**Критерии приема**

* Создан репозиторий goit-js-hw-07
* При сдаче домашней работы есть две ссылки: на исходные файлы и рабочую страницу на GitHub pages
* Задание выполнено строго по ТЗ
* Каждое задание выполнено в отдельном файле с именем task-номер\_задания.js.
* При посещении рабочей страницы (GitHub pages) задания, в консоли нету ошибок и предупреждений
* Имена переменных и функций понятные, описательные
* Код отформатирован с помощью Prettier

**Стартовые файлы**

В папке [src](https://github.com/goitacademy/javascript-homework/blob/main/homework-07/src) ты найдешь стартовые файлы проекта с базовой разметкой и подключенными файлами скриптов для каждого задания.

**Задание 1**

В HTML есть список категорий ul#categories.

<ul id="categories">

<li class="item">

<h2>Животные</h2>

<ul>

<li>Кот</li>

<li>Хомяк</li>

<li>Лошадь</li>

<li>Попугай</li>

</ul>

</li>

<li class="item">

<h2>Продукты</h2>

<ul>

<li>Хлеб</li>

<li>Петрушка</li>

<li>Творог</li>

</ul>

</li>

<li class="item">

<h2>Технологии</h2>

<ul>

<li>HTML</li>

<li>CSS</li>

<li>JavaScript</li>

<li>React</li>

<li>Node</li>

</ul>

</li>

</ul>

Напиши скрипт, который выполнит следующие операции.

Посчитает и выведет в консоль количество категорий в ul#categories, то есть элементов li.item. Получится 'В списке 3 категории.'.

Для каждого элемента li.item в списке ul#categories, найдет и выведет в консоль текст заголовка элемента (тега h2) и количество элементов в категории (всех вложенных в него элементов li).

Например для первой категории получится:

* Категория: Животные
* Количество элементов: 4

**Задание 2**

В HTML есть пустой список ul#ingredients.

<ul id="ingredients"></ul>

В JS есть массив строк.

const ingredients = [

'Картошка',

'Грибы',

'Чеснок',

'Помидоры',

'Зелень',

'Приправы',

];

Напиши скрипт, который для каждого элемента массива ingredients создаст отдельный li, после чего вставит все li за одну операцию в список ul.ingredients. Для создания DOM-узлов используй document.createElement().

**Задание 3**

Напиши скрипт для создания галлереи изображений по массиву данных.

В HTML есть список ul#gallery.

<ul id="gallery"></ul>

Используй массив объектов images для создания тегов img вложенных в li. Для создания разметки используй шаблонные строки и insertAdjacentHTML().

* Все элементы галереи должны добавляться в DOM за одну операцию вставки.
* Добавь минимальное оформление галереи флексбоксами или гридами через css-классы.

const images = [

{

url: 'https://images.pexels.com/photos/140134/pexels-photo-140134.jpeg?auto=compress&cs=tinysrgb&dpr=2&h=750&w=1260',

alt: 'White and Black Long Fur Cat',

},

{

url: 'https://images.pexels.com/photos/213399/pexels-photo-213399.jpeg?auto=compress&cs=tinysrgb&dpr=2&h=750&w=1260',

alt: 'Orange and White Koi Fish Near Yellow Koi Fish',

},

{

url: 'https://images.pexels.com/photos/219943/pexels-photo-219943.jpeg?auto=compress&cs=tinysrgb&dpr=2&h=750&w=1260',

alt: 'Group of Horses Running',

},

];

**Задание 4**

Счетчик состоит из спана и кнопок, которые должны увеличивать и уменьшать значение счетчика на 1.

* Создай переменную counterValue в которой будет хранится текущее значение счетчика.
* Создай функции increment и decrement для увеличения и уменьшения значения счетчика
* Добавь слушатели кликов на кнопки, вызовы функций и обновление интерфейса

<div id="counter">

<button type="button" data-action="decrement">-1</button>

<span id="value">0</span>

<button type="button" data-action="increment">+1</button>

</div>

**Задание 5**

Напиши скрипт который, при наборе текста в инпуте input#name-input (событие input), подставляет его текущее значение в span#name-output. Если инпут пустой, в спане должна отображаться строка 'незнакомец'.

<input type="text" placeholder="Ваше имя?" id="name-input" />

<h1>Привет, <span id="name-output">незнакомец</span>!</h1>

**Задание 6**

Напиши скрипт, который бы при потере фокуса на инпуте, проверял его содержимое на правильное количество символов.

<input

type="text"

id="validation-input"

data-length="6"

placeholder="Введи 6 символов"

/>

* Сколько символов должно быть в инпуте, указывается в его атрибуте data-length.
* Если введено подходящее количество, то border инпута становится зеленым, если неправильное - красным.

Для добавления стилей, используй CSS-классы valid и invalid.

#validation-input {

border: 3px solid #bdbdbd;

}

#validation-input.valid {

border-color: #4caf50;

}

#validation-input.invalid {

border-color: #f44336;

}

**Задание 7**

Напиши скрипт, который реагирует на изменение значения input#font-size-control (событие input) и изменяет инлайн-стиль span#text обновляя свойство font-size. В результате при перетаскивании ползунка будет меняться размер текста.

<input id="font-size-control" type="range" />

<br />

<span id="text">Абракадабра!</span>

**Задание 8 - дополнительное, выполнять не обязательно**

Напиши скрипт создания и очистки коллекции элементов. Пользователь вводит количество элементов в input и нажимает кнопку Создать, после чего рендерится коллекция. При нажатии на кнопку Очистить, коллекция элементов очищается.

Создай функцию createBoxes(amount), которая принимает 1 параметр amount - число. Функция создает столько div, сколько указано в amount и добавляет их в div#boxes.

Каждый созданный div:

* Имеет случайный rgb цвет фона
* Размеры самого первого div - 30px на 30px
* Каждый следующий div после первого, должен быть шире и выше предыдущего на 10px

Создай функцию destroyBoxes(), которая очищает div#boxes.

<div id="controls">

<input type="number" min="0" max="100" step="1" />

<button type="button" data-action="render">Создать</button>

<button type="button" data-action="destroy">Очистить</button>

</div>

<div id="boxes"></div>