Лабораторная работа по JavaScript

ГЛАВНАЯ ЦЕЛЬ

Вашему ресторану-бистро Goose Bistro нужен способ увлечь посетителей и рассказать им о разнообразии ваших блюд. Вы придумали замечательную стратегию — игру «Угадай слово», но с блюдами, которые вы предлагаете, и их фотографиями из реальной жизни. Для этого вам на помощь придёт ваш друг JavaScript.

ИНСТРУКЦИЯ

1. Скопируйте папку с предыдущей лабораторной работой и назовите её `lab_js`, затем откройте её с помощью `vscode`.

2. Создайте новый файл в папке `lab_js` и назовите его `guess_word.html`.

3. Скопируйте следующий текст разметки `html` в `guess_word.html`.

HTML

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Моя страница</title>
    <link rel="stylesheet" href="main.css" />
  </head>
  <body>
    <h1>Добро пожаловать в игру «Угадай блюдо из бистро Гуся»</h1>
    <div id="letters"></div>
    <hr />
    <div>
      <input
        type="text"
        maxlength="1"
        placeholder="введите букву"
        size="2"
        id="userInput"
      />
      <button id="guess">guess</button>
    </div>
    <script src="guess_game.js"></script>
  </body>
</html>
```

Посмотрите на страницу в браузере. Вы должны увидеть что-то вроде следующее:

2. Создайте новый файл в папке `lab_js` и назовите его `guess_game.js`.

3. В файл `guess_game.js` добавьте функцию для генерации случайного целого числа.

JS JavaScript

```
function getRandomIntInclusive(min, max) {
  // Убедитесь, что min и max — целые числа.
  min = Math.ceil(min);
  max = Math.floor(max);
  // Сгенерируйте случайное целое число в диапазоне [min, max].
  return Math.floor(Math.random() * (max - min + 1)) + min;
}
```

Эта функция поможет нам выбрать случайное слово (блюдо) из массива заранее заданных слов.

4. Создайте массив, в котором будут храниться слова и из которого будут выбираться слова.

JS JavaScript

```
const words = ["pizza", "tapenade", "pasta", "borsh"];
```

5. Выберите случайное слово и сохраните его в переменной `word`

JS JavaScript

```
randInt = getRandomIntInclusive(0, words.length - 1);
var word = words[randInt];
```

6. Получите элемент `div`, содержащий элементы `span`, содержащие буквы слова. Обратите внимание, что его идентификатор — `"letters"`.

JS JavaScript

```
var letters = document.getElementById("letters");
```

Этот элемент (`div`) будет полезен для генерации начальных элементов `span` и изменения буквы, отображаемой на странице.

7. Получите текстовый элемент `input`, в который пользователь вводит букву, которая, по его мнению, является частью слова.

JS JavaScript

```
var userInput = document.getElementById("userInput");
```

Этот элемент будет полезен для получения текста (буквы), написанного пользователем, и проверки, является ли он частью слова.

8. Объявите массив, который будет служить заполнителем для элементов `span`, чтобы мы могли их изменять.

JS JavaScript

```
const spanArray = [];
```

9. Объявите массив, который будет служить заполнителем для текущей буквы, угаданной пользователем, чтобы мы могли сохранить её для будущих сравнений с угаданным словом.

JS JavaScript

```
const userArray = [];
```

10. Используйте цикл для генерации элементов `span`, в которых будут отображаться буквы слова.

JS JavaScript

```javascript
for (i = 0; i < word.length; i++) {
  var initialText = document.createTextNode(" _____ ");
  spanArray[i] = document.createElement("span");
  spanArray[i].setAttribute("id", "letter" + i);
  spanArray[i].appendChild(initialText);
  letters.appendChild(spanArray[i]);
  userArray[i] = " _____ ";
}
```

- метод `spanArray[i] = document.createElement("span");` создаёт элементы `span` и сохраняет их в массиве `spanArray`.
- `var initialText = document.createTextNode(" _____ ");` создаёт начальный текст для заполнения элементов `span`.
- `spanArray[i].appendChild(initialText);` устанавливает текст элемента `span`.
- `letters.appendChild(spanArray[i]);` добавляет элементы `span` к элементу `div`.
- `spanArray[i].setAttribute("id", "letter" + i);` устанавливает идентификатор текущего элемента spam.
- `userArray[i] = " _____ ";` устанавливает начальный текст в userArray.

11. Создайте следующую функцию (`fillUserWord`) и поместите её после функции `getRandomIntInclusive`.

JS JavaScript

```javascript
function fillUserWord(userArray, userInput, word) {
  for (i = 0; i < word.length; i++) {
    if (word[i] === userInput) {
      userArray[i] = userInput;
    }
  }
}
```

Эта функция сравнивает введённую пользователем букву с буквами в загаданном слове и подставляет загаданную букву на

место.

12. Создайте следующую функцию (`mirrorUserWord`) и поместите её после функции `fillUserWord`.

JS JavaScript

```
function mirrorUserWord(userArray) {
  for (i = 0; i < word.length; i++) {
    spanArray[i].innerHTML = userArray[i];
  }
}
```

Эта функция записывает содержимое массива userArray в элементы `span`, зеркально отображая или обновляя содержимое букв на странице.

13. Получите элемент кнопки, чтобы можно было сканировать, прослушивать или опрашивать событие нажатия и обновлять игру. Добавьте следующую строку кода в конец текущего документа.

JS JavaScript

```
var guessButton = document.getElementById("guess");
```

14. После предыдущей строки кода добавьте прослушиватель событий для события нажатия и выполните необходимые действия. Используется анонимная функция.

JS JavaScript

```
guessButton.addEventListener("click", function () {
  var userLetter = userInput.value;
  fillUserWord(userArray, userLetter, word);
  mirrorUserWord(userArray);
  userInput.value = "";
  userInput.focus();
});
```

Код внутри анонимной функции в прослушивателе событий выполняет следующие действия (после нажатия кнопки):

- получает букву, написанную пользователем в поле ввода;
- обновляет массив userArray, сравнивая её с введённой буквой и угадываемым словом;
- обновляет элементы `span`, отображаемые на странице;
- очищает поле ввода;
- устанавливает курсор мыши в поле ввода, чтобы пользователю не приходилось делать это каждый раз вручную.

15. Добавьте в игру больше блюд (слов) и попробуйте.

16. Добавьте игру в меню навигации (`nav`).