

# JavaScript

---

справочник

## Методы изменения классов

**document.classList.remove('класс');**

убирает с элемента тот класс, который указан в скобках

**document.classList.add('класс');**

добавляет элементу класс, который указан в скобках

**document.classList.toggle('класс');**

убирает у элемента указанный класс, если он есть, и добавляет, если этого класса нет

// - однострочный комментарий

/\*

...

\*/ - многострочный комментарий



комментарии в JavaScript

ИСТОЧНИК: <https://htmlacademy.ru/courses/343/run/10>

## Обработчик событий по клику кнопки и метод toggle

```
let page = document.querySelector('.page'); // сокращаем запрос селектора '.page'  
let themeButton = document.querySelector('.theme-button'); // сокращаем запрос селектора
```

прописываем функцию переключения тем по нажатию на кнопку (селектор `'.theme-button'`):

```
themeButton.onclick = function() { // обработчик событий onclick (по клику кнопки)  
  page.classList.toggle('light-theme'); // удалит светлую тему  
  page.classList.toggle('dark-theme'); // добавит темную тему (следующий цикл - обратный)  
};
```

фрагмент HTML, где указана тема:

```
<body class="page light-theme">
```

фрагмент HTML, где прописана кнопка:

```
<button class="theme-button" type="button">Изменить тему</button>
```

ИСТОЧНИК: <https://htmlacademy.ru/courses/343/run/11>

## Свойство textContent

Свойство `textContent` хранит в себе текстовое содержимое элемента.

Свойствам можно присваивать новые значения:

```
let message = document.querySelector('.subscription-message'); // сокращаем запрос селектора
message.textContent = 'Какой <strong>непонятливый</strong> браузер!';
```

фрагмент HTML, к которому применяется это свойство:

```
<p class="subscription-message">Обещаем присылать вам новости не чаще одного раза в день,
причем только самые интересные и важные.</p>
```

Текст `'Обещаем присылать...'` заменится на `'Какой <strong>непонятливый</strong> браузер!'`

Причем теги не будут распознаны и отобразятся как текст.

Текст изменяется непосредственно в браузере, а не в файле HTML.

источник: <https://htmlacademy.ru/courses/343/run/13>

## Свойство value

У полей ввода (`input`) есть особое свойство - `value`.

Оно хранит данные, введенные в поле. Мы можем вывести их прямо на страницу:

```
let input = document.querySelector('input');  
console.log(input.value); // выведет в консоль keks@flashnews.ru
```

фрагмент HTML, к которому применяется это свойство:

```
<input type="email" class="subscription-email" placeholder="keks@flashnews.ru"  
value="keks@flashnews.ru" required id="subscription-email">
```

## Обработчик событий `onsubmit` и конкатенация строк в `value`

В HTML кнопка отправки имеет тип `submit`, а в JavaScript за обработку этого события отвечает свойство `onsubmit`. Обработчик событий в данном случае добавляется не на кнопку, а на саму форму:

```
let message = document.querySelector('.subscription-message');
let form = document.querySelector('.subscription');
let email = document.querySelector('.subscription-email');
form.onsubmit = function(evt) {
  evt.preventDefault();
  message.textContent = 'Адрес ' + email.value + ' добавлен в список получателей рассылки.';
};
```

фрагменты HTML, которые связаны с применением `onsubmit` и `value`:

```
<p class="subscription-message">Обещаем присылать вам новости не чаще одного раза в день, причем только
самые интересные и важные.</p> - <!-- заменится на значение message.textContent -->
<form action="https://echo.htmlacademy.ru/courses" method="post" class="subscription"> (часть кода; см.ссылку)
<input type="email" class="subscription-email" placeholder="keks@flashnews.ru" value="keks@flashnews.ru"
required id="subscription-email"> - <!-- адрес из value пойдет в значение email.value -->
```

ИСТОЧНИКИ: <https://htmlacademy.ru/courses/343/run/14> и <https://htmlacademy.ru/courses/343/run/16>

# Счетчик лайков, метод `classList.contains`, конструкция `if / else`

HTML:

```
<!-- Лайка нет, сердечко пустое -->
<button class="heart" type="button"><span class="likes-number"></span></button>
<!-- Лайк есть, сердечко закрашено. Класс added добавляется/удаляется посредством classList.toggle -->
<button class="heart added" type="button"><span class="likes-number"></span></button>
```

JS:

```
let heart = document.querySelector('.heart');
let likesNumber = document.querySelector('.likes-number');
heart.onclick = function () {
  if (heart.classList.contains('added')) { // метод classList.contains проверяет наличие класса added у heart
    likesNumber.textContent--; // уменьшает значение счетчика на 1 если heart имеет класс added (если true)
  } else {
    likesNumber.textContent++; // в обратном случае увеличивает на 1, (если условие возвращает false)
  }
  heart.classList.toggle('added'); // удаляет или добавляет added к классу heart в зависимости от его наличия
};
```

ИСТОЧНИК: <https://htmlacademy.ru/courses/347/run/2> и до 7-го задания включительно

# Комментарии, методы createElement и append

HTML:

```
<ol class="comment-list">
  <li class="user-comment">Да ну, бред какой-то...</li>
</ol>
<form action="https://htmlacademy.ru/courses" method="post" class="comment-form">
  <div class="comment-inner">
    <label class="comment-label" for="comment-input">ваш комментарий</label>
    <input type="text" class="comment-field" value="А Java и JavaScript - это одно и то же?" placeholder="Комментарий"
      required id="comment-input">
  </div>
  <button class="button" type="submit">Отправить</button>
</form>
```

JS:

```
let commentForm = document.querySelector('.comment-form');
let commentList = document.querySelector('.comment-list');
let commentField = document.querySelector('.comment-field');
commentForm.onsubmit = function (evt) {
  evt.preventDefault();
  let newComment = document.createElement('li'); // для newComment присваиваем создание нового элемента списка: <li>
  newComment.classList.add('user-comment'); // добавляем новому элементу <li> класс user-comment для сохранения стиля
  newComment.textContent = commentField.value; // в newComment записываем значение value из поля input
  commentField.value = ""; // очищаем поле для нового комментария
  commentList.append(newComment); // сохраненное значение value в newComment записывается новым элементом в список <ol>
};
```

ИСТОЧНИК: <https://htmlacademy.ru/courses/347/run/8> и до 14-го задания включительно

[тренировка](#)



# Метод querySelectorAll, попап (tooltip), data-атрибут, цикл for

HTML:

```
<div class="tooltip">
  <span class="tooltip-text"></span>
  <button class="button close-button" type="button"><span>Закрыть</span></button>
</div> ...
...<p>Редакция Flashnews! подготовила подарок для своих подписчиков - приложение <button class="tooltip-button" type="button"
data-tooltip-text="это когда у тебя в гостиной покемоны">дополненной реальности</button>, в котором вы сможете погладить <button
class="tooltip-button" type="button" data-tooltip-text="что-то среднее между свиньей и тушканчиком">трубкозуба</button>, вырастить
<button class="tooltip-button" type="button" data-tooltip-text="сам не знаю, погугли">питахайю</button> или поговорить с <button
class="tooltip-button" type="button" data-tooltip-text="доступно только для владельцев годовой подписки">Кексом</button>.</p>
JS:
```

```
let tooltip = document.querySelector('.tooltip');
let closeButton = document.querySelector('.close-button');
let tooltipButtons = document.querySelectorAll('.tooltip-button'); // создание коллекции селекторов: с tooltipButtons[0] по tooltipButtons[3]
let tooltipText = document.querySelector('.tooltip-text');
closeButton.onclick = function () { // обработчик событий закрывает попап с текстом
  tooltip.classList.remove('opened'); // удаляя класс opened,
};
for (let tooltipButton of tooltipButtons) { // перебор значений коллекции tooltipButtons через переменную tooltipButton
  tooltipButton.onclick = function () { // обработчик событий срабатывает для каждого значения tooltipButton (от [0] до [3])
    tooltipText.textContent = tooltipButton.dataset.tooltipText; // каждое значение data-tooltip-text записывается в свой tooltipText
    tooltip.classList.add('opened'); // добавляется класс opened, чтобы открылся попап с соответствующим значением tooltipText
  };
}
```

ИСТОЧНИК: <https://htmlacademy.ru/courses/349/run/1> и до 9-го задания включительно

## Обработчик событий oninput, свойство length, тег <output>, свойство disabled (блокирование кнопки)

HTML:

```
<ol class="comment-list">
  <li class="user-comment">С точки зрения банальной эрудиции, каждый индивидуум, цинизм помыслов которого ассоциирует
  концепции парадоксальных иллюзий, просто не может не игнорировать критерии утопического субъективизма. А с точки зрения
  ценности или содержания этого логического утверждения посредством чисто отрицательного предиката и определяется, прибавляет ли
  оно что-нибудь ко всей совокупности знания.<br>
  </li>
</ol>
<form action="https://echo.htmlacademy.ru/courses" method="post" class="comment-form">
  <div class="comment-inner">
    <label class="comment-label" for="comment-textarea">ваш комментарий</label>
    <textarea class="comment-field comment-area" rows="3" placeholder="Комментарий" required id="comment-textarea"></textarea>
  </div>
  <span class="text-counter">Использовано <output class="char-counter" for="comment-textarea">0</output>/142 символов</span>
  <button class="button submit-button" type="submit">Отправить</button>
</form>
```

JS:

```
let commentForm = document.querySelector('.comment-form'); // класс в самой форме для ввода нового комментария
let commentList = document.querySelector('.comment-list'); // класс в списке, куда добавляются новые комментарии
let commentField = document.querySelector('.comment-field'); // поле ввода текста
let charCounter = document.querySelector('.char-counter'); // счетчик набранных символов в комментарии
let submitButton = document.querySelector('.submit-button'); // кнопка "отправить" под полем набора текста

commentForm.onsubmit = function (evt) { // обработчик события "отправка формы"
  evt.preventDefault();
  let newComment = document.createElement('li'); // создание нового элемента в списке комментариев
  newComment.classList.add('user-comment'); // приведение новой записи к единому стилю с другими комментариями
  newComment.textContent = commentField.value; // сохранение введенного комментария в переменной нового элемента списка
  commentField.value = ''; // очистка поля ввода комментария
  commentList.append(newComment); // добавление нового комментария в основной список комментариев
  charCounter.textContent = 0; // обнуление счетчика использованных символов
};

commentField.oninput = function () { // обработчик событий при изменении значения (при вводе)
  charCounter.textContent = commentField.value.length; // отображает кол-во набранных символов в реальном времени
  if (commentField.value.length > 142) {
    commentForm.classList.add('warning'); // добавляет класс о предупреждении превышения 142 символа
    submitButton.disabled = true; // disabled блокирует кнопку отправки
  } else {
    commentForm.classList.remove('warning'); // удаляет класс о предупреждении
    submitButton.disabled = false; // снимает блокировку кнопки "Отправить" при length <= 142
  }
};
```

ИСТОЧНИК: <https://htmlacademy.ru/courses/349/run/10> и до 15-го задания включительно

[тренировка](#)