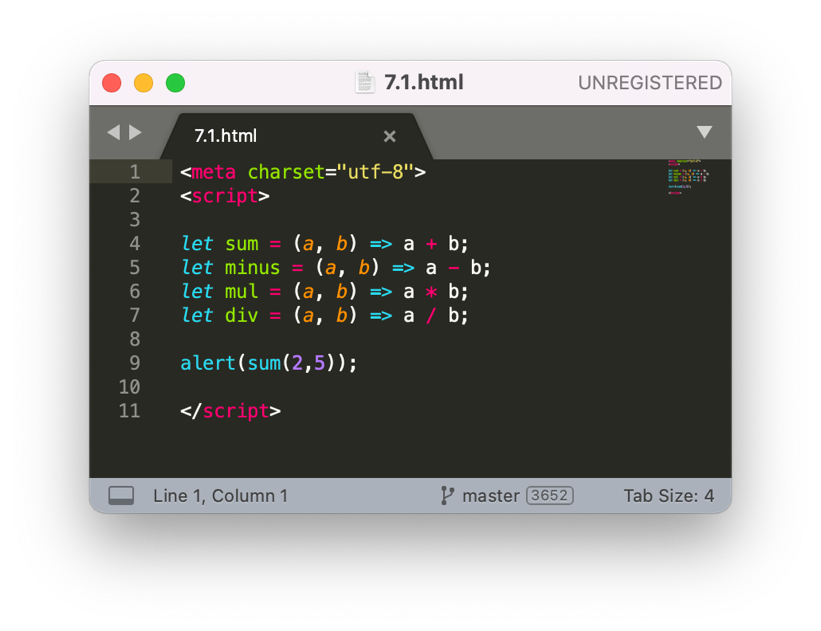
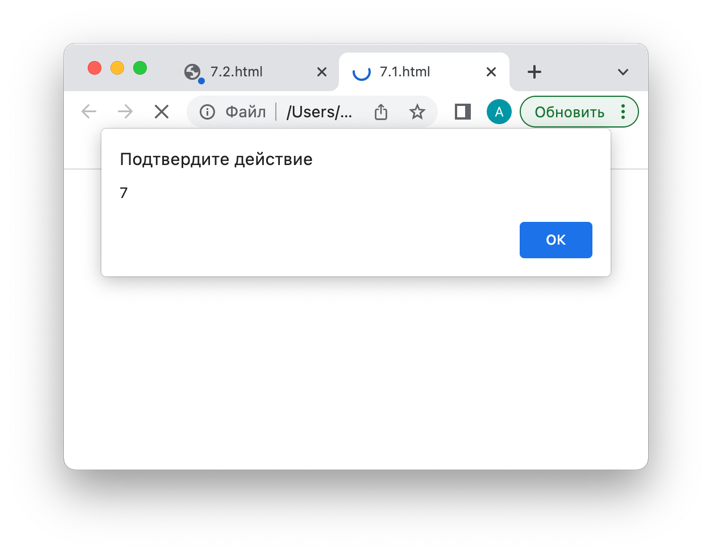
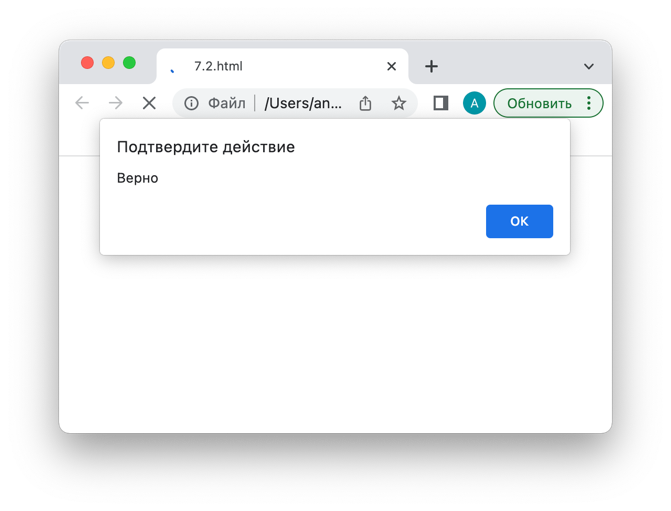
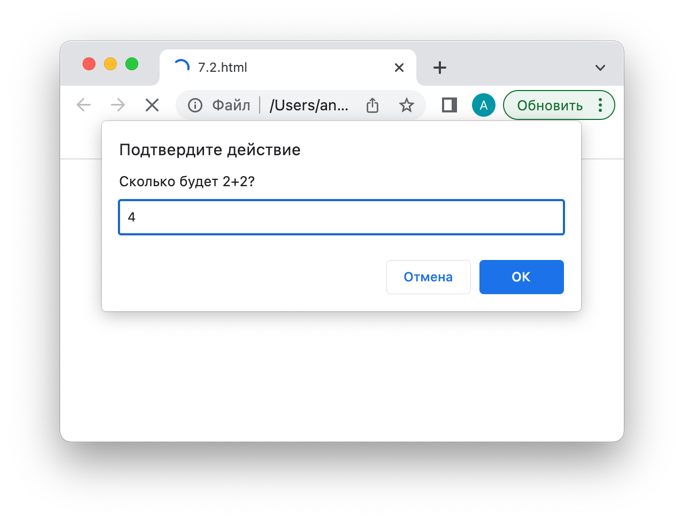
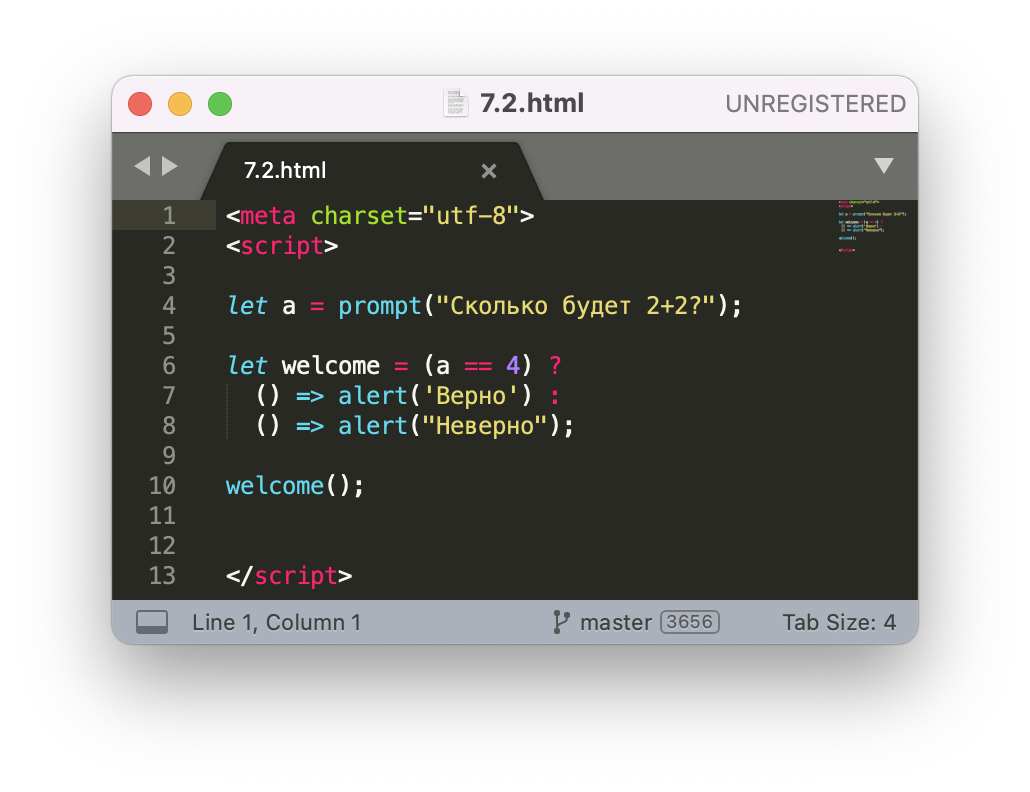
# Стрелочные функции

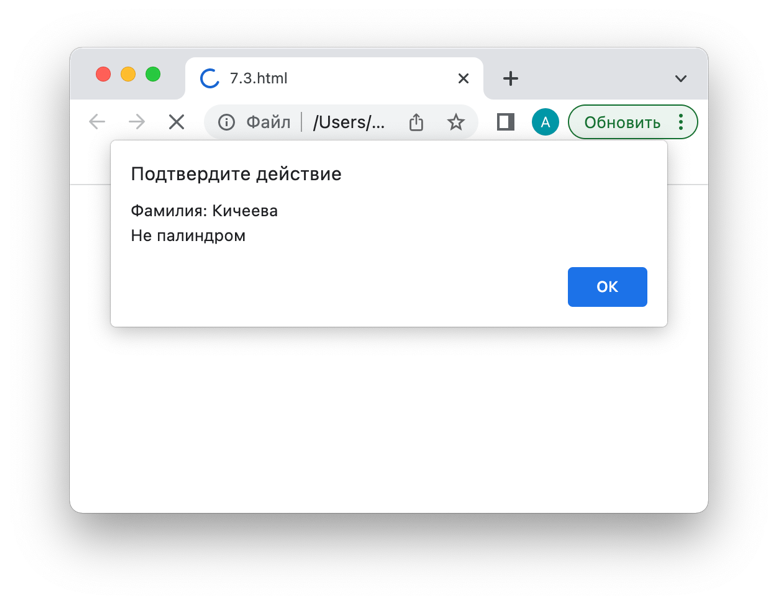
### Задания для выполнения

1. Воспользовавшись методическим указанием реализуйте Калькулятор в виде стрелочных функций.
2. Используя  Function Expression реализуйте проверку условий тестовых заданий.

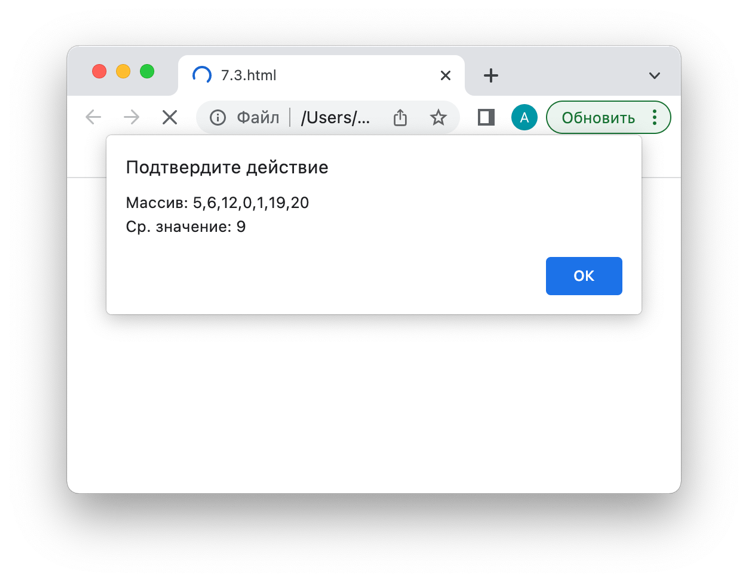




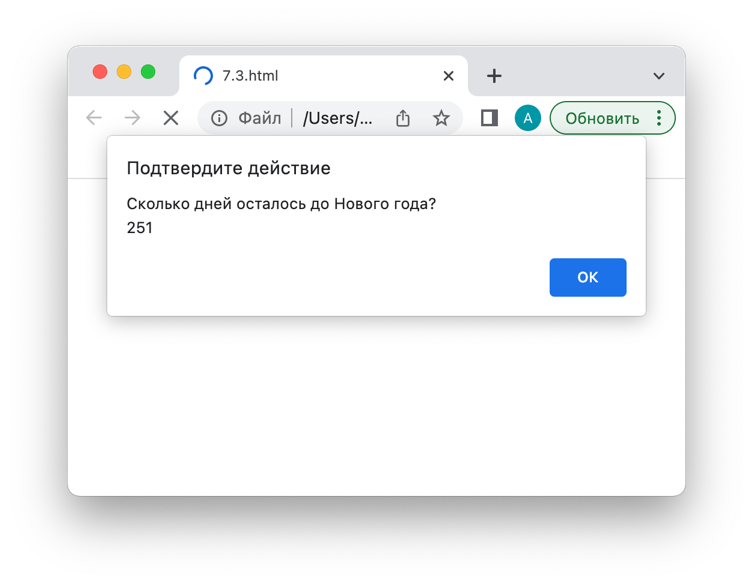
1. Реализуйте в виде стрелочной функции функцию, проверяющую вашу фамилию на полиндром.



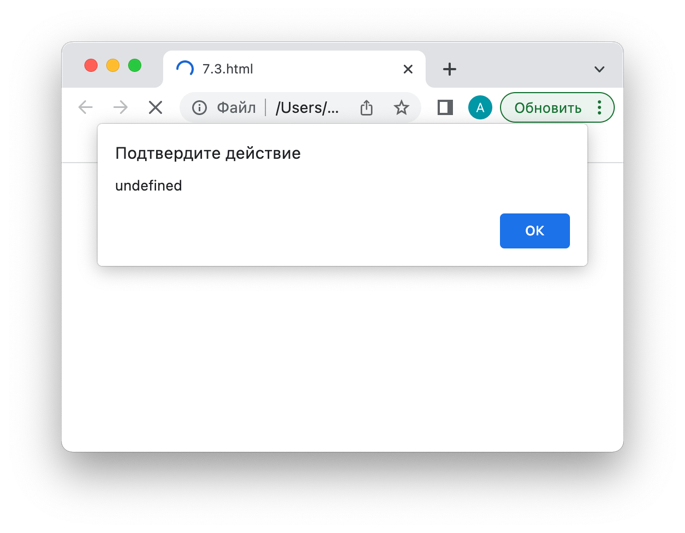
1. Реализуйте в виде стрелочной функции вычисление среднего значения данных в массиве



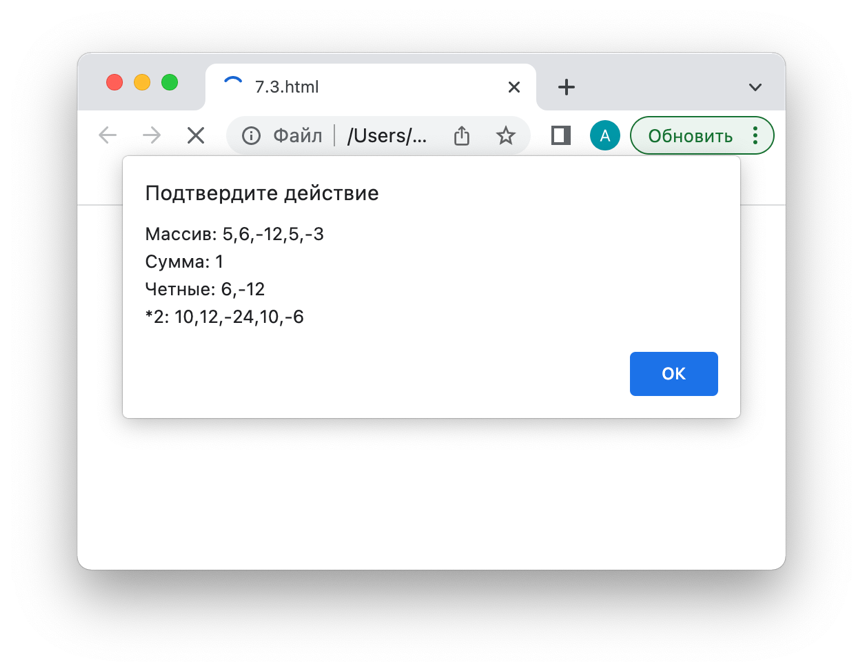
1. В виде стрелочной функции реализуйте функцию, вычисляющую количество дней до нового года.

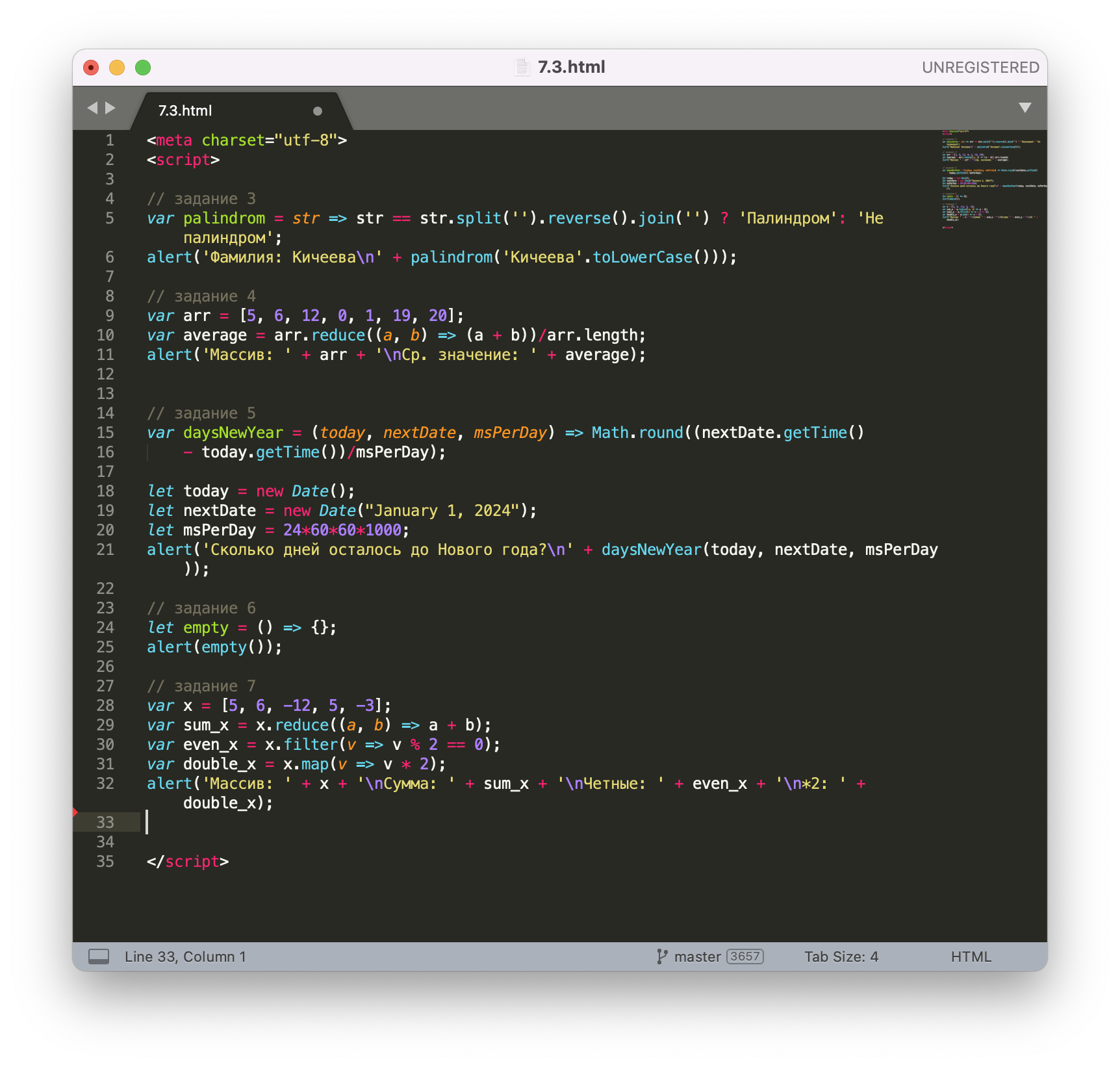


1. Создайте пустую стрелочную функцию возвращает undefined

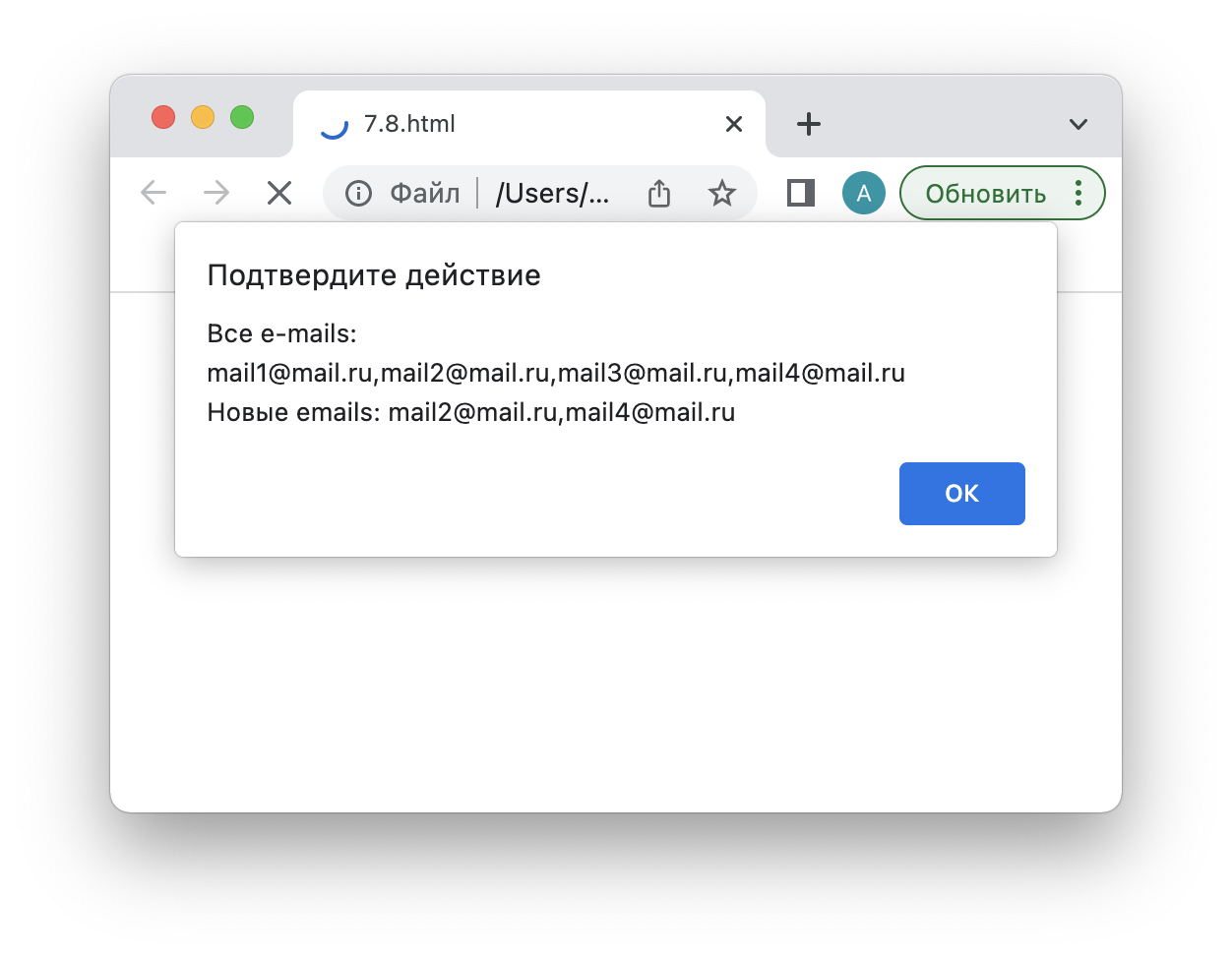


1. Создайте массив и напишите стрелочные функции для него: суммирование всех элементов, выявить все четные, умножить каждый элемент на 2.





1. Создайте массив с e-mail. Организуйте фильтр на странице html по названию почты.



### Контрольные вопросы

1. Есть ли у стрелочной функции метод this?

У стрелочных функций нет this. Если происходит обращение к this, его значение берётся снаружи.

1. Может ли стрелочная функция быть без аргументов?

Да. Если аргументов нет, круглые скобки будут пустыми, но они должны присутствовать:

let sayHi = () => alert("Hello!");

1. Из-за чего появилась стрелочная функция?

Стрелочные функции – это не просто «сокращение», чтобы меньше писать. У них есть ряд других полезных особенностей.

При написании JavaScript-кода часто возникают ситуации, когда нам нужно написать небольшую функцию, которая будет выполнена где-то ещё.

Также можно создать функцию и передать её куда-нибудь. И в таких функциях мы обычно не хотим выходить из текущего контекста. Здесь как раз и полезны стрелочные функции.

1. Объяснить работу алгоритма:
2. // Стрелочные функции без параметров, которые визуально легче разбирать
3. setTimeout( () => {
4. console.log('Я буду раньше');
5. setTimeout( () => {
6. // deeper code
7. console.log('Я буду позже');
8. }, 1);
9. }, 1);