*// 2 день*

*// Сделали функцию closeModal и closeModalEsc, использовал метод spread, проверку валидации формы*

*// создали константу dataBase, добавили событие форме('submit')*

const dataBase = [];

modalBtnWarning = document.querySelector('.modal\_\_btn-warning');

*// Метод spread(...) коллекция разбивает на элементы, затем все элементы мы засовывает в массив, потом фильтруем элементы, чтоб не было BUTTON*

const elementsModalSubmit = [...modalSubmit.elements].filter((elem) => elem.tagName !== 'BUTTON');

*// Функция закрытия модального окна*

const closeModal = function(event) {

  const target = event.target;

*if* (target.closest('.modal\_\_close') || target === this) {

    this.classList.add('hide');

*if* (this === modalAdd) {

      modalSubmit.reset();

    }

  }

};

*// Функия закрытия модального окна через Esc*

const closeModalEsc = (event) => {

*if* (event.code === 'Escape') {

    modalAdd.classList.add('hide');

    modalItem.classList.add('hide');

    modalSubmit.reset();

    document.body.removeEventListener('keydown', closeModalEsc);

  }

};

*// Закрытие модалки с помощью крестика и фона, перезагрузка формы*

modalAdd.addEventListener('click', closeModal);

*// Закрытие модалки - товара*

modalItem.addEventListener('click', closeModal);

*// Проверка валидации формы, разблокировка кнопки и убирание текста*

*// Every возвращает true если все элементы не пустые*

modalSubmit.addEventListener('input', () => {

  const validForm = elementsModalSubmit.every(elem => elem.value);

  modalBtnSubmit.disabled = !validForm;

  modalBtnWarning.style.display = validForm ? 'none' : '';

});

*// Отправка формы*

modalSubmit.addEventListener('submit',(event) => {

  event.preventDefault();

  const itemObj = {};

*for* (const elem of elementsModalSubmit) {

    itemObj[elem.name] = elem.value;

  }

  dataBase.push(itemObj);

  console.log(dataBase);

});