
кафедра

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

должность, уч. степень, звание

подпись, дата

инициалы, фамилия

ОТЧЕТ О ПРАКТИЧЕСКОМ ЗАДАНИИ №5

«Базовые основы языка JavaScript»

по курсу: ИТ-модуль «Основы Frontend-разработки»

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

подпись, дата

инициалы, фамилия

Санкт-Петербург 2023

Цель работы:

Приобретение практических навыков адаптивного дизайна веб-страниц с использованием CSS-Bootstrap.

Задание:

1-я часть задания

1. Создайте веб-страницу, содержащую следующий текст:

Цель работы – приобретение навыков создания для HTML-документа внешних и внутренних скриптов на JavaScript, формирование практических навыков использования операторов и создания функций.

2. Создайте и подключите к HTML-документу внешний скрипт, выводящий сообщение, в котором напишите свою фамилию и номер группы.

3. Создайте внутренний скрипт, выводящий сообщение о том, появились ли у Вас навыки создания веб-страниц с использованием HTML и CSS.

4. Создайте внутренний скрипт, демонстрирующий пример приведения операндов к строковому типу (например, можно вывести сообщение о номере Вашей группы, предварительно задав строковую переменную для текста и числовой тип данных для написания номера).

5. В скрипте, который выводит сообщение в соответствии с 4-ым пунктом задания, преднамеренно допустите ошибку (пропустите одну из скобок, которые должны присутствовать). Приведите в отчете скриншот консоли, с выведенным сообщением об ошибке.

2-я часть задания

Добавьте на созданную в предыдущей части задания страницу:

скрипт, иллюстрирующий работу условного оператора с несколькими ветвями else if...(например, можно составить такой оператор для вывода рекомендаций по питанию в зависимости от значения индекса массы тела либо на какую-то другую тему, подразумевающую несколько вариантов действий);

скрипт, иллюстрирующий применение оператора унарный плюс.

3-я часть задания

Добавьте на страницу скрипт с кодами объявления функций, написанными с использованием трех способов:

Function Declaration,

Function Expression,

Arrow Function.

Каждая из этих функций должна решать одну и ту же достаточно простую задачу, например, определение времени, за которое вы можете пройти весь Невский проспект. При этом надо задать длину проспекта и скорость, с которой вы обычно ходите. В конце функции должно быть выведено сообщение, содержащее соответствующий текст с вычисленным результатом в минутах. Вызовите каждую из этих функций.

Скриншоты, иллюстрирующие выполнение отдельных пунктов задания.

1-я часть задания:

1 пункт:

Основы Frontend-разработки

Цель работы – приобретение навыков создания для HTML-документа внешних и внутренних скриптов на JavaScript, формирование практических навыков использования операторов и создания функций.

Рисунок 1

На рисунке 1 показан вид страницы после всех уведомлений - скриптов. На странице расположена цель работы.

2 пункт:

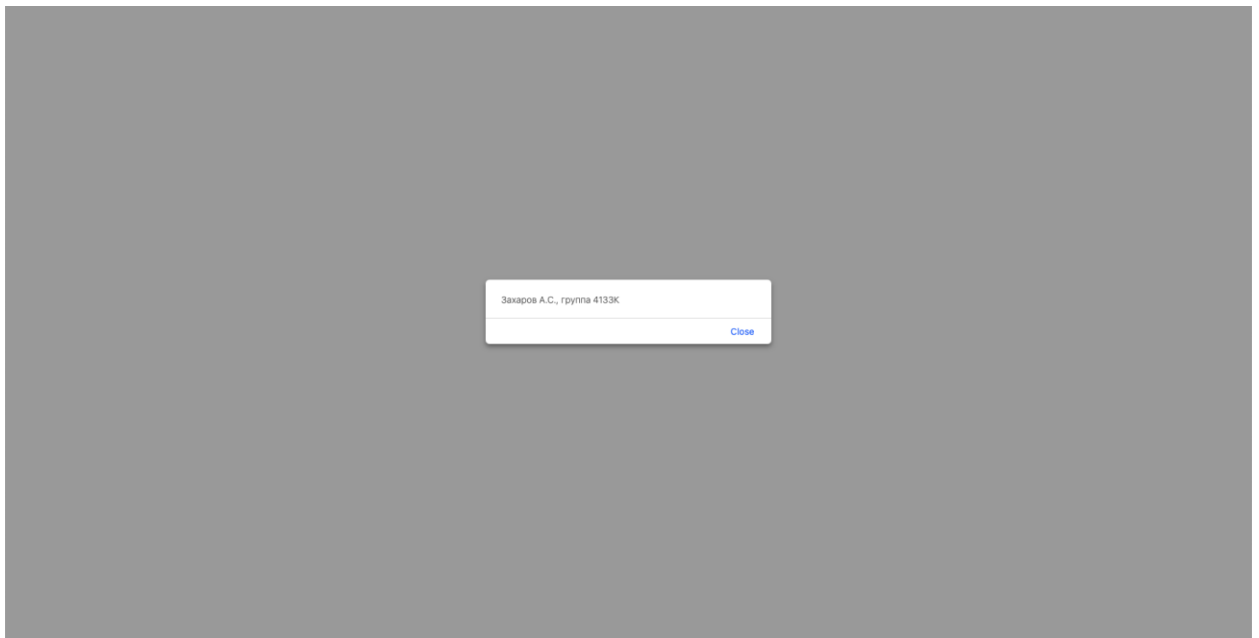


Рисунок 2

На рисунке 2 – вид страницы при открытии. Срабатывает внешний скрипт, выводящий сообщение, в котором написано моё ФИО и номер группы.

Содержимое файла script.js для вывода внешнего скрипта:

```
// Вывод сообщения с фамилией и номером группы  
alert('Захаров А.С., группа 4133К');
```

Подключаем скрипт в HTML файл:

```
<!-- Подключение внешнего скрипта -->  
<script src="script.js"></script>
```

3 пункт:

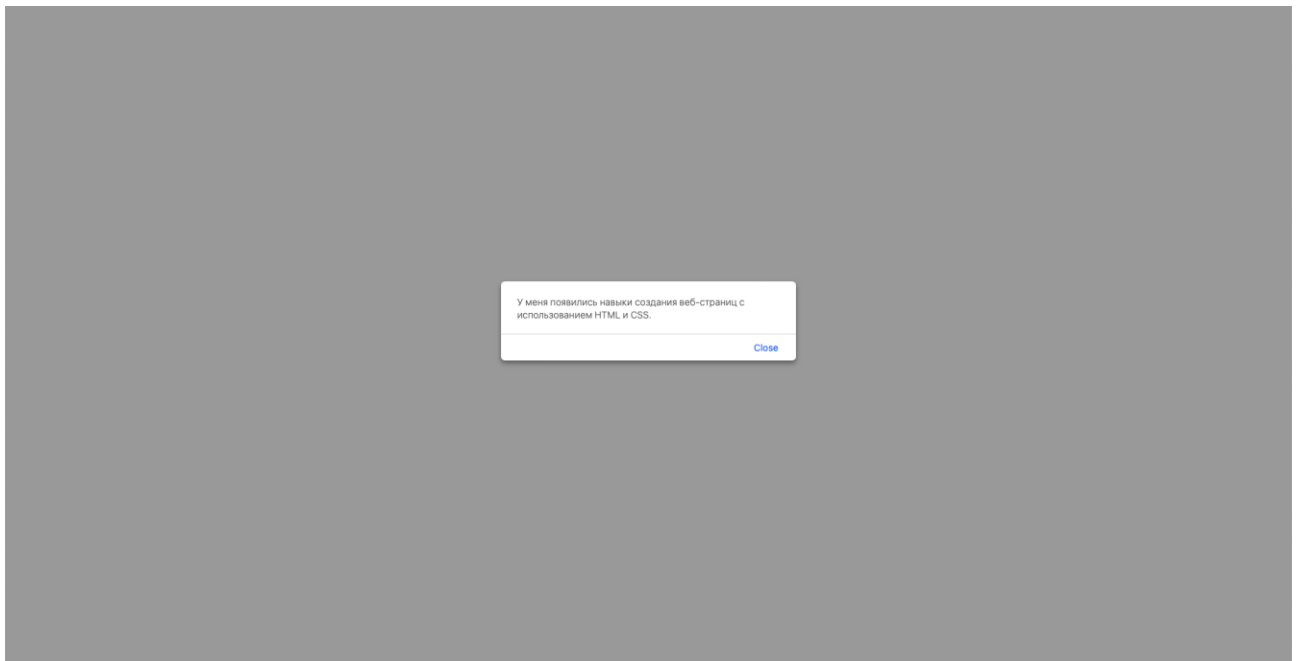


Рисунок 3

На рисунке 3 изображено следующее сообщение, но уже со внутренним скриптом

Работа внутреннего скрипта с уведомлением о приобретении навыков:

```
<!-- Внутренний скрипт -->  
<script>  
    // Вывод сообщения о навыках HTML и CSS  
    alert('У меня появились навыки создания веб-страниц с использованием HTML и CSS.');
```

4 пункт:

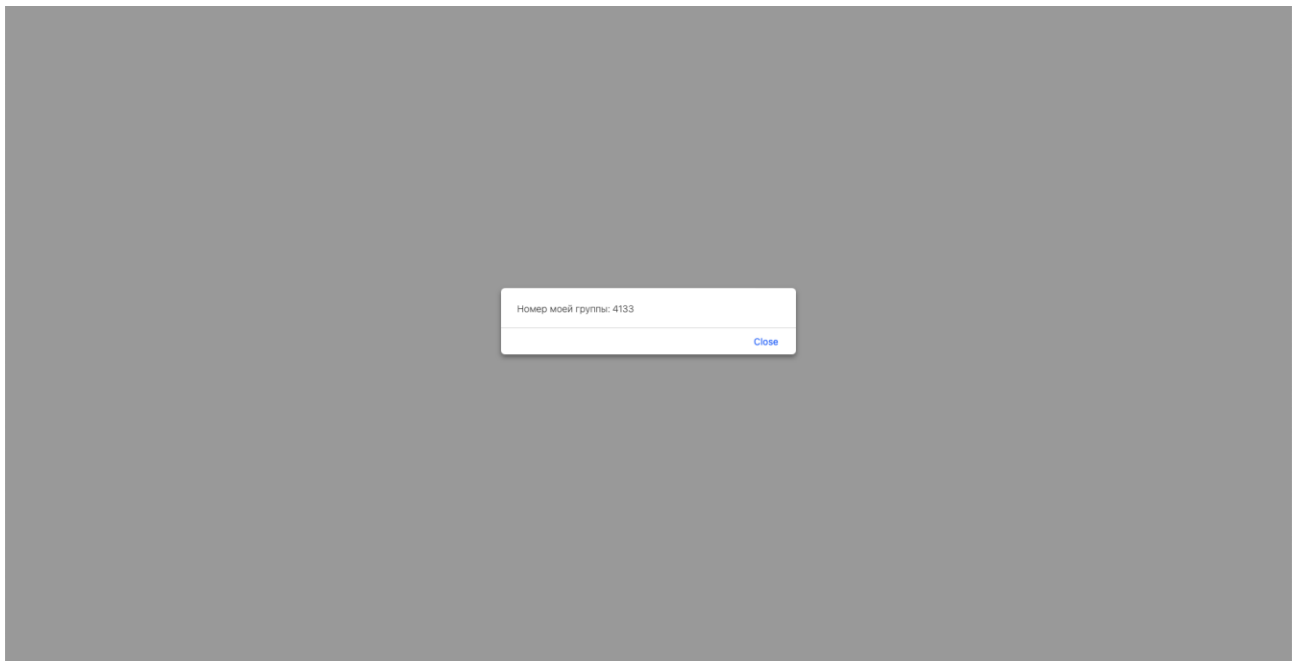


Рисунок 4

На рисунке 4 изображен результат работы внутреннего скрипта, демонстрирующего пример приведения операндов к строковому типу

Работа скрипта:

```
// Демонстрация приведения типов  
var text = 'Номер моей группы: ';  
var groupNumber = 4133; // Здесь должно быть число  
alert(text + groupNumber); // Приведение числа к строке и вывод
```

5 пункт:

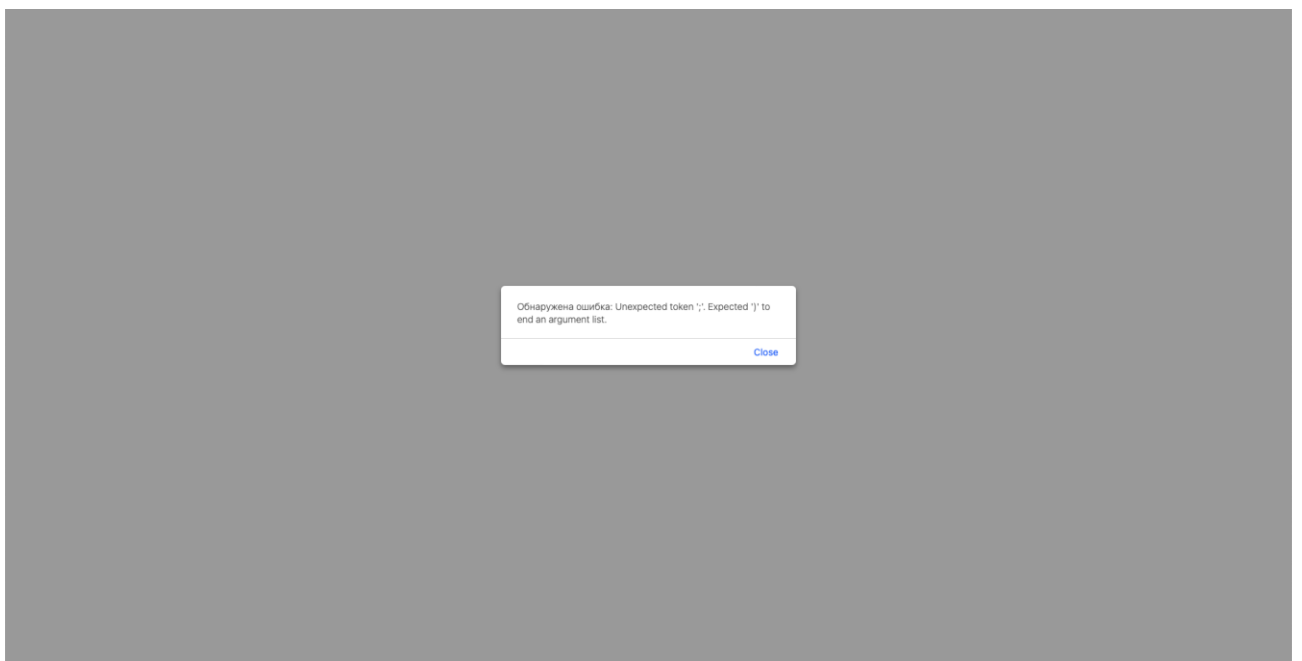
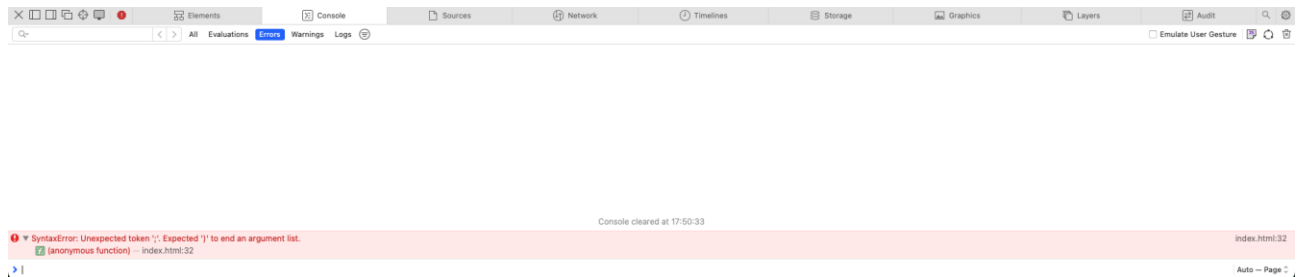


Рисунок 5

На рисунке 5 продемонстрировано как в скрипте, который выводит сообщение в соответствии с 4-ым пунктом задания, преднамеренно допущена ошибка

Скриншот консоли браузера Safari, с выведенным сообщением об ошибке:



Код скрипта с ошибкой:

```
// Скрипт с преднамеренной ошибкой и обработкой ошибки
try {
  // Исполнение кода с синтаксической ошибкой
  eval("alert('Ошибка в следующем сообщении: ' + text + groupNumber;");
} catch(e) {
  // Вывод сообщения об ошибке
  alert('Обнаружена ошибка: ' + e.message);
}
```

2-я часть задания

1 пункт:

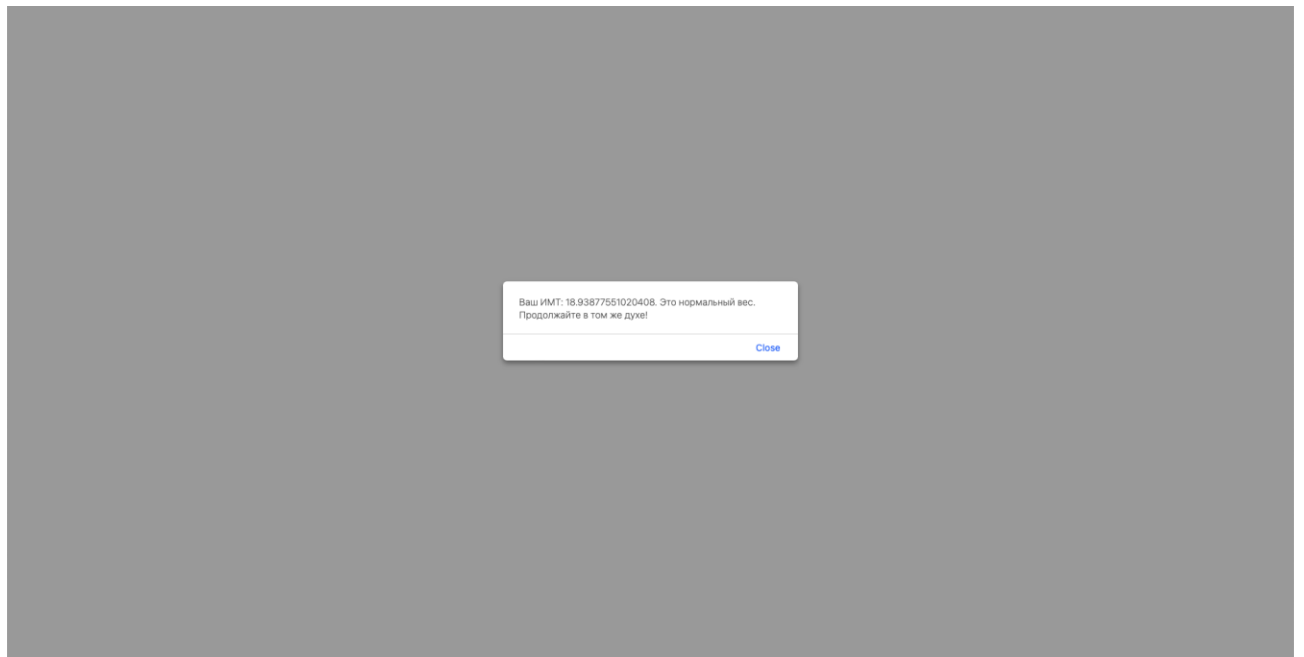


Рисунок 6

На рисунке 6 изображено уведомление с результатом подсчёта от оператора для вывода рекомендаций по питанию. Добавлен на созданную в предыдущей части задания страницу скрипт, иллюстрирующий работу условного оператора с несколькими ветвями else if.

Код с условными операторами

```
<!-- Скрипт с условным оператором -->
<script>
  var weight = 58; // ваш вес в килограммах
  var height = 1.75; // ваш рост в метрах
  var bmi = weight / (height * height); // расчет индекса массы тела

  if (bmi < 18.5) {
    alert("Ваш ИМТ: " + bmi + ". Это меньше нормы. Рекомендуется набор массы.");
  } else if (bmi >= 18.5 && bmi <= 24.9) {
    alert("Ваш ИМТ: " + bmi + ". Это нормальный вес. Продолжайте в том же духе!");
  } else if (bmi >= 25 && bmi <= 29.9) {
    alert("Ваш ИМТ: " + bmi + ". Избыточный вес. Рекомендуется сбалансированное питание и упражнения.");
  } else {
    alert("Ваш ИМТ: " + bmi + ". Ожирение. Рекомендуется обратиться к врачу.");
  }
</script>
```

2 пункт:

Пример ввода: 123

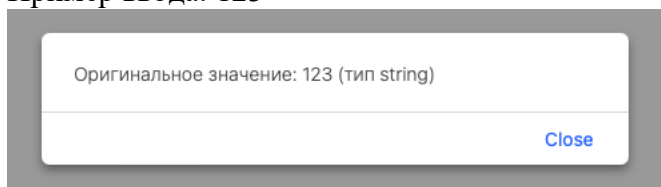


Рисунок 7

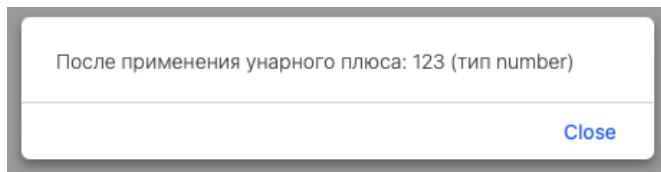


Рисунок 8

Пример ввода: true

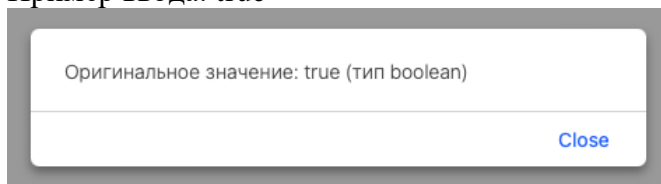


Рисунок 9

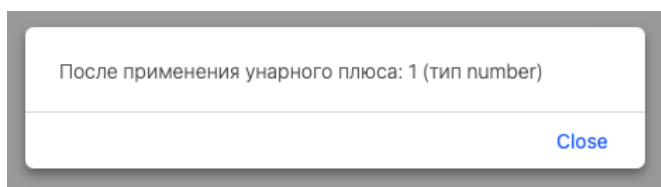


Рисунок 10

Пример ввода: false

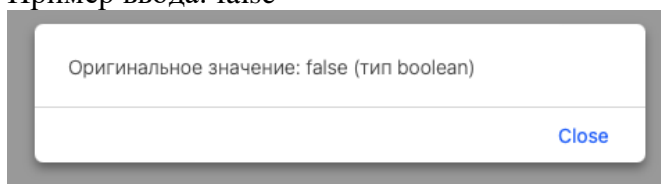


Рисунок 11

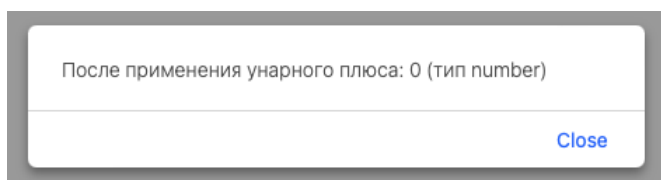


Рисунок 12

На рисунках 7-12 изображён результат работы скрипта, иллюстрирующего применение оператора унарный плюс. В пример приведены тип number и boolean для наглядности.

скрипт, иллюстрирующий применение оператора унарный плюс в коде:

```
<!-- Скрипт, иллюстрирующий применение унарного плюса -->
<script>
  // Преобразование строки, содержащей числовое значение
  var inputString = "123";
  alert("Оригинальное значение: " + inputString + " (тип " + typeof inputString + ")");
  inputString = +inputString; // применяем унарный плюс для преобразования строки в число
  alert("После применения унарного плюса: " + inputString + " (тип " + typeof inputString + ")");

  // Преобразование булевых значений
  var inputTrue = true;
  alert("Оригинальное значение: " + inputTrue + " (тип " + typeof inputTrue + ")");
  inputTrue = +inputTrue; // применяем унарный плюс для преобразования true в число
  alert("После применения унарного плюса: " + inputTrue + " (тип " + typeof inputTrue + ")");

  var inputFalse = false;
  alert("Оригинальное значение: " + inputFalse + " (тип " + typeof inputFalse + ")");
  inputFalse = +inputFalse; // применяем унарный плюс для преобразования false в число
  alert("После применения унарного плюса: " + inputFalse + " (тип " + typeof inputFalse + ")");
</script>
```

3-я часть задания:

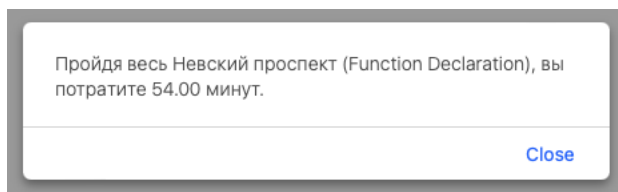


Рисунок 13

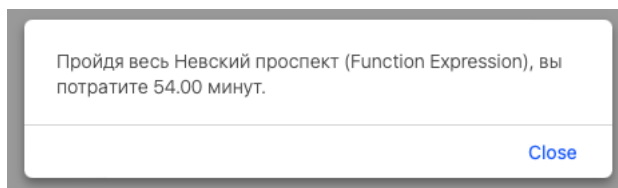


Рисунок 14

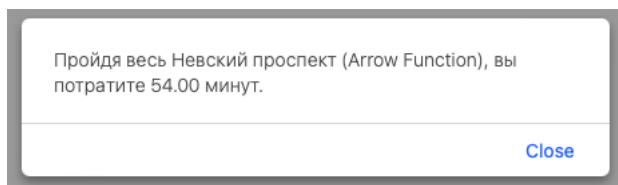


Рисунок 15

На рисунках 13-15 изображены результаты работы трёх функций (Function Declaration, Function Expression, Arrow Function). Каждая из этих функций решает одну и ту же достаточно простую задачу.

Код скрипта с объявлением функций, написанными с использованием трех способов: Function Declaration, Function Expression, Arrow Function:

```
<!-- Скрипты объявления функций для третьей части задания -->
<script>
  // Function Declaration
  function calculateWalkingTimeDecl(length, speed) {
    var time = (length / speed) * 60; // Преобразуем часы в минуты
    return 'Пройдя весь Невский проспект (Function Declaration), вы потратите ' + time.toFixed(2) + ' минут.';
  }

  // Function Expression
  var calculateWalkingTimeExpr = function(length, speed) {
    var time = (length / speed) * 60; // Преобразуем часы в минуты
    return 'Пройдя весь Невский проспект (Function Expression), вы потратите ' + time.toFixed(2) + ' минут.';
  };

  // Arrow Function
  var calculateWalkingTimeArrow = (length, speed) => {
    var time = (length / speed) * 60; // Преобразуем часы в минуты
    return 'Пройдя весь Невский проспект (Arrow Function), вы потратите ' + time.toFixed(2) + ' минут.';
  };

  // Параметры для функций: длина Невского проспекта и средняя скорость ходьбы
  var lengthOfNevsky = 4.5; // Допустим, длина Невского проспекта 4.5 км
  var averageWalkingSpeed = 5; // Средняя скорость ходьбы в км/ч

  // Вызов функций
  alert(calculateWalkingTimeDecl(lengthOfNevsky, averageWalkingSpeed));
  alert(calculateWalkingTimeExpr(lengthOfNevsky, averageWalkingSpeed));
  alert(calculateWalkingTimeArrow(lengthOfNevsky, averageWalkingSpeed));
</script>
```

Вывод.

В ходе выполнения данной работы были успешно реализованы ключевые задачи, связанные с основами Frontend-разработки, в частности, с применением языка программирования JavaScript. Работа охватила создание внешних и внутренних скриптов, применение операторов JavaScript, объявление функций разными способами, и демонстрацию приведения типов.

Таким образом, цель работы, состоящая в приобретении практических навыков создания скриптов, использования операторов и создания функций для HTML-документа, была достигнута. Эти навыки лежат в основе разработки динамических веб-страниц и являются критически важными для любого Frontend-разработчика.