МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

ГЛАВНОЕ УПРАВЛЕНИЕ ПО ОБРАЗОВАНИЮ

МОГИЛЁВСКОГО ОБЛАСТНОГО ИСПОЛНИТЕЛЬНОГО КОМИТЕТА

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«МОГИЛЕВСКИЙ ГОСУДАРСТВЕННЫЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

Специальность 2-40 01 01

Учебная группа ПО-455

**Контрольная работа №1**

**Программные средства**

**создания Интернет приложений**

Разработал О. И. Любаль

Руководитель: А. В. Карманов

Шифр14

2022

**25 Опишите синтаксис и алгоритм работы оператора цикла for в**

**языке JavaScript. Приведите примеры использования.**

Самый распространённый цикл — цикл for.

Выглядит он так:

for (начало; условие; шаг) {

// ... тело цикла ...

}

Разберёмся, что означает каждая часть, на примере. Цикл ниже выполняет alert(i) для i от 0 до (но не включая) 3:

for (let i = 0; i < 3; i++) { // выведет 0, затем 1, затем 2

alert(i);

}

Рассмотрим конструкцию for подробней:

Начало : let i = 0 Выполняется один раз при входе в цикл

Условие: i < 3 Проверяется перед каждой итерацией цикла. Если оно вычислится в false, цикл остановится.

Тело: alert(i) Выполняется снова и снова, пока условие вычисляется в true.

Шаг: i++ Выполняется после тела цикла на каждой итерации перед проверкой условия.

В целом, алгоритм работы цикла выглядит следующим образом:

Выполнить начало

→ (Если условие == true → Выполнить тело, Выполнить шаг)

→ (Если условие == true → Выполнить тело, Выполнить шаг)

→ (Если условие == true → Выполнить тело, Выполнить шаг)

→ ...

То есть, начало выполняется один раз, а затем каждая итерация заключается в проверке условия, после которой выполняется тело и шаг.

Вот в точности то, что происходит в нашем случае:

// for (let i = 0; i < 3; i++) alert(i)

// Выполнить начало

let i = 0;

// Если условие == true → Выполнить тело, Выполнить шаг

if (i < 3) { alert(i); i++ }

// Если условие == true → Выполнить тело, Выполнить шаг

if (i < 3) { alert(i); i++ }

// Если условие == true → Выполнить тело, Выполнить шаг

if (i < 3) { alert(i); i++ }

// ...конец, потому что теперь i == 3

Встроенное объявление переменной

В примере переменная счётчика i была объявлена прямо в цикле. Это так называемое «встроенное» объявление переменной. Такие переменные существуют только внутри цикла.

for (let i = 0; i < 3; i++) {

alert(i); // 0, 1, 2

}

alert(i); // ошибка, нет такой переменной

Вместо объявления новой переменной мы можем использовать уже существующую:

let i = 0;

for (i = 0; i < 3; i++) { // используем существующую переменную

alert(i); // 0, 1, 2

}

alert(i); // 3, переменная доступна, т.к. была объявлена снаружи цикла

Любая часть for может быть пропущена.

Для примера, мы можем пропустить начало если нам ничего не нужно делать перед стартом цикла.

let i = 0; // мы уже имеем объявленную i с присвоенным значением

for (; i < 3; i++) { // нет необходимости в "начале"

alert( i ); // 0, 1, 2

}

Можно убрать и шаг:

let i = 0;

for (; i < 3;) {

alert( i++ );

}

Это сделает цикл аналогичным while (i < 3).

А можно и вообще убрать всё, получив бесконечный цикл:

for (;;) {

// будет выполняться вечно

}

При этом сами точки с запятой ; обязательно должны присутствовать, иначе будет ошибка синтаксиса.

Прерывание цикла: «break»

Обычно цикл завершается при вычислении условия в false.

Но мы можем выйти из цикла в любой момент с помощью специальной директивы break.

Например, следующий код подсчитывает сумму вводимых чисел до тех пор, пока посетитель их вводит, а затем – выдаёт:

let sum = 0;

while (true) {

let value = +prompt("Введите число", '');

if (!value) break; // (\*)

sum += value;

}

alert( 'Сумма: ' + sum );

Директива break в строке (\*) полностью прекращает выполнение цикла и передаёт управление на строку за его телом, то есть на alert.

Вообще, сочетание «бесконечный цикл + break» – отличная штука для тех ситуаций, когда условие, по которому нужно прерваться, находится не в начале или конце цикла, а посередине или даже в нескольких местах его тела.

Переход к следующей итерации: continue

Директива continue – «облегчённая версия» break. При её выполнении цикл не прерывается, а переходит к следующей итерации (если условие все ещё равно true).

Её используют, если понятно, что на текущем повторе цикла делать больше нечего.

Например, цикл ниже использует continue, чтобы выводить только нечётные значения:

for (let i = 0; i < 10; i++) {

// если true, пропустить оставшуюся часть тела цикла

if (i % 2 == 0) continue;

alert(i); // 1, затем 3, 5, 7, 9

}

Для чётных значений i, директива continue прекращает выполнение тела цикла и передаёт управление на следующую итерацию for (со следующим числом). Таким образом alert вызывается только для нечётных значений.

Директива continue позволяет избегать вложенности

Цикл, который обрабатывает только нечётные значения, мог бы выглядеть так:

for (let i = 0; i < 10; i++) {

if (i % 2) {

alert( i );

}

}

С технической точки зрения он полностью идентичен. Действительно, вместо continue можно просто завернуть действия в блок if.

Однако мы получили дополнительный уровень вложенности фигурных скобок. Если код внутри if более длинный, то это ухудшает читаемость, в отличие от варианта с continue.

Нельзя использовать break/continue справа от оператора „?“

Метки для break/continue

Бывает, нужно выйти одновременно из нескольких уровней цикла сразу.

Например, в коде ниже мы проходимся циклами по i и j, запрашивая с помощью prompt координаты (i, j) с (0,0) до (2,2):

for (let i = 0; i < 3; i++) {

for (let j = 0; j < 3; j++) {

let input = prompt(`Значение на координатах (${i},${j})`, '');

// Что если мы захотим перейти к Готово (ниже) прямо отсюда?

}

}

alert('Готово!');

Нам нужен способ остановить выполнение если пользователь отменит ввод.

Обычный break после input лишь прервёт внутренний цикл, но этого недостаточно. Достичь желаемого поведения можно с помощью меток.

Метка имеет вид идентификатора с двоеточием перед циклом:

labelName: for (...) {

...

}

Вызов break <labelName> в цикле ниже ищет ближайший внешний цикл с такой меткой и переходит в его конец.

outer: for (let i = 0; i < 3; i++) {

for (let j = 0; j < 3; j++) {

let input = prompt(`Значение на координатах (${i},${j})`, '');

// если пустая строка или Отмена, то выйти из обоих циклов

if (!input) break outer; // (\*)

// сделать что-нибудь со значениями...

}

}

alert('Готово!');

В примере выше это означает, что вызовом break outer будет разорван внешний цикл до метки с именем outer.

Таким образом управление перейдёт со строки, помеченной (\*), к alert('Готово!').

Можно размещать метку на отдельной строке:

outer:

for (let i = 0; i < 3; i++) { ... }

Директива continue также может быть использована с меткой. В этом случае управление перейдёт на следующую итерацию цикла с меткой.

Метки не позволяют «прыгнуть» куда угодно

Метки не дают возможности передавать управление в произвольное место кода.

Например, нет возможности сделать следующее:

break label; // не прыгает к метке ниже

label: for (...)

Директива break должна находиться внутри блока кода. Технически, подойдет любой маркированный блок кода, например:

label: {

// ...

break label; // работает

// ...

}

…Хотя в 99.9% случаев break используется внутри циклов, как мы видели в примерах выше.

Сontinue возможно только внутри цикла.

**55 Разработать три взаимосвязанных HTML-страницы для**

**размещения на сайте.**

Первый HTML-документ предназначен для размещения на сайте в качестве первой страницы (имеющей обычно имя index.html, default.html или home.html). На странице должны располагаться следующие элементы:

* название фирмы;
* логотип фирмы;
* обращение к посетителю страницы;
* адрес фирмы;
* HTML-ссылки на две другие страницы сайта.

Второй HTML-документ предназначен для размещения простейшей рекламы товаров. На странице должны располагаться следующие элементы:

* название фирмы;
* логотип фирмы;
* обращение к посетителю страницы;
* прайс-лист в форме таблицы;
* HTML-ссылки на две другие страницы сайта.

Третий HTML-документ предназначен для размещения меню ресторана. На странице должны располагаться следующие элементы:

* заголовок “МЕНЮ РЕСТОРАНА” с указанием имени или фамилии учащегося;
* логотип фирмы;
* список блюд;
* HTML-ссылки на две другие страницы сайта.

При разработке HTML-страницы использовать указанную CSS-таблицу.

Название фирмы - заголовок первого уровня, выравнивается по левому краю, буквы зеленого цвета.

Остальной текст по правому краю, шрифт цвета azure, начертание- курсив, размер шрифта на 4 единицы больше текущего.

Таблица выровнена по правому краю, ширины рамки 3 пикселя, расстояние между границами ячеек равно 4.

Список вложенный. Типы маркеров выбрать самостоятельно.

Для задания свойств гипертекстовых ссылок использовать внедренную таблицу стилей.

**Код страницы Index.html**

<!DOCTYPE html>

<html lang='ru'>

<head>

<meta charset='utf-8'>

<title>Вечерний ресторан "Магнолия"</title>

<link rel="stylesheet" href="style.css" type="text/css" />

</head>

<body>

<center>

<img src='logo.png' width=300px> <br>

<h1>Вечерний ресторан "Магнолия"</h1>

</center><br><br>

<p class='rightazure'>

Приглашаем Вас посетить наш уютный ресторан. Вашену вниманию предлагаются самые изасканные блюда

<a href='z3.html'>(Меню)</a>,

домашняя обстановка, качественное обслуживание!

Каждый день повар готовит как минимум одно новое блюдо из европейских рецептов.

Приглашаем Вас и Ваших гостей отведать европейскую кухню!

<br>

Наш адрес: г. Минск, ул. Селицкого д. 69

</p>

<br><br>

<center>

<a href='z2.html'>Наша реклама</a>

<a href='z3.html'>Наше меню</a>

</center>

</body>

</html>

**Код страницы z2.html**

<!DOCTYPE html>

<html lang='ru'>

<head>

<meta charset='utf-8'>

<title>Вечерний ресторан "Магнолия"</title>

<link rel="stylesheet" href="style.css" type="text/css" />

</head>

<body>

<center>

<img src='logo.png' width=300px> <br>

<h1>Вечерний ресторан "Магнолия"</h1>

</center><br><br>

<br>Наш ресторан предлагает вам ассортимент товаров по доступным антикризисным ценам <br><br>

<br><br>

<table border="3" align = right >

<caption>Наш ассортимент:</caption>

<tr><th>Товар </th><th>Цена</th></tr>

<tr><td align=center>Белорусский погребок</td align=center><td align=center>2</td align=center></tr>

<tr><td align=center>Бутерброд с форшмаком</td align=center><td align=center>3</td align=center></tr>

<tr><td align=center>Тарелка к хреновухе</td align=center><td align=center>4</td align=center></tr>

<tr><td align=center>Cалат из куриного филе «Шляхетский»</td align=center><td align=center>5</td align=center></tr>

<tr><td align=center>Салат с сёмгой домашнего посола</td align=center><td align=center>2</td align=center></tr>

<tr><td align=center>Салат из свёклы с домашним сыром</td align=center><td align=center>43</td align=center></tr>

<tr><td align=center>Салат «Пан Тадеуш»</td align=center><td align=center>4</td align=center></tr>

<tr><td align=center>Сельдь «под шубой»</td align=center><td align=center>5</td align=center></tr>

<tr><td align=center>Сельдь по-Радзивилловски</td align=center><td align=center>6</td align=center></tr>

<tr><td align=center>Смалец с ржаным хлебом</td align=center><td align=center>7</td align=center></tr>

<tr><td align=center>Завтрак пана Лявона</td align=center><td align=center>2</td align=center></tr>

<tr><td align=center>Овощная тарелка с домашним жареным сыром</td align=center><td align=center>3</td align=center></tr>

<tr><td align=center>Салат «Бабушкин»</td align=center><td align=center>5</td align=center></tr>

<tr><td align=center>Тартар из говядины с черемшой</td align=center><td align=center>4</td align=center></tr>

<tr><td align=center>Дегустационный сет горячих закусок</td align=center><td align=center>3</td align=center></tr>

<tr><td align=center>Голубцы пани Стефы</td align=center><td align=center>2</td align=center></tr>

<tr><td align=center>Голубцы Литовские со сливочным хреном</td align=center><td align=center>6</td align=center></tr>

<tr><td align=center>Баклажаны в медовом соусе</td align=center><td align=center>7</td align=center></tr>

<tr><td align=center>Борщ виленский с пампушками</td align=center><td align=center>8</td align=center></tr>

<tr><td align=center>Домашняя солянка</td align=center><td align=center>4</td align=center></tr>

<tr><td align=center>Крем-суп из щавеля и шпината</td align=center><td align=center>4</td align=center></tr>

<tr><td align=center>Юшка «Кронон» из лосося и озёрной рыбы</td align=center><td align=center>5</td align=center></tr>

</table>

<br>

<a href='index.html'>Главная страница</a>

<a href='z3.html'>Наше меню</a>

</body>

</html>

**Код страницы z3.html**

<!DOCTYPE html>

<html lang='ru'>

<head>

<meta charset='utf-8'>

<title>“МЕНЮ РЕСТОРАНА” Любаль Ольга Игоревна</title>

<link rel="stylesheet" href="style.css" type="text/css" />

</head>

<body>

<center>

<img src='logo.png' width=300px> <br>

<h1>“МЕНЮ РЕСТОРАНА”<br> Любаль Ольга Игоревна </h1>

</center><br><br>

<ul>

<li>Белорусский погребок</li>

<li>Бутерброд с форшмаком</li>

<li>Тарелка к хреновухе</li>

<li>Cалат из куриного филе «Шляхетский»</li>

<li>Салат с сёмгой домашнего посола</li>

<li>Салат из свёклы с домашним сыром</li>

<li>Салат «Пан Тадеуш»</li>

<li>Сельдь «под шубой»</li>

<li>Сельдь по-Радзивилловски</li>

<li>Смалец с ржаным хлебом</li>

<li>Завтрак пана Лявона</li>

<li>Овощная тарелка с домашним жареным сыром</li>

<li>Салат «Бабушкин»</li>

<li>Тартар из говядины с черемшой</li>

<li>Дегустационный сет горячих закусок

<ul>

<li>Суп</li>

<li>Щи</li>

<li>Борщ</li>

</ul>

</li>

<li>Голубцы пани Стефы</li>

<li>Голубцы Литовские со сливочным хреном</li>

<li>Баклажаны в медовом соусе</li>

<li>Борщ виленский с пампушками</li>

<li>Домашняя солянка</li>

<li>Крем-суп из щавеля и шпината</li>

<li>Юшка «Кронон» из лосося и озёрной рыбы</li>

</ul>

<a href='index.html'>Главная страница</a>

<a href='z2.html'>Наша реклама</a>

</body>

</html>

**Код файла style.css**

h1 {

font-size: 32px;

color: green;

text-align : left;

}

p.rightazure{text-align: right;

font-size: 1.2em;

font-style: italic;

color : azure;

}

table {

border: 3px;

}

td {

padding: 4px;

}

th {

padding: 4px;

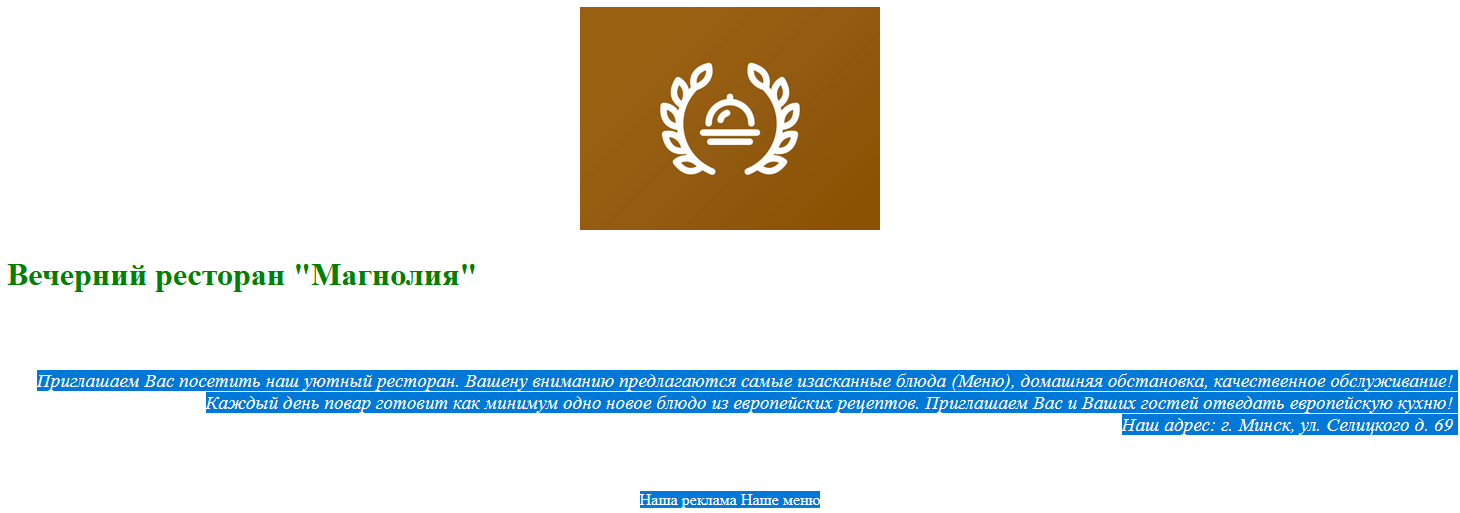
}

a{

color: magenta;

}

Скриншот страницы 1 представлен на рисунке 1

Рисунок 1. Скриншот страницы 1

Скриншот страницы 2 представлен на рисунке 2.

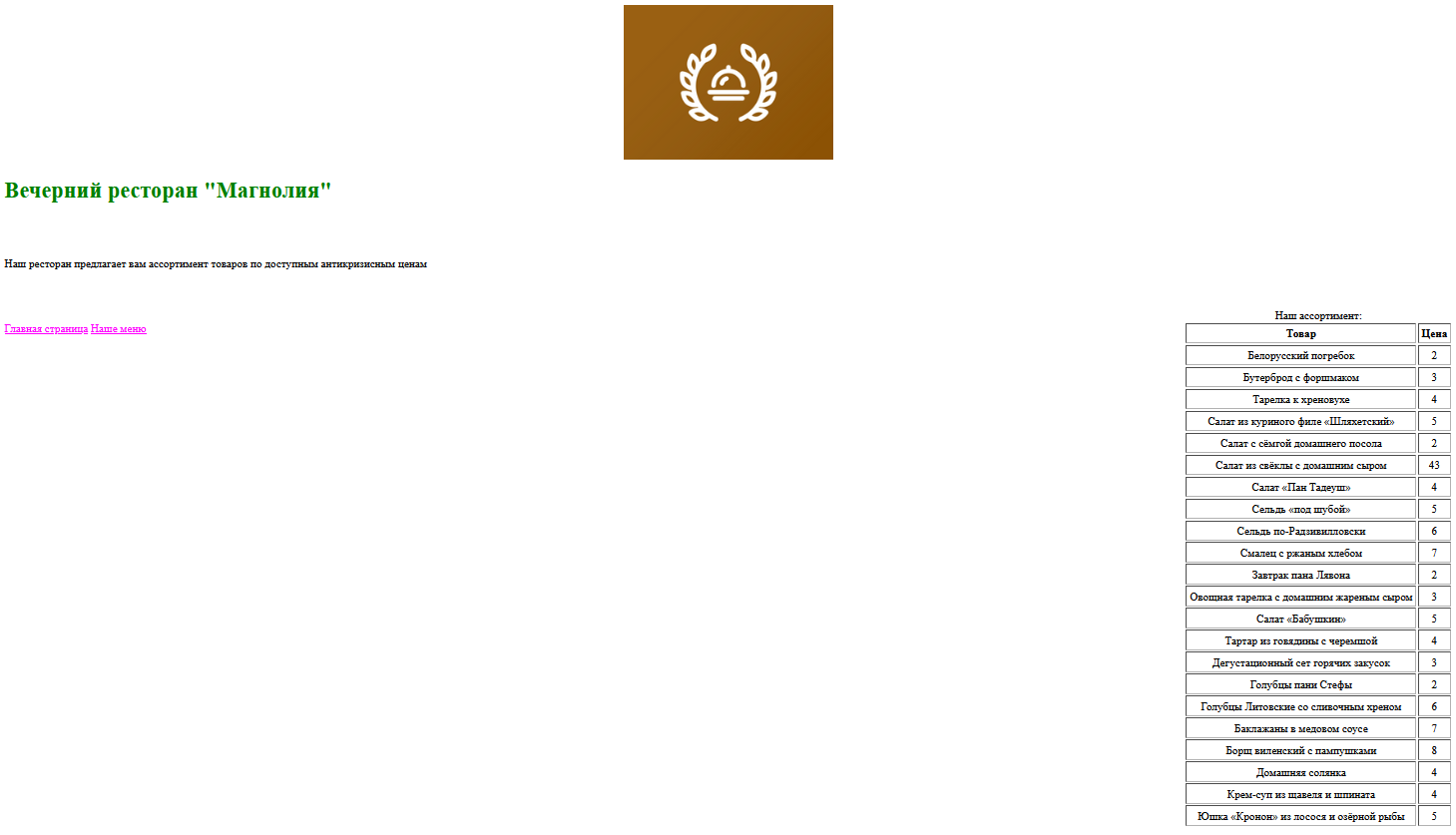


Рисунок 2. Скриншот страницы 2

Скриншот страницы 3 представлен на рисунке 3.

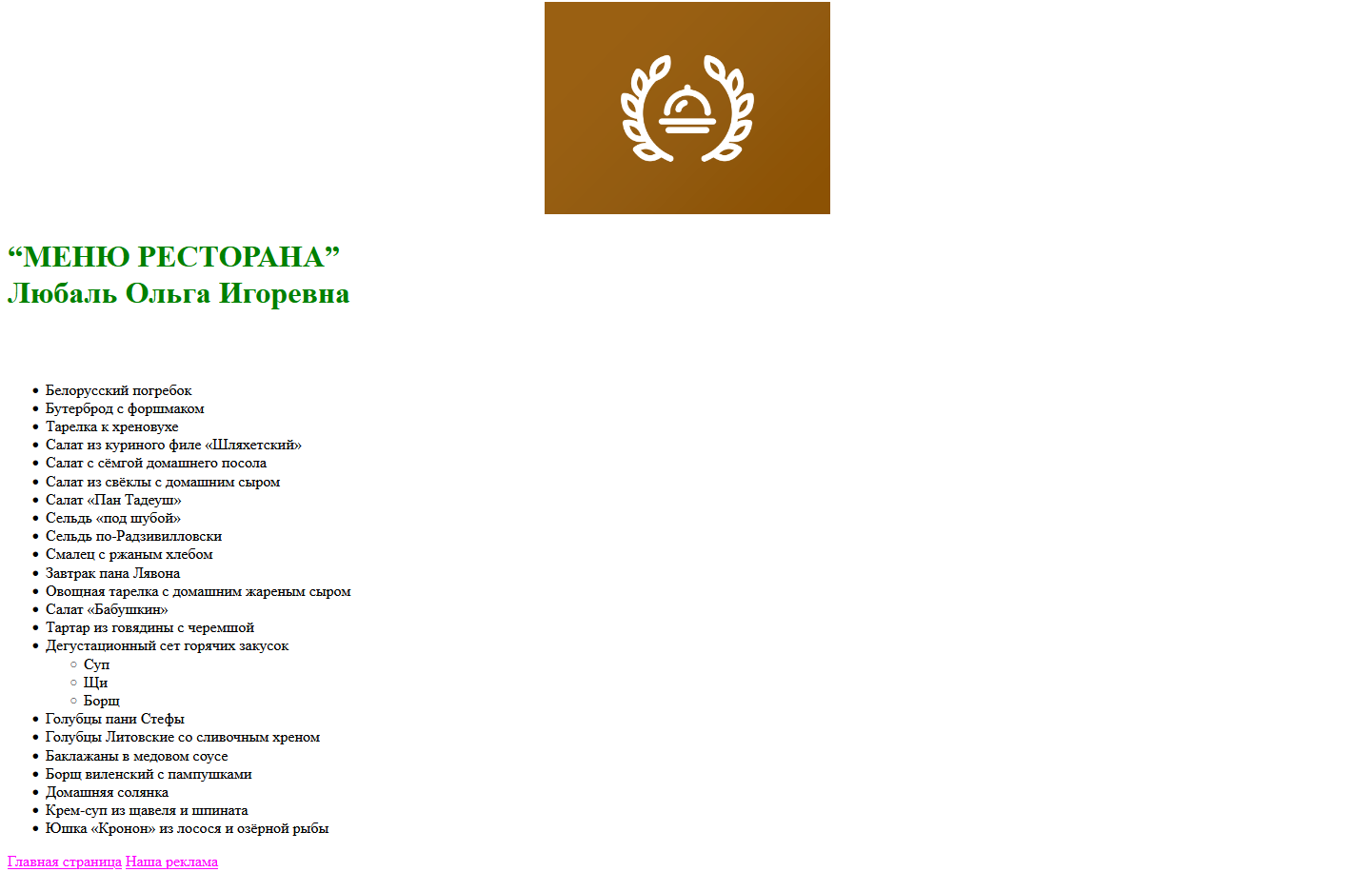


Рисунок 3. Скриншот страницы 3

**79 Требуется создать три скрипта на языке JavaScript и подключить их одной HTML-странице.**

Первый скрипт позволяет разместить в строке заголовка браузера Ваши фамилию, имя и отчество. Установить время вывода заголовка и длительность паузы до начала очередного вывода заголовка в соответствии с Вашим вариантом.

Второй скрипт содержит список, при выборе элемента которого цвет фона страницы меняется на выбранный. Список должен содержать не менее трех элементов. Цвета выбрать самостоятельно.

Третий скрипт предназначен для проверки знания таблицы истинности логического элемента. Тип логической функции, форму ввода информации и вид запускающего проверку события выбрать из таблицы 2 в соответствии с вариантом. Для обозначения событий приняты следующие сокращения: ЩЛК - щелчок левой кнопкой мыши (onClick); ДЩК - двойной щелчок левой кнопкой мыши (onDbClick); НКК - наведение курсора мыши на кнопку (onMouseOver)

**Код страницы Script.html**

<!DOCTYPE html>

<html lang='ru'>

<head>

<meta charset='utf-8'>

<title>Заголовок</title>

<link rel="stylesheet" href="style.css" type="text/css" />

</head>

<body>

<center>

<h1>Java Script Любаль Ольга</h1>

</center><br><br>

<div>

Работа с заголовком, через 4 секунды после нажатия запуска меняем<br>

заголовок на фамилию и затем до нажатия кнопки стоп<br>

переключаем с интервалом в 3 секунды<br>

<button id="start">Запустить</button>

<button id="stop">Остановить</button>

<script src="jquer.js"></script>

<script>

function sleep(milliseconds) {

const date = Date.now();

let currentDate = null;

do {

currentDate = Date.now();

} while (currentDate - date < milliseconds);

}

var timer;

var i;

i = true;

$('#start').click(function(){ sleep(1000);

timer = setInterval(function(){

if (i==true){

parent.document.title = 'Ресторан';

i = false;

}

else{

i = true;

parent.document.title = 'Любаль Ольга Игоревна';

//sleep(2000);

}

}, 3000);

});

$('#stop').click(function(){

clearInterval(timer);

parent.document.title = start\_title;

});

</script>

</div>

<br><br>

<hr>

Работа со списком, смена фона по клику на элемент

<script>

$(function() {

$('ul#col li').click( function() {

document.body.style.background = $(this).attr('value');

});

});

</script>

</script>

<ul id="col">

<li name='black' value='black'> черный </li>

<li name='green' value='green'> зеленый </li>

<li name='blue' value='blue'> синий </li>

<li name='magenta' value='magenta'> фиолетовый </li>

</ul>

<br><br>

<hr>

Проверка знания таблицы истинности логического элемента. A ∧ (B ∨ C) Включатель ЩЛК <br><br>

<input id="r11" name="r1" type="checkBox" value="false"> Проверить<br><br>

<input id = 'ca' type="checkbox" id=”a”>A<br>

<input id = 'cb' type="checkbox" id=”b”>B<br>

<input id = 'cc' type="checkbox" id=”c”>C<br>

<script>

$("#r11").mouseover(function(){ var c1;

var c2;

var c3;

c1=document.getElementById('ca');

c2=document.getElementById('cb');

c3=document.getElementById('cc');

res = c1.checked && (c2.checked || c3.checked);

if (res) {

alert('Верно');

}

else

{ alert('Неверно');

}

});

</script>

</body>

</html>

Скриншоты страницы Script.html представлены на рисунках 4-5

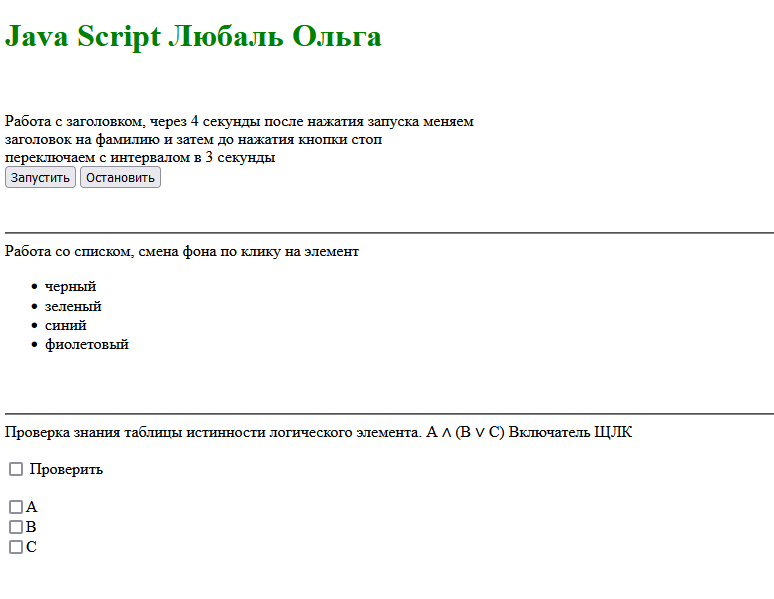


Рисунок 4. Скриншот стартового состояния страницы

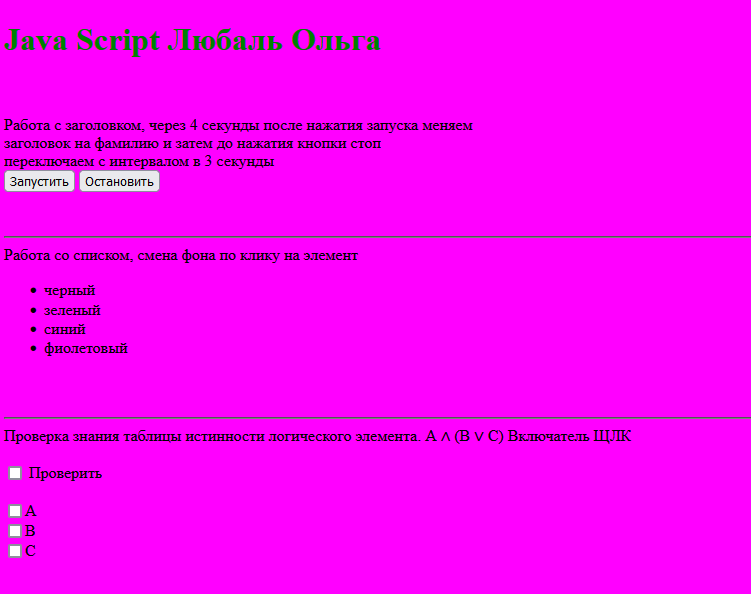


Рисунок 5. Скриншот страницы с измененным цветом.

**95 С использованием средств языка PHP разработать и отладить программу доступа к базе данных.**

База данных хранится на стороне сервера. Пользователь должен иметь возможность просмотреть все записи и сделать запрос.

Разрабатываемая программа должна удовлетворять следующим требованиям:

* разрабатываемое программное приложение должно содержать HTML-документ с формой для ввода данных и PHP-программу, вызываемую по окончании работы с HTML-формой;
* ввод параметров задания в HTML-форме может быть осуществлен либо путем ввода значений в текстовом виде, либо посредством выбора значений из предлагаемого списка;
* программа должна быть написана в предположении, что любой пользователь без ограничений может иметь доступ к данным;
* в программе должен быть предусмотрен вывод сообщений обо всех шагах ее выполнения, в том числе и о возможных ошибках;
* программа должна быть достаточно документирована.

В базе данных содержится информация о туристических поездках: страна, город, изображение городской достопримечательности, количество дней, дата поездки, класс отеля, цена.

Запросы:

а) вывести информацию о турах на заданную дату;

б) вывести информацию о турах, продолжительность которых

находится в заданном диапазоне.

**Код страницы database.html**

<!DOCTYPE html>

<html lang='ru'>

<head>

<title>Работа с базой данных</title>

</head>

<body>

<form name="form1" action="database.php?act=create" method="post">

<h4>Создать пустую базу данных</h4><br>

Если база существует то она не изменится, для того чтобы создать пустую базу надо удалить файл database.db3<br>

<input type="submit" value="Создать">

</form>

<hr>

<form name="form1" action="database.php?act=insert" method="post">

<h4>Добавление записи</h4><br>

Страна: <input type=text name='country'><br>

Город: <input type=text name='gorod'><br>

Фото: <input type=text name='photo'>(гиперссылка или имя файла)<br>

Дней: <input type=text name='day'><br>

Дата: <input type=text name='data'><br>

Класс: <input type=text name='class'><br>

Цена: <input type=text name='price'><br>

<input type="submit" value="Добавить">

</form>

<hr>

<form name="form2" action="database.php?act=listall" method="post">

<h4>Вывод всех записей</h4><br>

<input type="submit" value="Вывести все"></form>

<hr>

<h4>Вывод записей по условию:</h4><br>

<form name="form3" action="database.php?act=zapros&s=dt" method="post">

вывести информацию о турах на заданную дату;<br><br>

Дата:<input type=text name='dt'>

<input type="submit" value="Вывести"></form>

<form name="form4" action="database.php?act=zapros&s=list" method="post">

вывести информацию о турах, продолжительность которых

находится в заданном диапазоне.<br><br>

начало:<input type=text name='p1'><br>

конец:&nbsp;<input type=text name='p2'>

<input type="submit" value="Вывести">

</form>

<hr>

</body>

</html>

**Код страницы database.php**

<?php

$p1=-1;

$p2=-1;

if(isset($\_GET['act']))$act=$\_GET['act'];

if(isset($\_GET['s']))$step=$\_GET['s'];

if(isset($\_POST['p1']))$p1=$\_POST['p1'];

if(isset($\_POST['p2']))$p2=$\_POST['p2'];

if(isset($\_POST['dt']))$dt=$\_POST['dt'];

if(isset($\_POST['country']))$country=$\_POST['country'];

if(isset($\_POST['gorod']))$gorod=$\_POST['gorod'];

if(isset($\_POST['photo']))$photo=$\_POST['photo'];

if(isset($\_POST['day']))$day=$\_POST['day'];

if(isset($\_POST['data']))$data=$\_POST['data'];

if(isset($\_POST['class']))$class=$\_POST['class'];

if(isset($\_POST['price']))$price=$\_POST['price'];

$db = new SQLite3('database.db3');

switch ($act) {

case "create":

$results = $db->query('CREATE TABLE IF NOT EXISTS kurort ('.

'country text(200), gorod TEXT (200), photo text(500), day NUMERIC,

data text(30), class numeric, price numeric);');

echo "База создана<br>";

break;

case "listall":

$results = $db->query("select \* from kurort");

$p="<table border=1><caption>Записи о турах:</caption><tbody>";

$p.="<tr><th>Страна</th><th>Город</th><th>Фото</th><th>Дней</th><th>Дата</th><th>Класс</th><th>Цена</th></tr>";

while($r = $results->fetchArray()){

$p.="<tr>";

$p.= "<td>".$r['country']."</td>";

$p.= "<td>".$r['gorod']."</td>";

$p.= "<td><a href='".$r['photo']."' target=\_blank><img src='".$r['photo']."' width=100></a></td>";

$p.= "<td>".$r['day']."</td>";

$p.= "<td>".$r['data']."</td>";

$p.= "<td>".$r['class']."</td>";

$p.= "<td>".$r['price']."</td>";

$p.="</tr>";

}

$p.="</tbody></table>";

echo $p;

break;

case "zapros":

switch ($step){ case "dt":

if($dt=='' ){

echo "Непорядок с датой, выводим все записи ";

$results = $db->query("select \* from kurort where 1=1");

}

else {

$results = $db->query("select \* from kurort where data='".$dt."'");

}

break;

case "list":

if($p1=='' || $p2=='' ){

echo "Непорядок с диапазоном, выводим все записи ";

}

else {

echo "Дни от: ".$p1." и до: ".$p2;

$results = $db->query("select \* from kurort where day between $p1 and $p2");

}

break;

}

$p="<table border=1><caption>Записи о турах:</caption><tbody>";

$p.="<tr><th>Страна</th><th>Город</th><th>Фото</th><th>Дней</th><th>Дата</th><th>Класс</th><th>Цена</th></tr>";

while($r = $results->fetchArray()){

$p.="<tr>";

$p.= "<td>".$r['country']."</td>";

$p.= "<td>".$r['gorod']."</td>";

$p.= "<td><a href='".$r['photo']."' target=\_blank><img src='".$r['photo']."' width=100></a></td>";

$p.= "<td>".$r['day']."</td>";

$p.= "<td>".$r['data']."</td>";

$p.= "<td>".$r['class']."</td>";

$p.= "<td>".$r['price']."</td>";

$p.="</tr>";

}

$p.="</tbody></table>";

echo $p;

break;

case "insert":

if($country=='' || $gorod==''||$photo==''||$day==''||$data==''||$class==''||$price==''){

echo "Заполниите все поля!!";

}

else{

$z="insert into kurort(country,gorod,photo,day,data,class,price) values('$country','$gorod','$photo','$day','$data','$class','$price')";

$results = $db->query($z);

echo "Запись вставлена"; }

break;

}

echo"<br><br><center><a href='database.html'>Перейти в начало</a></center>";

?>

Скриншоты работы программы представлены на рисунках 6-9

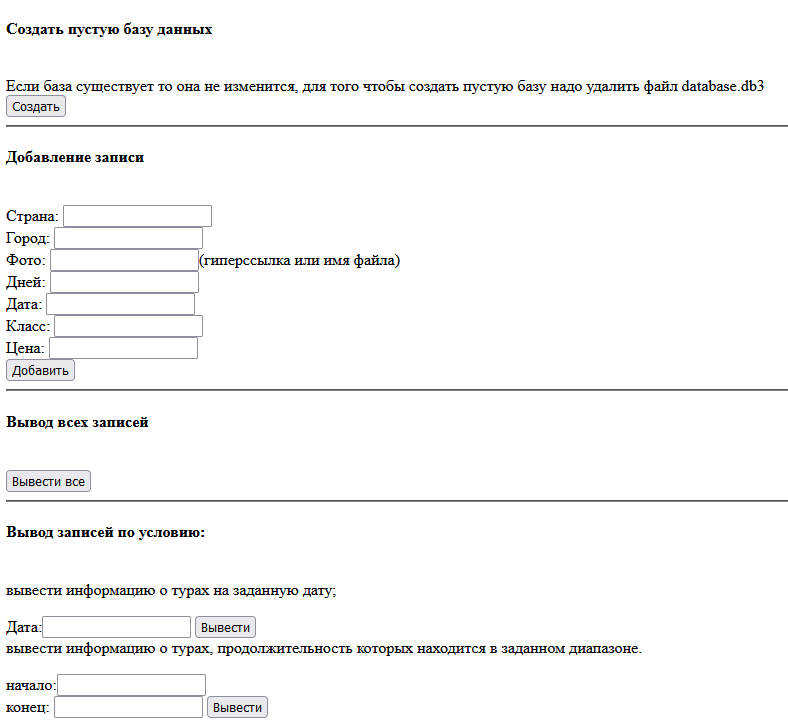


Рисунок 6. Стартовая страница работы с базой данных

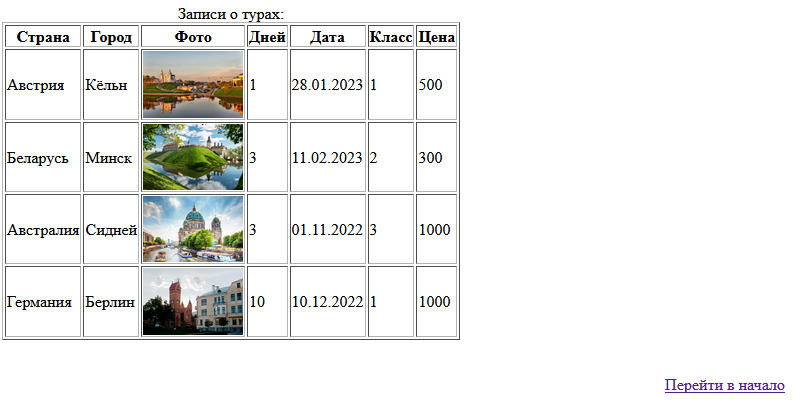


Рисунок 7. Вывод всех записей базы данных

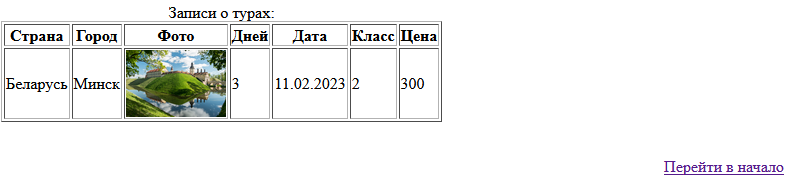


Рисунок 8. Вывод записей на дату 11.02.2023

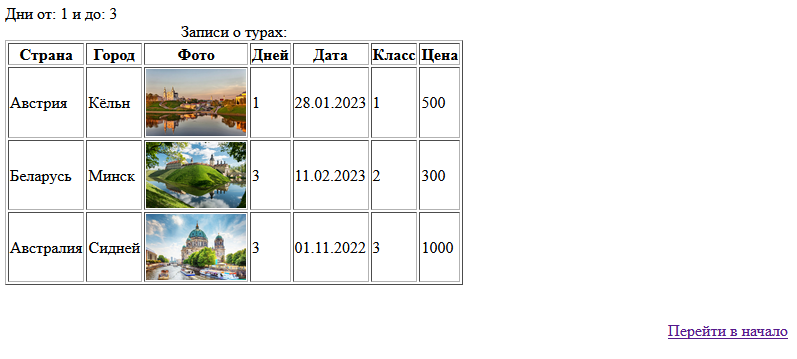


Рисунок 9. Вывод записей с длительностью тура от 1 до 3 дней.

**Список использованных источников**

1 Комолова, Н. HTML : Самоучитель / Н. Комолова, Е. Яковлева. 2-е изд. - СПб. : Питер, 2011. - 288 c.

2 Кузнецов, М. В. PHP. Народные советы / М. В.Кузнецов, И. В. Симдяков. – СПб.: БХВ-Петербург, 2007. – 368 с.: ил.

3 Соколов, С. А. HTML и CSS в примерах, типовых решениях и задачах. Профессиональная работа / С. А. Соколов. – М.: Издательский дом «Вильямс», 2007. – 416 с.: ил.

4 Фленаган, Д. JavaScript. Подробное руководство / Д. Фленаган. - СПб. : Питер, 2012. - 1080 с.

5 Фримен, Эл. Изучаем HTML, XHTML и CSS / Эл. Фримен, Эр. Фримен. - СПб. : Питер, 2014. - 720 с.