

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ГОРОДА МОСКВЫ ДОПОЛНИТЕЛЬНОГО  
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ ЦЕНТР  
ПРОФЕССИОНАЛЬНЫХ КВАЛИФИКАЦИЙ И СОДЕЙСТВИЯ  
ТРУДОУСТРОЙСТВУ «ПРОФЕССИОНАЛ»**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к итоговой аттестационной работе на тему  
**«Разработка web-ресурса с использованием  
технологий HTML , CSS, JavaScript, jQuery»**  
**(на примере web-ресурса: «Сайт изготовителя ювелирных изделий»)**

слушателя Овчинникова А.И. группы № 1103  
программы профессиональной переподготовки «Frontend-разработка»

**Москва 2021**

# **1. ПОСТАНОВКА ЗАДАЧИ И ПЛАН РАБОТЫ**

## **1.1 Цель проекта:**

Создание тематического веб-ресурса «Клавиатурный тренажёр», посредством использования языка гипертекстовой разметки HTML, каскадных таблиц стилей CSS, языка программирования JavaScript и/или его библиотек.

## **1.2 План работы:**

1. Выбор тематики проекта.
2. Составление структуры и функциональной части веб-ресурса.
3. Написание программного кода и запуск веб-приложения.
4. Тестирование веб-приложения: выявление существенных недостатков работы веб-приложения с последующим устранением.
5. Оформление пояснительной записки к итоговой аттестационной работе.

# **2. ОСНОВНАЯ ЧАСТЬ**

## **2.1 Назначения веб-ресурса, описание его функционала:**

Данная работа представлена в виде веб-ресурса (веб-сайт). Тема «Клавиатурный тренажёр». Предназначен для тренировки скорости печати слепым методом.

### **Этапы разработки:**

#### **Первый этап HTML-вёрстка:**

- Создание структуры сайта (Кнопка с ссылкой внутри выполняющая функцию открытия вступительного модального окна, div-ы для ориентации

основных полей интерфейса, div-ы для каждой кнопки клавиатуры, div-ы для формирования вступительного модального окна);

- Подключение стилей, js-скриптов;
- **Второй этап CSS-таблица каскадных стилей:**
- Написание общих стилей
- **Третий этап JavaScript:**
- Назначение переменных **soundErr**(sound Error), **soundCor**(sound Correct), **startS**(startSound) для привязки соответствующих звуковых файлов действия.
- Написание функции **rKey**(random key), вызывающую случайный div-клавишу следующую для нажатия
- Написание функций **InputFocus** и **InputOutFocus** для визуализации фокуса на поле ввода.
- Написание функции **InputValue** для отображения нажатой клавиши в поле вывода.
- Написание функции **init** - в ней назначены слушатели «Keydown», «Keypress» и «Keydown» которые вызывают соответствующие функции.
- Основная и самая объёмная функция **KeypressHandler** вызывается событием «Keypress».
- Написание функции **ClearClass** удаляет с клавиш клавиатуры(Поле 4) выделение цветами.
- В конце js-файла идут 10 строк кода, которые останавливают анимацию библиотеки «Animate», до нажатия клавиши **Enter**, которая закрывает модальное окно.

## 2.2 Описание функционала главной страницы (index.html)

### 1. Модальное окно(рорир).

При загрузке страницы появляется модальное окно с инструкцией к тренажеру, которое останавливает анимацию появления интерфейса. Окно представляет собой сочетание вложенных друг в друга дивов. Старшему родительскому диву присвоено свойство `visibility: hidden`, которое исполняется тегом `<a>` в случае получения псевдокласса `target` при нажатии. В свою очередь назначена функция `javascript`, которая имитирует клик по ссылке нажатием клавиши `Enter`.

Скрыть окно можно только с помощью нажатия клавиши **Enter**.

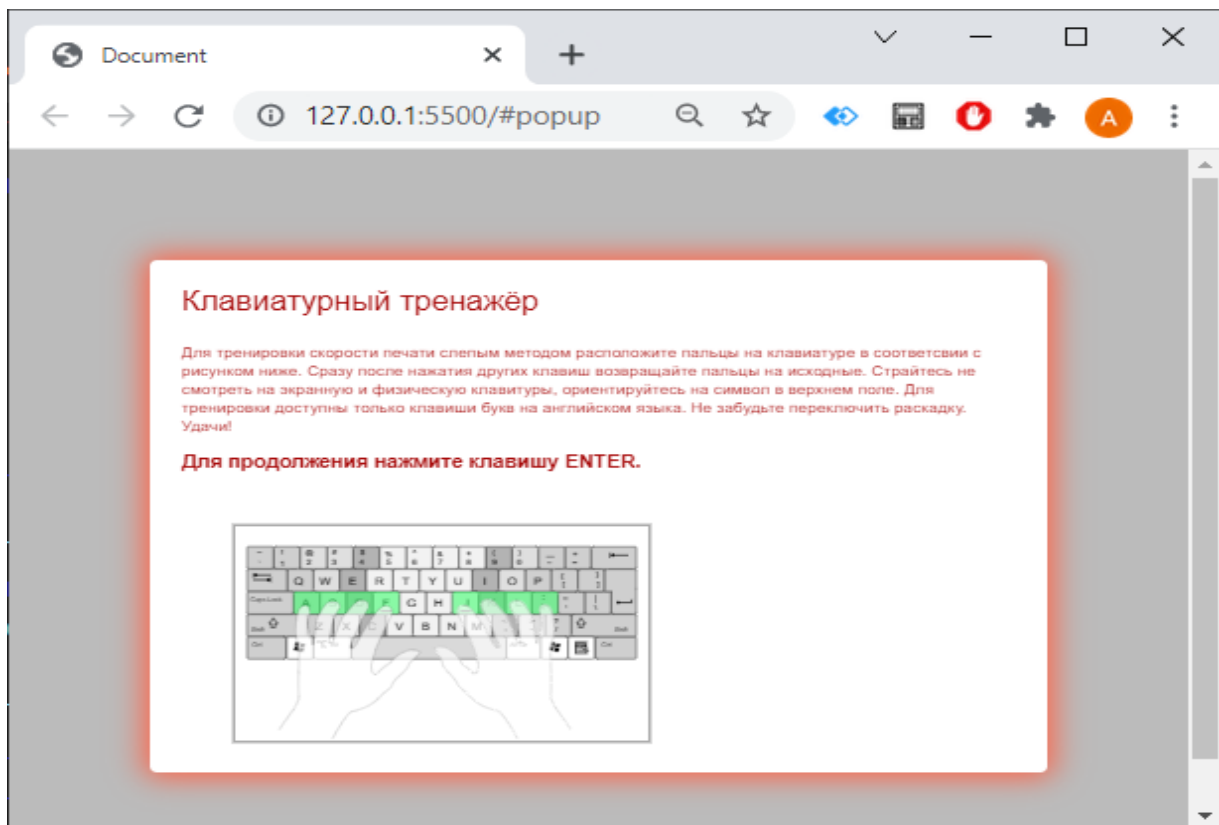
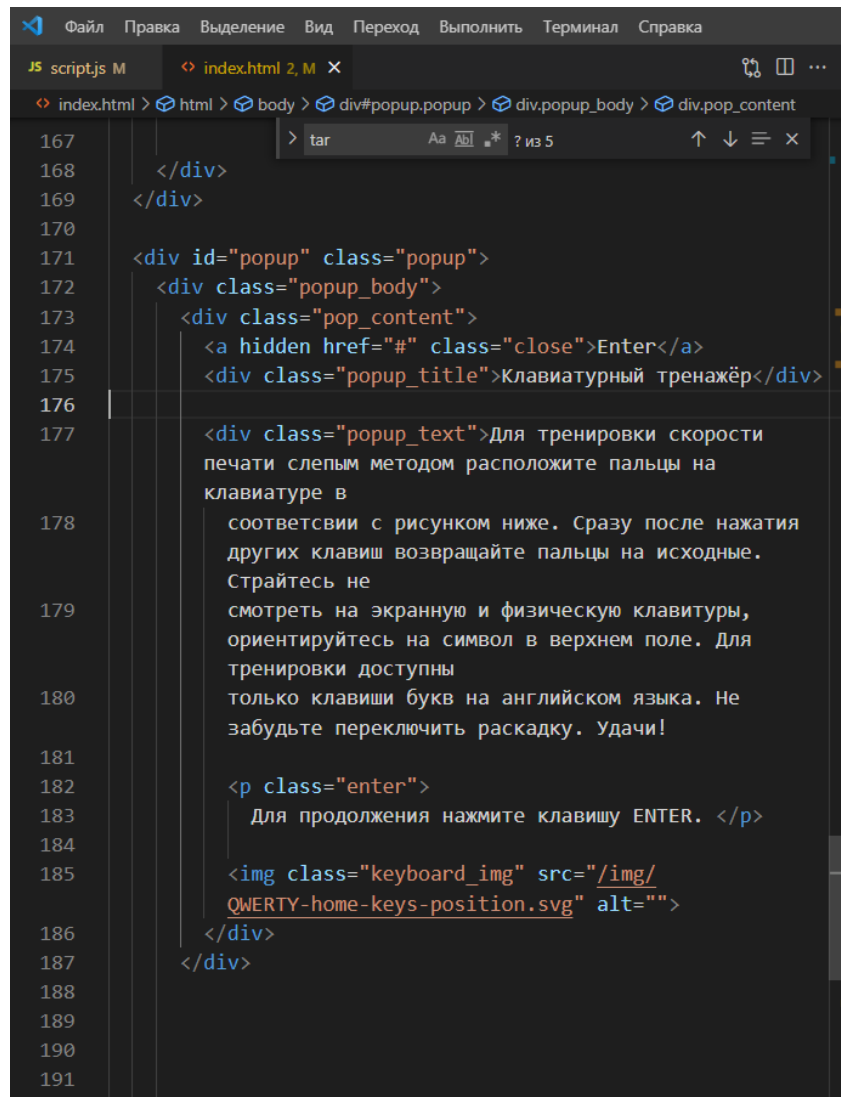


Рисунок 1 – раздел Рорир.



```
167     </div>
168 </div>
169 </div>
170
171 <div id="popup" class="popup">
172   <div class="popup_body">
173     <div class="pop_content">
174       <a hidden href="#" class="close">Enter</a>
175       <div class="popup_title">Клавиатурный тренажёр</div>
176
177       <div class="popup_text">Для тренировки скорости
178         печати слепым методом расположите пальцы на
179         клавиатуре в
180         соответствии с рисунком ниже. Сразу после нажатия
181         других клавиш возвращайте пальцы на исходные.
182         Странно не
183         смотреть на экранную и физическую клавиатуры,
184         ориентируйтесь на символ в верхнем поле. Для
185         тренировки доступны
186         только клавиши букв на английском языке. Не
187         забудьте переключить раскладку. Удачи!
188
189       <p class="enter">
190         Для продолжения нажмите клавишу ENTER. </p>
191       
193     </div>
194   </div>
195 </div>
```

Рисунок 2 –Html верстка Роруп

## 2.3 Основной интерфейс.

Интерфейс тренажера представлен четырьмя полями(сверху вниз):

1. Поле статистики это div с классами «field» и «stats\_bar», содержащий в себе 3 тега <h2> с вложенными в каждый из них тегами img с привязанными svg изображениями. Тегам <h2> присвоены псевдо-элементы для позиционирования соответствующих выводимых значений.
2. Поле вывода следующей для нажатия буквы - это div с классами «field» и «output field» содержащий в себе тег <p>. Полю присвоена функция вывода буквы клавиатуры для эстетической визуализации - в поле не будет копиться

вводимый текст, предыдущий символ будет заменяться следующим и располагаться в центре поля.

3. Поле ввода букв это div с вложенным тегом `<textarea>`. Тег `<textarea>` дополнен атрибутом `placeholder` с декоративным текстом «КЛИК!», целью которого является привлечь внимание пользователя и побудить его кликнуть указателем мыши на поле, чтобы начать ввод букв.
4. Клавиатура - это множество div-ов для каждой из клавиш. Div-ы имеют класс по названию буквы, например «Key\_space», «Key\_s», «Key\_5» и содержат текст (TextContent), который выводится в поле 2 и означает букву, которую необходимо нажать следующей.

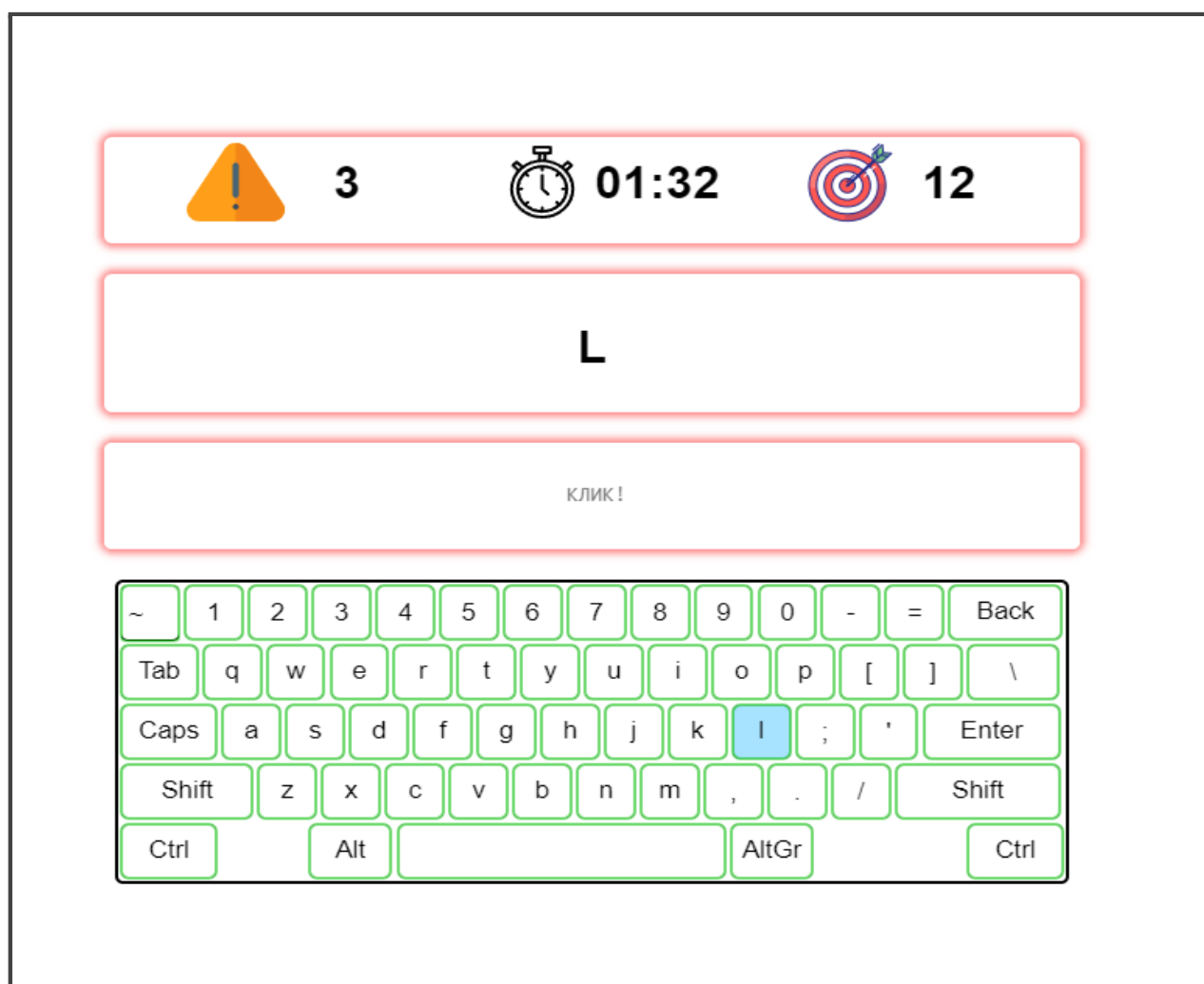


Рисунок 3 – Основной интерфейс

```
JS script.js M index.html 2, M
index.html > html > body > div.flex-div.animate__animated.animate__fadeInUpBig
BUTTON>
30
31 <div class="field stats_bar animate__animated
32 animate__fadeInDownBig">
33   <h2 class="stats wrongnum">
35   <h>
36     <h2 class="stats timer">
38     <h>
39       <h2 id="targetdata" class="stats targetdata">
41       <h>
42     </div>
43   </div>
44   <div class="field output_field animate__animated
45   animate__fadeInBottomLeft">
46     <p class="output"></p>
47   </div>
48   <div class="field animate__animated animate__zoomIn">
49     <textarea placeholder="КЛИК!" cols="80" rows="6"
50     class="input input_field "></textarea></div>
51   <div class="flex-div animate__animated animate__fadeInUpBig">
52     <div class="kb">
53       <div class="key key_tilde">
54         <div class="key_top">~</div>
55       </div>
56       <div class="key key_1">
57         <div class="key_bottom">1</div>
58       </div>
59     </div>
60   </div>
61   <div class="flex-div animate__animated animate__fadeInUpBig">
```

Рисунок 4 - Код полей 1-3 и фрагмент кода поля «Клавиатура».

### 3. Написание программного кода и запуск веб-приложения.

#### 3.1 Озвучка интерфейса.

Назначение переменных **soundErr**(sound Error), **soundCor**(sound Correct), **startS**(startSound) для привязки соответствующих звуковых файлов действия. Почти все действия в веб-приложении имеют озвучку. Звуковые файлы подобраны таким образом, чтобы соответствовать выполняемому действию.

```
11 let soundErr = new Audio;
12 soundErr.src = "../sounds/sound_error.mp3";
13
14
15 let soundCor = new Audio
16 soundCor.src = "../sounds/sound_correct.mp3";
17
18 let starts = new Audio();
19 starts.src = "../sounds/start1.mp3";
20
```

Рисунок 5 – Фрагмент кода файлов озвучки.

Переменная **soundErr** озвучивает нажатие неправильной клавиши - отличной от ожидаемой кодом, **soundCor** соответственно - правильной. **StartS** озвучивает нажатие клавиши **Enter**, которое ознаменовывает запуск приложения. Функции **SoundErr.play** и **soundCor.play** помещены в функцию **KeypressHandler**.



### 3.2 Написание функции **rKey(random key)**.

Функция **rKey(random key)** вызываемая нажатием клавиши и обработчиком события «keypress» случайным образом определяет следующую клавишу для нажатия. **Arrkeys(array keys)** - массив в котором содержатся все возможные для нажатия клавиши. В функции назначена переменная **nkey** (next key) - это один из элементов массива **arrkeys** обработанный функциями **Math.floor** и **Math.random**. Первая округляет число до целого, вторая выбирает случайное число из 26-и - по количеству доступных клавиш.

Далее прописан цикл **for** который проходит по **arrkeys**-массиву и удаляет класс «**next\_key**» у клавиши, которая была нажата. Класс «**next\_key**» подсвечивает голубым цветом клавишу, которую необходимо нажать следующей.

### 3.3 Написание функций **InputFocus** и **InputOutFocus** .

Для визуализации фокуса на поле ввода написаны функции **InputFocus** и **InputOutFocus**. Функции сами по себе просты и назначение их больше эстетическое, чем функциональное. Функция **InputFocus** удаляет текст атрибута сразу после клика, без функции текст удаляется только после начала ввода. Также прописана функция **inputValue**, которая отображает в поле ввода <textarea> нажатую клавишу. **InputOutFocus** очищает поле ввода и возвращает текст атрибута placeholder в случае исчезновения фокуса.

Следующая по коду важная функция **KeypressHandler**. В функции назначена переменная **z**, в которую записывается **KeyCode** события «keypress». **KeyCode** это уникальный номер события клавиши. Всем работающим клавишам

в приложении присвоен id в html разметке, которому соответствует **KeyCode** каждой клавиши. Далее назначена переменная - **downed**. **Downed** обращается к документу и находит элемент в массиве **arrkeys** с id номером, соответствующим id номеру события «**keypress**». Функция **KeypressHandler** обрабатывает полученные данные переменных **z** и **Downed** и если нажатая клавиша не соответствует ожидаемой кодом (случайной букве в переменной **nkey**(next key), назначенной функцией (random key) из массива **arrkeys**(array keys)) функция **KeypressHandler** добавляет к соответствующему div-у класс «**wrong\_key**». Таким образом неверно нажатая клавиша на физической клавиатуре подсвечивается красным цветом на экранной клавиатуре.



**Рисунок 6** – Фрагмент клавиатуры.

Если ( **if** ) нажатая клавиша соответствует ожидаемой, **KeypressHandler** присваивает ожидаемой клавише класс «**correct**», запускает функцию **soundCor.play()**; , которая проигрывает звук правильно нажатой клавиши. Ожидаемая клавиша, которая была подсвечена голубым цветом классом «**next\_key**» подсвечивается зелёным цветом. Далее вызывается функция **rKey**.

Иначе ( **else** ) как было написано выше неправильно нажатая клавиша подсвечивается красным цветом проигрывается соответствующий звук функцией **soundErr.play()**; , далее вызывается функция **rKey**. Также сразу после любого из событий **if else** вызывается функция **clearClass**.

```

94
95
96 function KeypressHandler (event) {
97
98     z = event.keyCode
99     downed = document.getElementById(z)
100
101     if (OutPutField.textContent == event.key) {
102         clearClass();
103         arrkeys[nkey].classList.add("correct");
104         soundCor.play();
105         rKey();
106     } else {
107         clearClass();
108         downed.classList.add("wrong_key")
109         soundErr.play();
110         rKey();
111     }
112
113
114
115 }

```

Рисунок 7 – Функция KeypressHandler.

### 3.3 Функция clearClass .

Данная функция удаляет подсветку с клавиш после нажатия. Счётчик цикла **for** проходит по массиву **arrkeys** с добавленным методом **classList.remove** и классами «**correct**» и «**wrong\_key**». Таким образом при нажатии следующей клавиши та или иная подсветка предыдущей клавиши будет удалена.

```

129 function clearClass (){
130
131     for(i = 0; i < arrkeys.length; i++ ){
132         arrkeys[i].classList.remove("correct");
133         arrkeys[i].classList.remove("wrong_key")
134     }
135 }
136
137

```

Рисунок 8 – Функция clearClass.

### 3.4 Остальной код.

К веб-приложению подключена библиотека анимаций **Animate.css**. Обычно проигрывание анимаций начинается сразу после загрузки страницы, но особенностью данного проекта является то, что воспроизведение анимации остановлено свойством класса «**animate\_\_animated**» в **css** файле. Изначально с помощью метода **querySelector** в документе находится тег **<a>**, который верстался как ссылка-кнопка закрывающая модальное окно, тегу назначается слушатель события “клик мышки”. Далее назначается слушатель события «**Keypress**», после него функция **keyBoardEvent**, задача которой провести проверку, если **keyCode** события равен числу 13 (КейКод клавиши **enter**), то с помощью метода **querySelector** имитируется щелчок левой кнопки мыши по тегу **<a>**. Также данное событие запускает анимацию **Animate.css**. С помощью библиотеки **jQuery** создано обращение к свойству **animation-play-state** класса «**animate\_\_animated**» с целью сменить значение **paused** на **running**. Таким образом нажатием клавиши **Enter** запускается функция **starts.play()**, имитируется щелчок мыши по тегу закрывающему попап, начинается анимация.

```
//_____функция закрытия попапа по клавише enter(также останавливает анимацию)_____  
document.querySelector("a").addEventListener("click", () => {});{  
  document.querySelector("a").click();  
  
  document.addEventListener("keypress", keyBoardEvent);  
  
  function keyBoardEvent(event) {  
    if (event.keyCode == 13 ) {  
      document.querySelector(".close").click();  
      $(".animate__animated").css("animation-play-state", "running");  
      starts.play();  
    }  
  }  
}
```

### **Место размещения исходного программного кода.**

Ссылка на репозиторий GitHub с расположенным  
исходным кодом приложения: <https://github.com/AndreyOvchinnik0v>

GitHub: <https://andreyovchinnik0v.github.io/FINAL-PROJECT/>

### **Тестирование веб-приложения**

Выявление существенных недостатков работы веб-приложения с  
последующим устранением.

Отработка тех или иных функций js.

Проверка сайта на адаптивность ко всем устройствам (компьютер, планшет,  
телефон).

## СПИСОК ЛИТЕРАТУРЫ

1. «HTML и CSS. Разработка и создание веб-сайтов» (Джон Даккет)
2. «JavaScript и jQuery. Интерактивная разработка» (Джон Даккет)
3. «Новая большая книга CSS (2016)» (Дэвид Макфарланд)
4. Интернет-источник <https://itchief.ru/> (ИТ ШЕФ)
5. Интернет-источник <https://webref.ru/> (Web Reference)
6. Интернет-источник <https://www.w3schools.com/> (w3schools)
7. Видео-лекции с курса по frontend-разработке