## **Задачи для решения**

### **Работа с классами**

*Для решения задач данного блока вам понадобятся следующие свойства:*[*classList*](http://old.code.mu/javascript/dom/classList.html)*.*

 Дан элемент **#elem**. Добавьте ему класс **www**.

 Дан элемент **#elem**. Удалите у него класс **www**.

 Дан элемент **#elem**. Проверьте наличие у него класса **www**.

 Дан элемент **#elem**. Добавьте ему класс **www**, если его нет и удалите - если есть.

 Дан элемент **#elem**. Узнайте количество его классов.

 Дан элемент **#elem**. Выведите последовательно алертом его классы.

### **Работа с CSS**

*Для решения задач данного блока вам понадобятся следующие свойства:*[*cssText*](http://old.code.mu/javascript/dom/cssText.html)*.*

 Дан элемент **#elem**. Сделайте его красного цвета, размером 30px, добавьте ему границу.

### **Свойство tagName**

*Для решения задач данного блока вам понадобятся следующие свойства:*[*tagName*](http://old.code.mu/javascript/dom/tagName.html)*.*

 Дан элемент **#elem**. По клику на него выведите название его тега.

 Дан элемент **#elem**. По клику на него выведите название его тега *в нижнем регистре*.

 Даны элементы с классом **www**. Добавьте каждому элементу в конец название его тега в нижнем регистре.

### **Вставка элементов через appendChild**

*Для решения задач данного блока вам понадобятся следующие методы:*[*createElement*](http://old.code.mu/javascript/dom/createElement.html)*,*[*appendChild*](http://old.code.mu/javascript/dom/appendChild.html)*.*

 Дан **ol**. Вставьте ему в конец **li** с текстом 'пункт'.

 Дан **ul**. Дан массив. Вставьте элементы этого массива в конец **ul** так, чтобы каждый элемент стоял в своем **li**.

### **Привязывание событий при вставке**

 Дан **ul**. Дан массив. Вставьте элементы этого массива в конец **ul** так, чтобы каждый элемент стоял в своем **li**. Сделайте так, чтобы к вставляемым li было привязано следующее событие: по нажатию на li она должна вывести на экран свой текст.

### **Вставка элементов через insertBefore**

*Для решения задач данного блока вам понадобятся следующие методы:*[*insertBefore*](http://old.code.mu/javascript/dom/insertBefore.html)*.*

 Дан элемент **ul**, а в нем li **#elem**. Вставьте перед элементом **#elem** новую **li** с текстом '!!!'.

### **Вставка элементов через insertAdjacentHTML**

*Для решения задач данного блока вам понадобятся следующие методы:*[*insertAdjacentHTML*](http://old.code.mu/javascript/dom/insertAdjacentHTML.html)*.*

 Дан элемент **#elem**. Вставьте перед ним **span** с текстом '!!!'.

 Дан элемент **#elem**. Вставьте после него **span** с текстом '!!!'.

 Дан элемент **#elem**. Вставьте ему в начало **span** с текстом '!!!'.

 Дан элемент **#elem**. Вставьте ему в конец **span** с текстом '!!!'.

### **Потомки**

*Для решения задач данного блока вам понадобятся следующие свойства:*[*firstElementChild*](http://old.code.mu/javascript/dom/firstElementChild.html)*,*[*lastElementChild*](http://old.code.mu/javascript/dom/lastElementChild.html)*,*[*children*](http://old.code.mu/javascript/dom/children.html)*.*

 Дан элемент **#elem**. Найдите **первого** потомка этого элемента и сделайте его текст красного цвета.

 Дан элемент **#elem**. Найдите **последнего** потомка этого элемента и сделайте его текст красного цвета.

 Дан элемент **#elem**. Найдите **всех** потомков этого элемента и добавьте им в конец текст '!'.

### **Соседи**

*Для решения задач данного блока вам понадобятся следующие свойства:*[*previousElementSibling*](http://old.code.mu/javascript/dom/previousElementSibling.html)*,*[*nextElementSibling*](http://old.code.mu/javascript/dom/nextElementSibling.html)*.*

 Дан элемент **#elem**. Найдите его соседа сверху и добавьте ему в конец текст '!'.

 Дан элемент **#elem**. Найдите его соседа снизу и добавьте ему в конец текст '!'.

 Дан элемент **#elem**. Найдите его соседа снизу его соседа снизу (следующий элемент за соседним) и добавьте ему в конец текст '!'.

### **Родители**

*Для решения задач данного блока вам понадобятся следующие свойства:*[*parentElement*](http://old.code.mu/javascript/dom/parentElement.html)*,*[*parentNode*](http://old.code.mu/javascript/dom/parentNode.html)*.*

 Дан элемент **#elem**. Найдите его родителя и покрасьте его в красный цвет.

 Дан элемент **#elem**. Найдите родителя его родителя и покрасьте его в красный цвет.

### **Удаление и клонирование**

*Для решения задач данного блока вам понадобятся следующие свойства:*[*removeChild*](http://old.code.mu/javascript/dom/removeChild.html)*.*

 Дан элемент **#parent**, внутри него дан элемент **#child**. Дана кнопка. По нажатию на эту кнопку удалите элемент **#child**.

 Дан **ol**. По нажатию на кнопку получите его последнего потомка и удалите его.

 Дан элемент. Сделайте так, чтобы по нажатию по нему этот элемент удалялся.

 Дан **ol**, а внутри него **li**. Сделайте так, чтобы по нажатию на любую **li** эта **li** удалялась.

### **Клонирование**

*Для решения задач данного блока вам понадобятся следующие свойства:*[*cloneNode*](http://old.code.mu/javascript/dom/cloneNode.html)*.*

 Дан инпут. Дана кнопка. По нажатию на кнопку клонируйте этот инпут.

### **Практика**

 Дан массив. Создайте **ul** через createElement, затем вставьте каждый элемент этого массива в отдельную **li** внутри этой **ul**, затем вставьте эту **ul** в конец **body**.

 Дан инпут. Рядом с ним находится кнопочка "+". По нажатию на эту кнопку под нашим инпутом должен появится еще один пустой инпут.

 Дан инпут. В него вводится число. По потери фокуса сделайте так, чтобы каждая цифра вставилась в новый инпут. Инпутов для цифр изначально не существует, они должны создаться в процессе работы скрипта.

 Дана кнопка. Сделайте так, чтобы по нажатию на эту кнопку, скрывался родитель этой кнопки.

## **Задачи для решения**

### **На величину границы**

*Для решения задач данного блока вам понадобятся следующие методы:*[*clientTop*](http://old.code.mu/javascript/metric/clientTop.html)*,*[*clientLeft*](http://old.code.mu/javascript/metric/clientLeft.html)*.*

 Дан элемент **#elem** с границами. По нажатию на кнопку выведите толщину его верхней границы. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-na-javascript.html)

 Дан элемент **#elem** с границами. По нажатию на кнопку выведите толщину его левой границы. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-na-javascript.html)

### **На полный размер элемента**

*Для решения задач данного блока вам понадобятся следующие методы:*[*offsetWidth*](http://old.code.mu/javascript/metric/offsetWidth.html)*,*[*offsetHeight*](http://old.code.mu/javascript/metric/offsetHeight.html)*.*

 Дан элемент **#elem** с границами. По нажатию на кнопку выведите его полную ширину с учетом границы и padding. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-na-javascript.html)

 Дан элемент **#elem** с границами. По нажатию на кнопку выведите его полную высоту с учетом границы и padding. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-na-javascript.html)

### **На размер элемента без границ, но с padding**

*Для решения задач данного блока вам понадобятся следующие методы:*[*clientWidth*](http://old.code.mu/javascript/metric/offsetWidth.html)*,*[*clientHeight*](http://old.code.mu/javascript/metric/clientHeight.html)*.*

 Дан элемент **#elem** с границами. По нажатию на кнопку выведите его полную ширину без учета границы, но с padding. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-na-javascript.html)

 Дан элемент **#elem** с границами. По нажатию на кнопку выведите его полную высоту без учета границы, но с padding. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-na-javascript.html)

### **Работа с getComputedStyle**

*Для решения задач данного блока вам понадобятся следующие функции:*[*getComputedStyle*](http://old.code.mu/javascript/dom/getComputedStyle.html)*.*

 Дан элемент **#elem**. Получите его ширину и высоту, без учета границы и padding. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-na-javascript.html)

### **Прокрутка элемента**

*Для решения задач данного блока вам понадобятся следующие методы:*[*scrollTop*](http://old.code.mu/javascript/metric/scrollTop.html)*,*[*scrollLeft*](http://old.code.mu/javascript/metric/scrollLeft.html)*.*

 Дан элемент **#elem** с вертикальной полосой прокрутки. По нажатию на кнопку выведите на сколько элемент прокручен сверху. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-na-javascript.html)

 Дан элемент **#elem** с горизонтальной полосой прокрутки. По нажатию на кнопку выведите на сколько элемент прокручен слева. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-na-javascript.html)

 Дан элемент **#elem** с вертикальной полосой прокрутки. По нажатию на кнопку прокрутите его до позиции 100px сверху. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-na-javascript.html)

 Дан элемент **#elem** с вертикальной полосой прокрутки. По нажатию на кнопку прокрутите его на 100px вниз от текущего положения. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-na-javascript.html)

### **Прокрутка элемента**

*Для решения задач данного блока вам понадобятся следующие методы:*[*scrollWidth*](http://old.code.mu/javascript/metric/scrollWidth.html)*,*[*scrollHeight*](http://old.code.mu/javascript/metric/scrollHeight.html)*.*

 Дан элемент **#elem** с вертикальной полосой прокрутки. По нажатию на кнопку выведите реальную высоту элемента с учетом прокрутки. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-na-javascript.html)

 Дан элемент **#elem** с горизонтальной полосой прокрутки. По нажатию на кнопку выведите реальную ширину элемента с учетом прокрутки. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-na-javascript.html)

 Дан элемент **#elem** с вертикальной полосой прокрутки. По нажатию на кнопку прокрутите его до позиции 100px от нижнего края элемента. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-na-javascript.html)

### **Прокрутка страницы**

*Для решения задач данного блока вам понадобятся следующие методы:*[*pageXOffset*](http://old.code.mu/javascript/metric/pageXOffset.html)*,*[*pageYOffset*](http://old.code.mu/javascript/metric/pageYOffset.html)*.*

 Дана страница с вертикальной полосой прокрутки. По нажатию на кнопку узнайте на сколько страница прокручена по вертикали. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-na-javascript.html)

 Дана страница с горизонтальной полосой прокрутки. По нажатию на кнопку узнайте на сколько страница прокручена по горизонтали. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-na-javascript.html)

### **Прокрутка страницы**

*Для решения задач данного блока вам понадобятся следующие методы:*[*scrollTo*](http://old.code.mu/javascript/metric/scrollTo.html)*,*[*scrollBy*](http://old.code.mu/javascript/metric/scrollBy.html)*,*[*scrollIntoView*](http://old.code.mu/javascript/metric/scrollIntoView.html)*.*

 Дана страница с горизонтальной и вертикальной полосами прокрутки. По нажатию на кнопку прокрутите ее в точку 300px слева, 500px сверху. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-na-javascript.html)

 Дана страница с вертикальной полосой прокрутки. По нажатию на кнопку прокрутите на 300px вниз от текущего положения. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-na-javascript.html)

### **Практика**

 По нажатию на кнопку прокрутите страницу до самого низа. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-na-javascript.html)

 По нажатию на кнопку прокрутите страницу на 400px от текущего положения. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-na-javascript.html)

 По нажатию на кнопку проверьте, прокручена ли страница до самого низа. Если это так - прокрутите ее в положение 100px от верхнего края. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-na-javascript.html)

 Дан элемент **#elem**. По клику на него увеличьте его ширину и высоту в 2 раза. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-na-javascript.html)

## **Задачи для решения**

 Дана кнопка. По нажатию на у кнопку узнайте, есть ли у окна браузера вертикальная прокрутка. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-dlya-window-na-javascript.html)эту кнопку выведите высоту окна браузера. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-dlya-window-na-javascript.html)

 Дана кнопка. По нажатию на эту кнопку прокрутите окно браузера до самого низа. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-dlya-window-na-javascript.html)

 Дана кнопка. По нажатию на эту кнопку узнайте, есть ли у окна браузера вертикальная прокрутка. [Показать решение.](http://old.code.mu/tasks/javascript/dom/rabota-s-metrikami-dlya-window-na-javascript.html)

## Нахождение и оборачивание текста в тег

Реализуйте функцию prettify, которая находит текст (дочерние текстовые ноды) внутри элемента div и оборачивает текст в параграф

## Поиск и нормализация имён классов

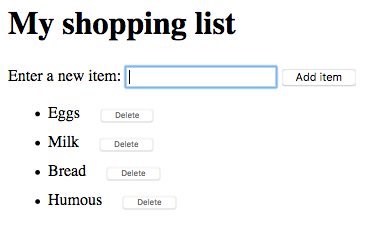
Реализуйте функцию normalize, которая нормализует имена классов для всех элементов на странице. В данном случае это означает что происходит преобразование всех классов написанных используя kebab нотацию в camelCase нотацию: text-center => textCenter

## динамический список покупок

Чтобы завершить статью, мы хотели бы задать вам небольшой вызов - мы хотим сделать простой пример списка покупок, который позволяет динамически добавлять элементы в список с помощью ввода формы и кнопки. Когда вы добавляете элемент на вход и нажимаете кнопку:

* Элемент должен появиться в списке.
* Каждому элементу должна быть предоставлена кнопка, которую можно нажать, чтобы удалить этот элемент из списка.
* Вход должен быть опустошен и сфокусирован, чтобы вы могли ввести другой элемент.

Готовое демо будет выглядеть примерно так:



Чтобы завершить упражнение, выполните следующие действия и убедитесь, что список ведет себя так, как описано выше.

1. Для начала загрузите копию нашего начального файла [shopping-list.html](https://github.com/mdn/learning-area/blob/master/javascript/apis/document-manipulation/shopping-list.html) и скопируйте его где-нибудь. Вы увидите, что у него есть минимальный CSS, список с меткой, ввод и кнопка, пустой список и элемент [<script>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/script). Вы будете делать все свои дополнения внутри скрипта.
2. Создайте три переменные, содержащие ссылки на список ([<ul>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/ul), [<input>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/input) и <button> элементы.
3. Создайте [function](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Building_blocks/Functions), которая будет запускаться в ответ на нажатие кнопки.
4. Внутри тела функции начните с сохранения текущего значения ([value](https://developer.mozilla.org/en-US/docs/Web/API/HTMLInputElement" \l "Properties)) входного элемента в переменной.
5. Затем очистите элемент ввода, установив его значение в пустую строку — ''.
6. Создайте три новых элемента - элемент списка ([<li>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/li)), [<span>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/span) и <button> и сохраните их в переменных.
7. Добавьте диапазон и кнопку в качестве дочерних элементов списка.
8. Установите текстовое содержимое диапазона на значение входного элемента, которое вы сохранили ранее, и текстовое содержимое кнопки «Удалить».
9. Добавить элемент списка в качестве дочернего списка.
10. Прикрепите обработчик события к кнопке удаления, чтобы при щелчке удалял весь элемент списка, внутри которого он находится.
11. Наконец, используйте метод [focus()](https://developer.mozilla.org/en-US/docs/Web/API/HTMLElement/focus), чтобы сфокусировать входной элемент, готовый для входа в следующий элемент списка покупок.