## Задание 1

На странице есть верстка:

<div id="container"></div>

<input id="add-item-btn" type="button" value="+">

Необходимо написать скрипт, который при нажатии кнопки "+" добавляет в контейнер новый <div> с классом element и текстом Element.

Результат в инспекторе после трех кликов:

<div id="container">

<div class="element">Element</div>

<div class="element">Element</div>

<div class="element">Element</div>

</div>

<input id="btn" type="button" value="+">

## Задание 2

На странице есть верстка:

<div id="container"></div>

<a id="start">Start</a> |

<a id="stop">Stop</a>

Необходимо написать функцию, которая каждые пять секунд будет добавлять в контейнер тег <div> с классом element, текст элемента - случайное число от 0 до 20.

Результат в инспекторе после 15-ти секунд (числа случайны и могут не совпадать с вашими):

<div id="container">

<div class="element">4</div>

<div class="element">10</div>

<div class="element">8</div>

</div>

Указания:

* При нажатии на ссылку "Stop" элементы должны прекратить добавляться
* При нажатии на ссылку "Start" элементы должны снова начать добавляться
* При запуске программы сделать, чтобы элементы начали добавляться.

## Задание 3

Есть такая верстка:

<div>

<span id="minutes">00</span>:

<span id="seconds">00</span>

</div>

<button id="start">Start</button>

<button id="pause">Pause</button>

<button id="reset">Reset</button>

Необходимо написать "таймер", который будет запускаться при нажатии кнопки "Start", приостанавливаться на "Pause", и сбрасываться на "Reset".

Указания:

* Один раз в секунду будет запускаться функция, которая должна обновлять текст в div id="seconds", прибавлять единицу.
* Вы можете пользоваться функцией "дополнить нулями" из одного из прошлых примеров в раздатках, чтобы выводить числа "01, 02, 03" и т.п.
* Когда число в секундах станет равным 60, нужно увеличить счетчик "минут" на единицу, а счетчик секунд сбросить на 0.
* Кнопка "сброса" сбрасывает таймер на 00:00
* Можете дополнительно стилизовать таймер и кнопки, чтобы выглядело красиво. Если нужно, можно модифицировать разметку как угодно.

## Задание 4

На странице есть верстка:

<form>

<input type="text" name="username" />

<input type="text" name="password" />

<input type="submit" value="Log in" disabled />

</form>

Необходимо написать код, который делает кнопку "Log in" активной, только если в поля username и password введены строки не меньше 6 символов длиной. При нажатии на кнопку "Log in" нужно отобразить alert со введенными данными и не перезагружать страницу.

Указания:

* Использовать событие "change" или "keyup" для отслеживания длины строк.
* Событие change срабатывает у элемента input, когда пользователь меняет в нем текст, а затем меняет фокус на другой элемент (например, кликнуть куда-то вне страницы, или перемещается на другой input.
* Событие keyup срабатывает у элемента input всякий раз когда пользователь нажимает и затем отпускает кнопку на клавиатуре при вводе текста в это текстовое поле.
* Активность элемента изменять через атрибут disabled. Чтобы включить, передаете true. Чтобы выключить - false.

## Задание 5 (дополнительное, необязательное)

На странице есть верстка:

<p class="question">

Висит груша - нельзя скушать.

<span class="answer">Лампочка</span>

</p>

<p class="question">

С луком с яйцами, но не пирожок?

<span class="answer">Робин Гуд</span>

</p>

<p class="question">

Один глаз, один рог, но не носорог?

<span class="answer">Корова из-за угла выглядывает</span>

</p>

Необходимо создать функцию, при вызове которой добавятся кнопки "Посмотреть ответ" к каждому вопросу, при нажатии на которую соответствующий элемент .answer отобразится, а при повторном нажатии - скроется.

Указания:

Для того, чтобы найти "текущий" элемент, в функции-обработчике события можно воспользоваться словом this:  
const currentElement = this;

* Еще можно использовать анимацию для скрытия и показа элементов
* В этом задании вы можете модифицировать верстку как угодно, чтобы вам было удобнее находить элементы