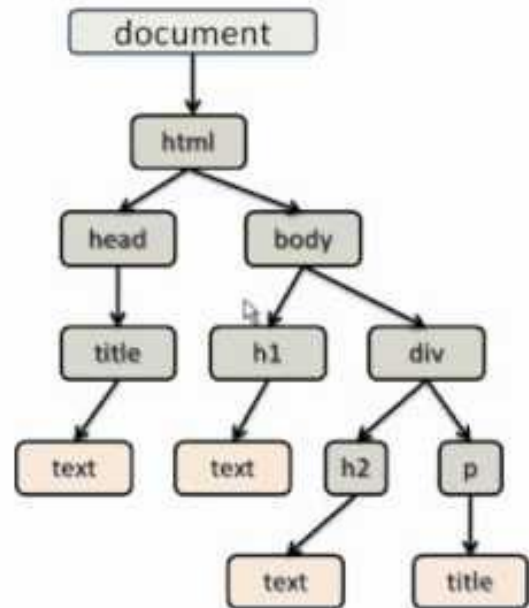


# DOM-дерево

В JavaScript вся страница представляется в DOM-элементов.

У каждого элемента может быть свой id, список классов, которые можно менять в нужный момент на JavaScript.

```
<html>  
  <head>  
    <title>Страница HTML</title>  
  </head>  
  <body>  
    <h1>Страница HTML</h1>  
    <div>  
      <h2>Раздел</h2>  
      <p>Параграф</p>  
    </div>  
  </body>  
</html>
```







## Получение элементов

Получить по ID:

```
let elem = document.getElementById('elem');
```

Получить несколько элементов по селектору:

```
let elements = document.querySelectorAll('p');
```

Получить один элемент по селектору:

```
let elem = document.querySelector('ul');
```



# События

*Событие* – это сигнал от браузера о том, что что-то произошло.

События мыши:

- **click** – происходит, когда кликнули на элемент левой кнопкой мыши (на устройствах с сенсорными экранами оно происходит при касании).
- **contextmenu** – происходит, когда кликнули на элемент правой кнопкой мыши.
- **mouseover** / **mouseout** – когда мышь наводится на / покидает элемент.
- **mousedown** / **mouseup** – когда нажали / отжали кнопку мыши на элементе.
- **mousemove** – при движении мыши.

События на элементах управления:

- **submit** – пользователь отправил форму `<form>`.
- **focus** – пользователь фокусируется на элементе, например нажимает на `<input>`.

Клавиатурные события:

- **keydown** и **keyup** – когда пользователь нажимает / отпускает клавишу.

События документа:

- **DOMContentLoaded** – когда HTML загружен и обработан, полностью построен и доступен.



# Изменение элементов

У элементов есть много параметров, например:

- **style.\*** где указывается css параметр например color, background или fontSize (в CSS он называется font-size)
- **className** содержит названия классов (например “mybutton mybluebutton”)
- **classList** содержит список классов, имеются методы **add**(“mybluebutton”) и **remove**(“mybutton”)

```
...
<body>
  <div id="elem">
    <div id="elem-content">Element</div>
  </div>
  <script>
    // получить элемент
    let elem = document.querySelector('#elem');

    // сделать его фон красным
    elem.style.background = 'red';
  </script>
</body>
...
```