МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизированные системы управления»

ТЕХНОЛОГИИ ИНТЕРНЕТ-ПРОГРАММИРОВАНИЯ

Методические рекомендации к лабораторным работам для студентов специальности 1-53 01 02 «Автоматизированные системы обработки информации» очной и заочной форм обучения

Могилев 2019

УДК 004.43
ББК 32.973.26 – 018.1
Т33

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Автоматизированные системы управления» «18» мая 2019 г., протокол № 10

Составитель канд. техн. наук, доц. Э. И. Ясюкович

Рецензент канд. техн. наук, доц. В. В. Кутузов

Методические рекомендации содержат основные базовые теоретические сведения, некоторые приемы реализации задач, а также практические задания для выполнения лабораторных работ по всем темам курса.

Учебно-методическое издание

ТЕХНОЛОГИИ ИНТЕРНЕТ-ПРОГРАММИРОВАНИЯ

Ответственный за выпуск	А. И. Якимов
Технический редактор	А. А. Подошевко
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 31 экз. Заказ № .

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, Могилев.

Содержание

Введение

Целью изучения дисциплины «Технологии интернет-программирования» является приобретение специальных знаний, умений и практических навыков, необходимых менеджеру по информационным технологиям для разработки интернет-приложений.

Язык html (Hyper Text Markup Language) используется для добавления разметки в обычный текст, позволяет создавать статические и динамические сайты и является языком, описывающим структуру и семантику веб-документа.

Стандарт языка html непрерывно обновляется и почти каждый год выходит его новая версия. Версия html 2.0 была опубликована в 1995 г., html 4.01 – основная версия html – в конце 1999 г. В настоящее время наиболее популярна версия html5, являющаяся расширением html 4.01.

Каскадные таблицы стилей (Cascading Style Sheets – CSS) влияют на отображение страниц в окнах браузеров (цвета, шрифты, фоновые изображения, интервалы между строками, отступы, границы, эффекты и даже анимация элементов). Благодаря CSS можно производить изменения, относящиеся ко всем страницам сайта, редактируя при этом лишь один единственный файл таблицы стилей.

Для упрощения процесса разработки JavaScript скриптов Джоном Резигом была разработана библиотека jQuery, которая постоянно дополняется добровольцами. Основная цель создания jQuery – возможность создания многоразовых фрагментов кода JavaScript, позволяющих упростить процесс использования их в html-документах. Также библиотека jQuery предоставляет удобный программный интерфейс приложения API для работы с AJAX – интерактивным пользовательским интерфейсом.

Методические рекомендации предназначены для изучения основ технологии интернет-программирования и состоят из двух частей. Первая часть содержит 12 лабораторных работ, вторая – семь. В каждой части излагается краткий теоретический материал по вопросам соответствующей темы, предлагаются технологии решения некоторых типовых задач, приводятся задания для выполнения работ, а также контрольные вопросы.

Первая лабораторная работа первой части методических рекомендаций посвящена изучению языка html и CSS. Следующие работы первой части ориентированы на изучение основ языка JavaScript и библиотеки jQuery.

Вторая часть методических рекомендаций (работы № 13–19), ориентирована на изучение серверного языка PHP.

Лабораторные работы № 1–6, 8, 12 рассчитаны на 2 ч, а работы № 7, 9–19 – на 4 ч. Цель каждой работы соответствует ее названию.

Выполнение каждой работы производится в следующем порядке:

1) ознакомиться с теоретическими положениями работы;

2) выбрать из таблицы «Варианты заданий» согласно варианту, указанному преподавателем, задания и исходные данные, выполнить задания и оформить отчет.

1 Клиентское программирование

1.1 Лабораторная работа № 1. Создание html-страницы с применением CSS

CSS используются для создания и изменения стиля элементов веб-страниц и пользовательских интерфейсов, написанных на языках html и xhtml.

Объявление стиля состоит из двух частей: элемента веб-страницы – селектора – и блока объявления. Селектор сообщает браузеру, какой элемент необходимо форматировать, а блок объявления (код в фигурных скобках), содержит форматирующие команды, указывающие свойства со значениями.

В работе предлагается, используя язык разметки гипертекста html5 и каскадную таблицу стилей CSS3, разработать веб-страницу формирования меню, содержащего главную ленту с графическим элементом, и два пункта с выпадающими подменю. Пример задачи приведен на листингах 1.1 и 1.2.

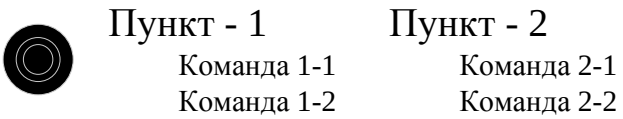

Рисунок 1.1 – Вид меню

Листинг 1.1 – html-код страницы

```
<!DOCTYPE html>
<html>
<head>
<title>Меню</title>
<link rel="stylesheet" type="text/css"
href="css/style.css" />
</head>
<body>
<div id="mainmenu">
   <ul id="nav">
<li id="settings">
<a href="#"><img src="images/settings.png"></a>
</li>
  <li><a href="">Пункт-1</a>
   <ul>
     <li><a href="#">Команда 1-1</a></li>
     <li><a href="#">Команда 1-2</a></li>
   </ul>
  </li>
 </ul>
</div>
</body>
</html>
```

Листинг 1.2 – CSS-код страницы

```
#mainmenu      {
  float:left;     }
#mainmenu ul  {
  margin: 10;  /* отступ ленты меню сверху*/
  padding: 5;          /* отступ ленты меню
слева*/
  list-style: none;      }
#mainmenu ul li        {
  padding: 10px 0 10px 0;
  text-align:center;
  background: #00f0f0;/*цвет фона гл  меню*/
}
#mainmenu li ul li a            {
  padding:5px 0 3px 10px;
  text-align:left;
  font-size:12px;
```

```
  position: relative;
  float:left;     }
#mainmenu ul li ul, #mainmenu ul li ul li     {
  width:130px;
  color:red;    /*      */                  }
#mainmenu li ul        {
  position: absolute;
  left: 0;
  top: 29px;
  display: none;
  float:left;     }
#mainmenu ul li a      {
  float:left;
  color:blue;   /*Цвет текста глав меню  */
  width:80px; /*Длина ленты глав меню  */
  font-size:16px;
  width:80px;
  background: #EEEEEE;    /*Цвет          поля
группы выбр-го подменю*/     }
 #mainmenu li ul li a:hover    {
  background: #0ffff0;   /*Цвет фона подменю
*/
  color:blue;   }
 #settings a {             /*Графич-й эл-т */
  padding: 18px;
  height: 19px;          /*Высота графич эл-та */
  font-size: 10px;
  line-height: 24px;   }
* html #mainmenu ul li {float: left; height: 1%;}
* html #mainmenu ul li a { height: 1%; }
#mainmenu li:hover ul,
#mainmenu  li.over ul { display: block;       }
```

Порядок выполнения работы.

Выполнить задания, приведенные в колонках 2, 3, 4 и 5 таблицы 1.1. В колонке 2 указаны темы, для которых необходимо разработать веб-страницу формирования меню, в колонке 3 – параметры главной ленты меню: направление главной ленты меню (гориз – горизонтальное, верт – вертикальное); фон – цвет фона ленты меню; текст – цвет текста пунктов меню. В колонках 4 и 5 указано количество пунктов второго и третьего подменю (два или три), цвет фона подменю и цвет текста его пунктов.

Главное меню должно отстоять от верхней границы окна браузера на 20 px, слева – на 12 px и содержать слева графический элемент.

Таблица 1.1 – Варианты заданий

Номер варианта	Тема	Главная лента меню: фон/текст	Подменю 1: пунктов/фон/текст	Подменю 2: пунктов/фон/текст
1	2	3	4	5
1	Торговая фирма	Гориз/Pink/Red	3/Blue/Blue	2/Aqua/Violet
2	Ремонт авто	Верт/Coral/Blue	2/Navy/Navy	3/Olive/Aqua
3	Расписание занятий	Гориз/Gold/Navy	3/Olive/Aqua	2/Navy/Navy
4	Летняя одежда	Верт/Khaki/Aqua	2/Aqua/Violet	3/Blue/Blue
5	Ремонт компьютеров	Гориз/Plum/Black	3/Black/Red	2/Tan/Gold
6	Швейное ателье	Верт/Purple/Violet	2/Tan/Gold	3/Violet/Blue
7	Ремонт телевизоров	Гориз/Indigo/Tan	3/Violet/Blue	2/Purple/Navy
8	Головные уборы	Верт/Peru/Coral	2/Purple/Navy	3/Blue/Red
9	Зимняя одежда	Гориз/Gray/Gold	3/Blue/Red	2/Olive/Aqua
10	Продукты питания	Верт/Red/Plum	2/Olive/Aqua	3/Peru/ Gold
11	Детские игрушки	Гориз/Lime/Peru	3/Peru/ Gold	2/Pink/ Blue
12	Модели авто	Верт/Aqua/Gray	2/Pink/ Blue	3/Blue/Blue

Контрольные вопросы

1 Какие программные средства используются для создания веб-страниц?
2 Для чего используются CSS в веб-программировании?
3 Какова структура объявления стиля?
4 Из каких разделов состоит html-документ?
5 Как изменить параметры текста в html-документе (размер, цвет, стиль)?
6 Как создать таблицу в html-документе и вставить рисунок в ее ячейку?
7 Как построить гиперссылку, заголовок и список в html-документе?
8 Как построить гиперссылку на графическом элементе?
9 Как установить параметры ленты меню веб-страницы?
10 Как подключить таблицу CSS стилей к html-документу?

1.2 Лабораторная работа № 2. Создание простых скриптов на JavaScript

Для расширения функциональных возможностей веб-страниц используются скрипты JavaScript, которые включаются в html-документы несколькими способами:

– в теговом контейнере <body>...</body>:

```
<body>
…
<script > команды скрипта</script>
…
</body>
```

– в контейнере <head>...</head>, если скрипт представляет собой функцию, вызываемую в ответ на какое-либо событие:

```
<head>
…
<script type=”text/javascript”> команды сценария </script>
…
</head>
```

– во внешнем файле с расширением js:

```
<head>
…
<script type = "text/javascript" src = "my.js"> </script>
…
</head>
```

Порядок выполнения работы.

Выполнить четыре следующих задания:

1 Создать строку текста из 22 первых букв русского алфавита: var str = 'abcde … '. Используя функцию ***alert***, вывести символы с указанными в

столбце «Задание 1» таблицы 1.2 номерами и разделить их номером варианта, например, для варианта 11: a-11-c-11-e11 и так далее.

2 Построить строку из цифр «Номера символов» первого задания. Из полученной строки, состоящей из 12 цифр, выделить четыре трехзначных числа. Используя полученные числа и заданные в колонке «Задание 2» таблицы 1.2 операции, построить оператор присваивания с полученным арифметическим выражением, полученный результат вывести в окно браузера.

3 Построить строку текста из букв, номера которых заданы в «Задание 1». Используя свойство ***innerHTML*** метода ***document.getElementById(id),*** вывести на страницу html построенную строку.

4 С помощью функции ***confirm*** построить запрос, содержащий две кнопки: Да и Нет. В зависимости от выбранной кнопки вычислить заданные в «Задании 3» таблицы 1.2 выражения: для Да – y, для Нет – f. Результат вывести на страницу html используя функцию writeln.

Таблица 1.2 – Задания к лабораторной работе

Номер вари-анта	Задание 1 – номера символов	Задание 2 – операции			Задание 3 – арифметическое выражение	
					y =	f =
1	01, 03, 05, 06, 08, 09	+	–	*	$(a + b)^2/c$;	d/(f – e/2)
2	02, 03, 04, 06, 07, 15	+	–	/	$(a * b) – c^2$;	2*d/(f + e/4)
3	04, 05, 07, 09, 12, 18	+	–	%	$(a – b/c)^2$;	d*(2*f – e/2)
4	03, 05, 06, 12, 20, 21	/	*	–	$(a^2 + b)/c$;	d/(f/5 + e^2)
5	01, 02, 08, 15, 17, 19	/	*	+	$(a – b/c)^2/c$;	2*d/(f/1.4 – e/2)
6	13, 15, 16, 17, 18, 20	/	*	%	$(a + b/c^2)/c$;	d/(f*e/2) + d
7	05, 07, 09, 13, 16, 18	*	+	–	$(a – b/a)/c^2$;	d/(f*e + 2*d) – d
8	03, 04, 09, 12, 14, 17	*	+	/	$(a + b/a^2)/2*c$;	d/(f*e – 2*d) + e
9	02, 09, 16, 17, 18, 21	*	+	%	$(a – b/a*2)/c^2$;	d/(e*f/2) + d*e/f
10	03, 07, 08, 09, 10, 20	–	*	/	$(a + b/c^2)/c-a$;	d/(f*e/2 + d) – 2*f
11	01, 03, 05, 07, 10, 11	–	*	+	$(a – b/c/2)/c^a$;	d/(f*e + 2*f) + d
12	06, 08, 09, 12, 13, 19	–	*	%	(a + b/c+2)/c–2*b;	d/(f*e – 2/f) – d

Контрольные вопросы

1 Как выделить символ из строки текста?
2 Как встроить JavaScript-код в html-документ?
3 Какие комментарии используются в языке JavaScript?
4 Какие функции вывода на страницу html Вы знаете?
5 Прокомментируйте технологию использования метода document.getElementById(id) вывода на страницу html.
6 Какие математические операции используются в JavaScripn?
7 Как извлечь символ строки по его номеру?
8 Для чего используются методы document.write(), alert(), console.log()?

9 Какие способы включения JavaScript-кода в html-документ Вы знаете?
10 Для каких целей используются методы prompt и Confirm?

1.3 Лабораторная работа № 3. Функции и обработка событий JavaScript

В JavaScript используются простые типы данных и объекты. К простым типам относятся числа, строки, логический тип, null и undefined. Среди объектов выделяют обычные и специальные объекты. Обычные объекты – это число или строка, а специальные объекты – это массивы, функции, объект даты, регулярные выражения и ошибки.

Если при наступлении события требуется произвести много действий, то удобно написать сценарий в виде функции и разместить его в контейнере <script> …</script>, предназначенном для сценариев. Например, для вывода модуля заданного числа используется функция Math.abs, а для округления чисел – функции Math.round, Math.ceil и Math.floor, а также методы toFixed и toPrecision.

Функции Math.min, Math.max, Math.sqrt, Math.pow, Math.random используются для определения минимального, максимального значений, вычисления квадратного корня, возведения в степень и генерации псевдослучайных чисел с равномерным законом распределения.

Для работы со строками текста используются следующие методы: length, toUpperCase, toLowerCase, substr, substring, slice, indexOf, replace, split и функция join. Чтобы обратить внимание пользователя веб-сайта на определённый элемент html-документа, его свойства можно менять, например, цвет или размер, при попадании на него курсора мыши, а при снятии курсора восстанавливать прежние значения.

Порядок выполнения работы.

Выполнить два задания (таблица 1.3).

Таблица 1.3 – Задания к лабораторной работе

Номер варианта	Задание 1		Задание 2
	Используя Math.abs, вычислить модуль числа	Округлить число до n знаков в дробной части, используя функцию	Выполнить операции с элементами одномерного массива из n вещественных чисел, используя функцию
1	2	3	4
1	278,785	Math.round, n = 1	Math.min, n = 8
2	547,473	Math.ceil, n = 2	Math.max, n = 9
3	182,487	Math.floor, n = 3	Math.sqrt, n = 5 и просуммирвать
4	264,289	toFixed, n = 2	Math.pow, n = 6 и просуммирвать
5	452,297	Math.round, n = 2	Math.random, n = 9 и просуммирвать
6	984,573	Math.ceil, n = 4	Math.min, n = 7

Окончание таблицы 1.3

1	2	3	4
7	893,284	Math.floor, n = 1	Math.max, n = 8
8	487,651	toFixed, n = `3	Math.sqrt, n = 5 и просуммирвать
9	774,652	Math.round, n = 3	Math.pow, n = 9 и просуммирвать
10	682,571	Math.ceil, n = 1	Math.random, n = 7 и просуммирвать
11	455,628	Math.floor, n = 2	Math.min, n = 9
12	671,475	toFixed, n = 4	Math.max, n = 6

Контрольные вопросы

1 Как получить модуль числа?
2 Какие функции и методы округления чисел Вы знаете?
3 Прокомментируйте технологию использования функций Math.sqrt, Math.pow, Math.random.
4 Прокомментируйте технологию использования функций isNaN, isFinite, parseInt, parseFloat.
5 Какая функция используется для округления числа до необходимого количества цифр в его дробной части?
6 Какие методы для работы со строками Вы знаете?
7 В каких ситуациях возникает необходимость использования регулярных выражений при работе с текстом?
8 Как сформировать последовательность псевдослучайных чисел с равномерным распределением?
9 Как выполняется глобальный поиск и замена символа в строке текста?
10 Для чего используется метод indexOf при работе со строками текста?

1.4 Лабораторная работа № 4. Операторы ветвлений и циклов JavaScript

Операторы ветвления используются для организации выполнения блоков кода при выполнении или невыполнении некоторых условий. К таким операторам относятся конструкции if, else, switch.

Синтаксис конструкции if:

```
if (логическое выражение) {
        код если логическое выражение true
} else {
        код если логическое выражение false
}
```

Синтаксис конструкции switch:

```
switch (переменная) {
  case '1':
     код выполняемый, если переменная имеет значение 1;
```

```
  break;
  ...
  case 'n':
      код, выполняемый, если переменная имеет значение n;
  break;
  default:
      код, выполняемый, если переменная не совпала ни с одним значением;
  break;
}
```

Синтаксис конструкции for:

```
for (начальное значение; условие окончания цикла; команды после прохода цикла) {
    тело цикла
}
```

Примеры

```
1 Вывести четные числа от 10 до 0
for (var i = 10; i >= 2; i -= 2) {
        document.write(i + ' ');
}
```

```
2 Вывести элементы массива
var books = ['JavaScript', 'PHP', html, 'CSS'];
for (var i = 0; i < books.length; i++) {
        var str = (i + 1) + '. ' + books[i] + '<br>';
        document.write(str);
}
```

Синтаксис конструкции while:

```
while ( пока выражение истинно ) {
    выполнять код
}
```

Пример 1 – Вывести числа от 1 до 10:

```
var i = 1;
while (i <= 10) {
    document.write(i + ' ');
    i++;
}
```

Инструкция ***break*** используется для принудительного выхода из цикла, а инструкция continue – для принудительного перехода к следующей итерации цикла.

Пример 2 – Вывести четные числа от 2 до 10:

```
var result = ' ';
for (var i = 2; i <= 10; i++) {
    if (i % 2) continue;
    result += i + ' ';
}
document.write(result);
```

Порядок выполнения работы.

Разработать консольное приложение на языке JavaScript для решения следующих задач.

1 Ввести переменную ***lang***, которая может принимать значения: рус, анг, бел или нем. В переменной *msw* сформировать массив дней недели на русском, английском, белорусском или немецком языке в зависимости от варианта. Задачу решить с помощью оператора ***if***; ***switch-case*** или многомерного массива без ***if*** и ***switch-case*** в зависимости от варианта.

2 Дана строка вида 'ab12cde345'. Проверить, является ли символ с заданным номером (k) этой строки буквой, а сумма ее цифр – четной.

3 Ввести дату и по ней определить: ВГ – время года (зима, весна, лето, осень); ДМ – декаду месяца; МГ – месяц; ВсГ – високосный / не високосный год.

4 Ввести массив из 25 целых вещественных чисел и определить: СЧЭ – сумма четных элементов; ПЭНН – произведение элементов с нечетными номерами; СЭКЗ – сумма элементов, номера которых кратны трем; ПНЧЭ – произведение нечетных элементов массива.

Варианты заданий приведены в таблице 1.4.

Таблица 1.4 – Варианты заданий

Номер варианта	Задача 1		Задача 2		Задача 3	Задача 4
	lang	операт/массив	строка	k	определить	определить
1	рус	If	'abcde12345'	2	ВГ; ДМ	СЧЭ; ПЭНН
2	рус	switch-case	'ab123cde45'	4	ВГ; МГ	СЧЭ; СЭКЗ
3	рус	массив	'abcd12345e'	6	ВГ; ВсГ	СЧЭ; ПНЧЭ
4	анг	If	'ab12cde345'	8	ДМ; ВГ	ПЭНН; СЧЭ
5	анг	switch-case	'a12bcde345'	7	ДМ; МГ	ПЭНН; СЭКЗ
6	анг	массив	'a1234bcde5'	1	ДМ; ВсГ	ПЭНН; ПНЧЭ
7	бел	If	'123abc45de'	3	МГ; ВГ	СЭКЗ; СЧЭ
8	бел	switch-case	'123abcde45'	5	МГ; ДМ	СЭКЗ; ПЭНН
9	бел	массив	'12ab34cde5'	7	МГ; ВсГ	СЭКЗ; ПНЧЭ
10	нем	If	'123abc45de'	9	ВсГ; ВГ	ПНЧЭ; СЧЭ
11	нем	switch-case	'1ab23c45de'	4	ВсГ; ДМ	ПННЭ; ПЭНН
12	нем	массив	'1ab2345cde'	7	ВсГ; МГ	ПННЭ; СЭКЗ

Контрольные вопросы

1 Какие операторы ветвлений Вам известны?
2 Какие варианты использования оператора if Вы знаете?
3 Прокомментируйте назначение и структуру оператора switch-case.
4 Для чего предназначена инструкция break?
5 Какие операторы цикла Вы знаете?
6 Для чего предназначен оператор for in?
7 Как реализовать досрочный выход из цикла?

8 Прокомментируйте синтаксис оператора while.
9 Для чего используется инструкция continue?
10 Прокомментируйте синтаксис оператора for.

1.5 Лабораторная работа № 5. Методы JavaScript

Методы JavaScript – это действия, которые могут выполняться над объектами, в то же время это свойства, содержащие определение функции, например,

```
var person = {
   firstName: "John",
   lastName : "Doe",
   id : 5566,
   fullName : function() {
      return this.firstName + " " + this.lastName;
   }
};
```

Ключевое слово *this* в данном примере относится к владельцу функции, т. е. к объекту *Person*, который владеет функцией *function*().

Методом объекта JavaScript может быть только функция, а значением свойства объекта – любой тип данных, за исключением функции.

Основные методы JavaScript

Методы строк. Строка в JavaScript является одновременно и объектом string и переменной, поэтому может быть создана двумя способами:

```
st1 = new String("Строка – это объект")
st2 = "Строка – это переменная"
```

В JavaScript используются следующие методы строк:

charAt() – извлекает из строки символ, находящийся в указанной позиции;

charCodeAt() – возвращает код юникода символа, находящегося в указанной позиции (16-разрядное целое число между 0 и 65 535);

concat() – выполняет конкатенацию одного или нескольких значений со строкой. преобразует все аргументы в строки и добавляет их по порядку в конец строки;

indexOf (подстрока, начало) – выполняет поиск в строке от начала к концу;

lastIndexOf() – выполняет поиск символа или подстроки в строке с конца;

match() – выполняет поиск по шаблону с помощью регулярного выражения.

Для выделения нескольких символов строки используется метод substr().

Замена подстроки текста выполняется методом replace(), построенным на использовании регулярных выражений.

Методы вывода:

alert – выводит модальное окно с сообщением;

confirm – выводит сообщение в окне с двумя кнопками: «ОК» и «ОТМЕНА» и возвращает выбор посетителя;

prompt – выводит окно с указанным текстом и полем для пользовательского ввода;

setInterval – выполняет код или функцию через указанный интервал времени.

Метод document.write() выводит на страницу переданные ему аргументы.

Глобальные методы JavaScript:

alert – выводит модальное окно с сообщением;

clearInterval – останавливает выполнение кода, заданное setInterval;

clearTimeout – отменяет выполнение кода, заданное setTimeout;

confirm – выводит сообщение в окне с двумя кнопками «ОК» и «ОТМЕНА» и возвращает выбор посетителя;

decodeURI – раскодирует URI, закодированный при помощи encodeURI;

decodeURIComponent – раскодирует URI, закодированный при помощи encodeURIComponent;

encodeURI – кодирует URI, заменяя каждое вхождение определенных символов на escape-последовательности, представляющие символы в кодировке UTF-8;

encodeURIComponent – кодирует компоненту URI, заменяя определенные символы на соответствующие UTF-8 escape-последовательности;

eval – выполняет строку javascript-кода без привязки к конкретному объекту;

isFinite – проверяет, является ли аргумент конечным числом;

isNaN – проверяет, является ли аргумент NaN.

Порядок выполнения работы.

Выполнить задания.

1 Заполнить массив из 14 элементов вещественными числами, используя заданную в колонке 2 таблицы 1.5 функцию. Рассортировать массив в указанном в колонке 3 порядке.

Таблица 1.5 – Варианты заданий

Номер варианта	Задание 1		Задание 2	Задание 3
	Функция	Сортировать	Даты	Вычислить
1	2	3	4	5
1	Sin(x)	По возрастанию	Год, месяц, день	Площадь усеченного конуса
2	Cos(x)	По убыванию	Месяц, день, час	Площадь трапеции
3	Tg(x)	По возрастанию	Час, минута, секунда	Площадь параллелограмма
4	Ctg(x)	По убыванию	День, час, минута	Макс. значение среди трех
5	Sin(x/2)	По возрастанию	Год, месяц, день	Макс. значение среди пяти
6	Cos(x/2)	По убыванию	Месяц, день, час	Объем конуса
7	Tg(x/2)	По возрастанию	Час, минута, секунда	Объем усеченного конуса
8	Ctg(x/2)	По убыванию	День, час, минута	Объем пирамиды
9	Sin(x^2)	По возрастанию	Год, мес, день	Объем усеченной пирамиды
10	Cos(x^2)	По убыванию	Месяц, день, час	Площадь шестиугольника
11	Tg(x^2)	По возрастанию	Час, минута, секунда	Площадь кольца
12	Ctg(x^2)	По убыванию	День, час, минута	Площадь цилиндра

2 Вывести текущую дату в заданном в колонке 4 формате и определить:

– используя функцию ***DataParse***, количество миллисекунд, прошедших с 01.01.1970 г. по текущий момент;

– используя метод ***getTime***, количество секунд с 01.01.1970 г. по текущий момент;

– используя метод ***getDay***, номер и название дня недели, в который Вы родились.

3 Вычислить значение элемента, заданного в колонке 5.

Контрольные вопросы

1 Для чего используются методы *alert*(), *prompt*() и *document.write*()?
2 Какие методы объекта *Math* Вы знаете?
3 Прокомментируйте назначение метода *document.getElementById(id)*.
4 Какие два способа создания строки Вы знаете?
5 Что понимается под методом в *JavaScript*?
6 Какие методы вывода в JavaScript Вы знаете?
7 Какой метод используется для поиска номера символа, с которого начинается подстрока?
8 Через какой объект выполняется работа с датами в *JavaScript*?
9 Какой метод используется для округления чисел?
10 Прокомментируйте назначение функции *Date.parse*.

1.6 Лабораторная работа № 6. Работа с массивами на JavaScript

Массивы в *JavaScript* являются нетипизированными, т. е. элементы одного и того же массива могут иметь разные типы. Элементы массива могут быть объектами или другими массивами, что позволяет создавать сложные структуры данных, такие как массивы объектов и массивы массивов.

Отсчет индексов массивов в языке JavaScript начинается с нуля, и для них используются 32-битные целые числа. Массивы в JavaScript являются динамическими: они могут увеличиваться и уменьшаться в размерах по мере необходимости, поэтому нет необходимости объявлять фиксированные размеры массивов при их создании или повторно распределять память при изменении их размеров.

Создать массив проще всего с помощью литерала, который представляет собой простой список разделенных запятыми элементов в квадратных скобках. Значениями литерала массива могут быть константы, выражения, литералы объектов:

```
var empty = [];                          // Пустой массив
var numbers = [2, 3, 5, 7, 11];          // Массив с пятью числовыми элементами
var misc = [ 1.1, true, "a", ];          // 3 элемента разных типов + завершающая запятая
var base = 1024;
var table = [base, base+1, base+2, base+3];    // Массив с переменными
var arrObj = [[1,{x:1, y:2}], [2, {x:3, y:4}]];  // 2 массива внутри, содержащие объекты
```

Синтаксис литералов массивов позволяет вставлять необязательную завершающую запятую, т. е. литерал [,,] соответствует массиву с двумя элементами, а не с тремя.

Другой способ создания массива – конструктор Array(), который можно вызвать тремя разными способами: без аргументов; с единственным числовым аргументом, определяющим длину массива; с явным указанием значений первых двух или более элементов или одного нечислового элемента:

```
var arr = new Array();                  // пустой массив, эквивалентный литералу []
var arr = new Array(10);                // пустой массив указанной длины
var arr = new Array(5, 4, 3, 2, 1, "тест");   //массив с явным указанием элементов
```

Массив является специализированной разновидностью объекта, поэтому квадратные скобки, используемые для доступа к его элементам, действуют аналогично доступу к свойствам объекта. Интерпретатор JavaScript преобразует указанные в скобках числовые индексы в строки, например, индекс 1 – в строку "1", а затем использует эти строки как имена свойств.

Следует четко отличать индексы в массиве от имен свойств объектов. Все индексы массива являются именами свойств, но только свойства с именами, представленными целыми числами, являются индексами. Массивы являются объектами, и к ним можно добавлять свойства с любыми именами. Однако если затрагиваются свойства, которые являются индексами массива, то массивы реагируют на это, обновляя при необходимости значение свойства length.

В качестве индексов массивов допускается использовать отрицательные и нецелые числа. В этом случае числа преобразуются в строки, которые используются как имена свойств.

Добавление и удаление элементов массива. Самый простой способ добавления элементов массива – это присваивание значений его новым индексам. Для добавления одного или более элементов в конец массива можно также использовать метод push():

```
var arr = [];           // Создать пустой массив
arr.push('zero');       // Добавить значение в конец массива
arr.push('one',2);      // Добавить еще два значения в массив
```

Добавить элемент в конец массива можно также, присвоив значение элементу arr[arr.length]. Для вставки элемента в начало массива можно использовать метод unshift(), при этом существующие элементы в массиве смещаются в позиции с более высокими индексами.

Удалять элементы массива можно как обычные свойства объекта – с помощью оператора delete:

```
var arr = [1,2,'three'];
delete arr[2];
2 in arr;               // false, индекс 2 в массиве не определен
arr.length;             // 3: оператор delete не изменяет свойство length массива
```

Многомерные массивы. JavaScript не поддерживает настоящие многомерные массивы, но позволяет имитировать их при помощи массива из массивов. Для доступа к элементу данных в массиве массивов достаточно дважды использовать оператор [ ].

Методы класса Array:

Array.join() – преобразует все элементы массива в строки, объединяет их и возвращает получившуюся строку.

Array.reverse() – меняет порядок следования элементов в массиве на обратный и возвращает переупорядоченный массив.

Array.sort() – сортирует элементы в исходном массиве и возвращает отсортированный массив.

Array.concat() – создает и возвращает новый массив, содержащий элементы исходного массива, для которого был вызван метод concat(), и значения всех переданных ему аргументов.

Array.slice() – возвращает фрагмент, или подмассив указанного массива.

Порядок выполнения работы.

Выполнить задания, варианты которых приведены в таблице 1.6.

Таблица 1.6 – Варианты заданий

Номер варианта	Задание 1		Задание 2	Задание 3		Задание 4
	строки	k, m, n	m, n	сортировать	k, m	m, k
1	2	3	4	5	6	7
1	['a', 'b', 'c']	3, 2, 4	4, 6	По возрастанию	7, 8	6, 9
2	[1, 2, 3]	4, 3, 2	5, 8	По убыванию	6, 5	7, 6
3	['p', 'n', 'k']	2, 1, 5	3, 5	По возрастанию	8, 6	8, 9
4	[7, 5, 8]	4, 2, 3	4, 6	По убыванию	5, 8	5,7
5	['r', 's', 't']	3, 4, 4	5, 7	По возрастанию	6, 7	5, 8
6	[2, 6, 8]	5, 3, 2	4, 6	По убыванию	5, 8	5, 9
7	['k', 'm', 's']	4, 1, 3	5, 8	По возрастанию	8, 7	6, 5
8	[3, 7, 5]	2, 5,4	3, 6	По убыванию	6, 8	6, 7
9	['q', 'z', 's']	5, 1, 3	4, 5	По возрастанию	5, 6	6, 8
10	[2, 9, 4]	3, 4, 5	5, 7	По убыванию	8, 8	6, 6
11	['t', 'r', 'n']	4, 2, 3	2, 6	По возрастанию	7, 7	7, 6
12	[4, 7, 2]	3, 5,2	3, 8	По убыванию	6, 5	7, 7

1 Создать одномерный массив R из k элементов, указанных в колонке 2 таблицы 1.6. Добавить m числовых элементов в его начало и n текстовых – в конец.

2 Создать двухмерный массив размерностью m × n (колонка 4 таблицы 1.6) и заполнить случайными равномерно распределенными числами. Строки с четными номерами рассортировать.

3 Создать двухмерный массив размерностью k × m, содержащий текстовые и числовые элементы. Используя метод Array.join(), преобразовать все числовые элементы массива в строки; изменить порядок следования элементов массива на обратный.

4 Создать двухмерный текстовый массив размерностью m × k. Используя методы unshift() и push(), добавить k – 4 строк в начало массива и с помощью методов shift() и pop() удалить m – 3 строк из конца заданного массива.

Контрольные вопросы

1 Какие методы создания массивов Вы знаете?
2 Что такое динамический массив в JavaScript?
3 Что может быть элементом массива в JavaScript?
4 С какого значения ведется отсчет индексов массивов в JavaScript?
5 Как построить доступ к элементу массива?
6 Какой тип может иметь индекс элемента массива?
7 Как понимать: «массив – специализированная разновидность объекта»?
8 Какие методы используются для добавления и удаления элементов массива?
9 Как в JavaScript поддерживается работа с многомерными массивами?
10 Какие методы класса Array Вы знаете?

1.7 Лабораторная работа № 7. Работа с элементами управления на JavaScript

Язык JavaScript позволяет создавать сложные веб-элементы управления сайтами, среди которых сложные меню, специализированные деревья и сложные сетки, а также два специальных – генератор всплывающих окон и динамически меняющаяся кнопка.

Всплывающие (popup-) окна – это один из способов, позволяющих показать пользователю дополнительный контент.

В недавнем прошлом всплывающими окнами злоупотребляли многие сайты, нацеленные на показ рекламы, и загружали пользователей множеством объявлений. Поэтому современные браузеры блокируют всплывающие окна.

Всплывающее окно достаточно просто отображается с помощью функции window.open() в блоке JavaScript:

```
window.open('http://www.google.com', 'myWindow',
   'toolbar=0, height=500, width=800, resizable=1, scrollbars=1');
window.focus();
```

Функция window.open() принимает параметры: ссылка на новую страницу и имя фрейма окна, в которое позже должен быть загружен новый документ посредством другой ссылки. Третий параметр – разделенная запятыми строка, конфигурирующая стиль и размер всплывающего окна с помощью атрибутов:

– height – высота и width – ширина в пикселях;

– toolbar – панель инструментов и menuBar – строка меню, которые могут быть установлены в 1 или 0, в зависимости от того, требуется ли отображение этих элементов;

– resizable = 1 – рамка изменяемого размера, = 0 – фиксированного;

– scrollbars = 1, если требуются линейки прокрутки, = 0 – если нет.

Чтобы закрыть popup-окно, необходимо вызвать функцию newWindow.close(). Метод close() можно вызвать для любого объекта window, но window.close() игнорируется почти всеми браузерами, если окно было открыто не с помощью window.open().

Эффективным элементом управления в JavaScript является динамически меняющаяся кнопка, которая выводит на экран одно изображение, если она появляется на веб-странице впервые, при задержке над ней курсора мыши – другое, при щелчке на этой кнопке – третье.

Для обеспечения такого эффекта кнопка обычно состоит из дескриптора <img>, который обрабатывает JavaScript-события onclick, onmouseover и onmouseout. Эти события вызывают функции, меняющие изображения для текущей кнопки:

```
function swapImg(id, url) {
   var elm = document.getElementById(id);
   elm.src = url;
}
```

В этом случае сконфигурированный дескриптор <img> выглядел бы следующим образом, html:

```
<img id="img" src="activebtn.png" style="cursor:pointer; margin:120px"
   onmouseover="swapImg('img', 'hoverbtn.png')"
   onmouseout="swapImg('img', 'activebtn.png')" />
```

Порядок выполнения работы.

Выполнить задания, варианты которых приведены в таблице 1.7.

1 Написать сценарий, позволяющий продемонстрировать изменения размеров и положения горизонтальной линии на странице html.

2 Написать сценарий формирования анкеты данных сотрудника, указанных в колонке 3 таблицы 1.7.

3 Написать сценарий обработки анкеты слушателя курсов повышения квалификации, содержащей: курс (первый, второй, третий); язык общения с преподавателем; изучаемые дисциплины; продолжительность курса; форму образования (очная, заочная, дистанционная); изучаемые дисциплины; стоимость. В зависимости от этих параметров определяется стоимость отдельного курса и стоимость всего обучения. Выбор значений выполнить с помощью флажков и выпадающего меню.

Таблица 1.7 – Варианты заданий

Номер варианта	Задание 1: длина линии, %; толщина, (пикс)	Задание 2 (данные сотрудника): МР – место рождения; НАЦ – национальность; ОБР – образование; ДР – дата рождения; СП –семейное положение	Задание 3 (поля выбора): ПР – продолжительность; ФО – форма образования; ФИОП – ФИО преподавателя; ИД – изучаемые дисциплины
1	2	3	4
1	45%; 2	Пол, МР, ОБР	Курс, язык, ПР, стоимость
2	72%, 3	ДР, пол, должность	Курс, язык, ФО,
3	85%, 4 пикс	МР, возраст, пол	Курс, язык, ФИО П, стоимость
4	57%, 2 пикс	МР, должность, пол	Курс, язык, ПР, ИД
5	80%, 3 пикс	Пол, должность, ДР	Курс, язык, ФО, стоимость
6	90%, 2 пикс	НАЦ, пол, возраст	Курс, язык, ФИОП, ИД
7	50%, 2 пикс	СП, пол, должность	Курс, язык, ФО, стоимость
8	70%, 3 пикс	Пол, НАЦ, возраст	Курс, язык, ПР, ИД
9	80%, 4 пикс	МР, возраст, должность	Курс, язык, ФИОП, стоимость
10	35%, 2 пикс	ДР. должность, возраст	Курс, язык, ПР, ИД
11	45%, 3 пикс	НАЦ, должность, образование	Курс, язык, ФО, стоимость
12	75%, 4 пикс	МР, возраст, должность	Курс, язык, ФИОП, ИД

Контрольные вопросы

1 Какие веб-элементы управления сайтами Вы знаете?
2 Для чего используется генератор всплывающих (popup-) окон?
3 Что такое динамически меняющаяся кнопка?
4 Почему современные браузеры блокируют всплывающие окна?
5 Для чего используется функция window.open()?
6 Какие параметры содержит функция window.open()?
7 Как закрыть popup-окно?
8 Где используется и для чего свойство z-index?
9 Как работают динамически изменяющиеся кнопки?
10 Как построить динамически изменяющуюся кнопку?

1.8 Лабораторная работа № 8. Работа с формами на JavaScript

html-формы используются для пересылки данных от удаленного пользователя к веб-серверу. С их помощью можно организовать простейший диалог между пользователем и сервером, например, регистрацию пользователя на сервере или выбор нужного документа из представленного списка. Формы поддерживаются всеми популярными браузерами.

Для идентификации формы и ее элементов через JavaScript можно использовать два атрибута: ***name*** и ***id***.

Атрибут id используется, если не требуется отправлять данные с формы на сервер, а атрибут name – для отправки формы на сервер. При этом все элементы управления обязательно располагаются в форме.

Если в html-документе определена форма, то она доступна сценарию JavaScript как объект, входящий в объект document с именем, заданным атрибутом name тега form.

Форма имеет два набора свойств, состав одного из которых фиксированный, а другого зависит от определенных в форме элементов.

Флажок (checkbox). Свойства: name – имя объекта; value – надпись на кнопке; checked – состояние флажка (true – флажок установлен, false – не установлен); defaultChecked – отражает наличие атрибута Checked (true – есть, false – нет); метод: click( ) – меняет состояние флажка, аналогичен щелчку мышкой по кнопке.

Переключатель radio. Свойства: name – имя объекта; value – надпись на кнопке: length – количество переключателей в группе; checked – состояние переключателя: true – включен, false – выключен; defaultChecked – отражает наличие атрибута checked: true – есть, false – нет; метод: click( ) – включает переключатель. Так как группа переключателей имеет одно имя name, то адресоваться к ним необходимо как к элементам массива.

Список (select). Свойства: name – имя объекта; selectedIndex – номер выбранного элемента или первого среди выбранных (если указан атрибут multiple); length – количество элементов (строк) в списке; options – массив элементов списка, заданных тегами option; методы: focus( ) – передает списку фокус ввода; blur( ) – отбирает у списка фокус ввода.

Каждый элемент массива options является объектом со следующими свойствами: value – значение атрибута Value; text – текст, указанный после тега Option; index – индекс элемента списка; selected – присвоив этому свойству значение true, можно выбрать данный элемент; defaultSelected – отражает наличие атрибута Selected (true – есть, false – нет).

Кроме работы с готовыми списками, JavaScript может заполнять список динамически. Для записи нового элемента списка используется конструктор Option с четырьмя параметрами, первый из которых задает текст, отображаемый в списке, второй – значение элемента списка, соответствующее значению атрибута Value, третий соответствует свойству defaultSelected, четвертый – свойству selected.

Порядок выполнения работы.

Создать html-страницу с формой, содержащей элементы, описания которых приведены в таблице 1.8.

Таблица 1.8 – Варианты заданий

Номер варианта	Описание формы
1	2
1	Центр изучения потребительского спроса собирает информацию о потребляемых соках в разрезе городов с населением более 1 миллиона. Используются следующие поля: фирма-производитель; название; концентрация (не более 100 % и не менее 50 %); цена; способ приготовления (переключатель), например: из концентрированного сока, из сухих материалов и т. д., содержание заменителя сахара (флажок). В случае использования заменителя сахара активизируется поле для записи его наименования. Кроме того, определена категория сока (переключатель: нектар/сок)
2	Создать форму, которую можно использовать для размещения не анонимных объявлений в глобальной сети Интернет с возможностью дальнейшего удаления автором объявления (для этого используется специальное поле-пароль). Обязательным параметром является адрес электронной почты автора объявления, который должен содержать символ @ и хотя бы одну точку
3	Пейджинговая компания желает реализовать возможность отправления сообщений абонентам через Интернет. Для этого используется следующая форма: время отправления сообщения; дата отправления сообщения, текст сообщения; номер абонента (четырехзначное число); количество повторов (1–5); автор; роуминг (флажок) по городам (список из пяти городов)
4	Создать форму для поиска книг каталога библиотеки, содержащую поле для ввода названия книги, автора, года издания. Предусмотреть поиск книг по автору, по названию книги. Для выбора темы поиска реализовать выпадающий список
5	Создать форму для заполнения электронной записной книжки, содержащую следующие поля: фамилия, имя, отчество, домашний адрес, город (выпадающий список из 10 городов), код местности (трехзначное число), номер телефона (может быть семизначным, шестизначным или пятизначным), использовать список типов контакта (друг, знакомый, коллега и т. д.). Рассмотреть 5–10 городов с разными масками телефонов
6	Один из сайтов знакомств планирует усовершенствовать свою деятельность. Для этого создана новая структура базы данных и необходимо создать форму, на основе которой будет создаваться запрос. В форме должны быть предусмотрены: информация о подающем запрос и информация об искомом человеке (две категории). Туда входят: пол (возможность выбора), возраст, цвет глаз, цвет волос (выпадающий список), рост, вес, телосложение (выпадающий список), вредные привычки (флажки)
7	Создать форму для заказов билетов на поезда, содержащую следующие поля: номер поезда; категория поезда (переключатель, два состояния – поезд местного формирования и проходящий); направление поезда (выпадающий список, например: Могилев – Минск, Могилев – Орша); название поезда (например: «Красная стрела»); количество билетов (не больше 4); класс желаемых вагонов (элементы списка: люкс, спальный, купе, плацкарт, общий)
8	Создать форму для перевода денег кредитными картами, которая должна содержать следующие поля: номер счета; номер кредитной карты; код карты (три цифры); дата окончания срока действия карты (два выпадающие списка с месяцем и годом); выбор валюты – переключатель (USD, EURO, РУБ). Форма также должна содержать кнопку «Перевести»

Окончание таблицы 1.8

1	2
9	Создать форму для дилерской регистрации на сайте торговой фирмы. Форма должна содержать поля: наименование фирмы дилера; город (поле с выпадающим списком областных центров Беларуси); адрес; телефон, E-mail, веб-сайт. В полях E-mail и веб-сайт предусмотреть автоматическое добавление символов @ и http://
10	Создать форму для регистрации аккаунта на сервере знакомств, которая должна содержать следующие поля: имя, фамилия; пароль; подтверждение пароля; секретный вопрос; ответ; страна (выпадающий список из 10 стран). Также форма должна содержать независимый переключатель «Запомнить меня на этом компьютере»
11	Создать форму для регистрации пользователей на интернет-курсы по общеобразовательным предметам. Форма должна содержать поля: фамилия, имя, отчество; дата рождения; адрес; переключатели с выбором м/ж пола; Checkbox для выбора общеобразовательных предметов, интересующих пользователя; выпадающий список для выбора образования
12	Сотовая компания решила организовать проект по анализу рейтинга популярности тарифных планов. Требуется реализовать форму, на которой должна содержаться информация: фамилия, имя, отчество абонента; телефонный номер абонента; тарифный план абонента – выпадающий список; сумму денежных средств, затраченных на оплаты телефонных разговоров; независимый переключатель «хотите поменять тарифный план»; поле «Пожелание компании»

Контрольные вопросы

1 Для чего используется html-формы?
2 Как организовать ввод информации в форму на JavaScript?
3 Какие существуют способы передачи данных формы?
4 Для чего предназначен метод focus()?
5 В каких целях используется программа обработки событий onSubmit?
6 Как использовать select box как навигационное меню?
7 Как использовать картинку для кнопки submit?
8 Как передавать данные между формами на различных страницах, используя JavaScript?
9 Почему document.formName.selectObject.value не отражает значение выбранного пункта в списке?
10 Как получить значение выбранной в данный момент radio button в radio group или группе checkboxes?

1.9 Лабораторная работа № 9. Работа с изображениями на JavaScript

Изображения в html-документах представляются в виде массива, что позволяет адресоваться к ним. Они имеют определенные свойства и рассматриваются как объекты image, к которым можно обращаться из языка JavaScript. Например, можно определить размер изображения, обратившись к его

свойствам width и height. То есть по записи document.images[0].width можно определить ширину первого изображения в пикселях на веб-странице.

Объект Image позволяет вносить изменения в графические образы на веб-странице, что позволяет создавать мультипликацию. Для отслеживания индекса всех изображений веб-страницы следует назначить им имена с помощью тега

```
<img src="img.gif" name="myImage" width=100 height=100>
```

Тогда для обращения к изображению необходимо написать document.myImage или document.images["myImage"].

Смена изображения на веб-странице выполняется с использованием атрибута src, который содержит адрес представленного изображения, и позволяет назначить ему новый адрес. В результате изображение будет загружено с этого нового адреса, заменяя на веб-странице старое:

```
<img src="img1.gif" name="myImage" width=100 height=100>
```

Здесь загружается изображение img1.gif и получает имя myImage. В следующей строке изображение img1.gif заменяется на новое – img2.gif:

```
document.myImage.src= "img2.src".
```

При этом новое изображение всегда получает тот же размер, который был у старого, и уже невозможно будет изменить размер поля, в котором это изображение размещается.

Одной их замечательных возможностей браузеров являются слои, позволяющие позиционировать изображения, организовывать их перемещение и делать их невидимыми.

Для создания слоев можно использовать тег <layer> или тег <ilayer>. При этом можно воспользоваться следующими параметрами:

name = "layerName" – название слоя;
left = xPosition – абсцисса левого верхнего угла;
top = yPosition – ордината левого верхнего угла;
z-index = layerIndex – номер индекса для слоя;
width = layerWidth – ширина слоя в пикселах;
clip = "x1_offset, y1_offset, x2_offset, y2_offset" – видимая область слоя;
above = "layerName" – определяет, какой слой окажется под текущим;
below = "layerName" – определяется, какой слой окажется над текущим;
visibility= show|hide|inherit – видимость этого слоя;
bgcolor = "rgbColor" – цвет фона, или название стандартного цвета, или rgb-запись;
background = "imageURL" – фоновая картинка.

Тег <layer> используется для слоев, которые можно точно позиционировать. Если не указать положение слоя с помощью параметров left и top, то

по умолчанию он помещается в верхний левый угол окна.

Тег <ilayer> создает слой, положение которого определяется при формировании документа.

Поверх изображения или под ним можно показать текст.

Свойства слоев можно изменять с помощью скриптов на JavaScript. Например, в следующей строке задается горизонтальное положение слоя, смещенное на 200 пикселей:

```
document.layers["myLayer2"].left = 200;
```

Следующий пример демонстрирует, как скрипт может реагировать на сигналы о нажатии клавиш.

```
<html>
<script language="JavaScript">
window.captureEvents(Event.KEYPRESS);
window.onkeypress= pressed;
function pressed(e) {
  alert("Key pressed! ASCII-value: " + e.which);       }
</script>
</html>
```

Порядок выполнения работы.

Создать html-документ, элементы и параметры которого указаны в таблице 1.9.

Таблица 1.9 – Варианты заданий

Номер варианта	Количество графических элементов (слоев)	Ориентация группы элементов	Изображения меняются местами	При наведении курсора
1	3	Горизонтально	При наведении курсора	Увеличить на 25 %
2	4	Горизонтально	При наведении курсора	Увеличить на 40 %
3	3	Вертикально	По щелчку мыши	Уменьшить на 30 %
4	4	Вертикально	По щелчку мыши	Уменьшить на 40 %
5	3	Горизонтально	При наведении курсора	Увеличить на 35 %
6	4	Горизонтально	При наведении курсора	Увеличить на 45 %
7	3	Вертикально	По щелчку мыши	Уменьшить на 35 %
8	4	Вертикально	По щелчку мыши	Уменьшить на 50 %
9	3	Горизонтально	При наведении курсора	Увеличить на 30 %
10	4	Горизонтально	При наведении курсора	Увеличить на 45 %
11	3	Вертикально	По щелчку мыши	Уменьшить на 20 %
12	4	Вертикально	По щелчку мыши	Уменьшить на 45 %

Контрольные вопросы

1 Для чего предназначен объект Image?
2 Как в html-документах представляются изображения?
3 Как из JavaScript можно адресоваться к изображениям?
4 Для чего предназначен атрибут src тега <img>?
5 Какие условия следует соблюдать, чтобы скрипт смены изображений сохранял свою гибкость?
6 Какие свойства определяют размеры объекта image?
7 Какой объект позволяет вносить изменения в графические образы на веб-странице для создания мультипликации?
8 Что позволяет делать изображения в браузерах невидимыми?
9 Какой тег используется для создания слоев?
10 Как изменить свойство слоя?

1.10 Лабораторная работа № 10. Обработка событий на JavaScript

Событие – это сигнал от браузера о том, что что-то произошло, например, load – загрузка страницы и unload – выгрузка ее. Клиентские программы на языке JavaScript основаны на модели программирования, когда выполнение программы управляется событиями. При таком стиле программирования веб-браузер генерирует событие, когда с документом или некоторым его элементом что-то происходит. Например, веб-браузер генерирует событие, когда завершает загрузку документа, когда пользователь наводит указатель мыши на гиперссылку или нажимает клавишу на клавиатуре.

События мыши: click – клик; dblclick – двойной клик; mouseover – наведение курсора мыши на элемент; mousemove – перемещение курсора мыши над элементом; mouseout – уведение курсора мыши с элемента; mousedown – нажатие левой кнопки мыши; mouseup – отпускание левой кнопки мыши; contextmenu – нажатие правой кнопки мыши и вывод контекстного меню.

Для того чтобы обратить внимание пользователя на определённый элемент html-документа, можно менять свойства этого элемента при попадании на него курсора мышки, а при снятии курсора восстанавливать прежние значения свойств. Например, можно менять цвет или размер элемента. Попадание курсора мышки на элемент фиксируется событием onMouseOver. Парное для него событие onMouseOut происходит при снятии курсора мышки с элемента.

Другая пара событий onMouseDown и OnMouseUp происходит при нажатии и отпускании левой кнопки мышки. Эту пару событий удобно применять для изменения свойств элементов или замены элементов на время удержания кнопки мышки нажатой.

События клавиатуры: keyup – возникает при отпускании клавиши клавиатуры; keydown – возникает при нажатии клавиши клавиатуры и длится, пока нажата клавиша; keypress – возникает при нажатии клавиши клавиатуры, но после события keydown и до события keyup.

Для того чтобы написать ответную реакцию на событие, создают обработчик события (event handler), который, как правило, представляет собой функцию.

Назначить обработчик события можно в виде атрибута элемента, безымянной функции, именованной функции или с помощью метода addEventListener().

Тип события – это строка, определяющая тип действия, вызвавшего событие. Тип mousemove, например, означает, что пользователь переместил указатель мыши, тип keydown – что была нажата клавиша на клавиатуре, а тип load – что завершилась загрузка документа или какого-либо другого ресурса из сети.

Обработчик события – это функция, которая обрабатывает событие или откликается на него. Приложения должны зарегистрировать свои функции обработчиков событий в веб-браузере, указав тип события и цель.

Реакция на событие в отдельном элементе. Так как в объектной модели объекты могут быть вложены друг в друга, то событие, происходящее в дочернем объекте, одновременно происходит и в родительском. JavaScript предоставляет различные способы локализации влияния события на иерархию объектов. Простейшей способ локализации заключается в размещении сценария в теге, на который должно воздействовать событие.

Порядок выполнения работы.

Создать html-документ, содержащий строку и изображение с подписью, параметры которого указаны в таблице 1.10.

Таблица 1.10 – Варианты заданий

Номер варианта	Создать строку, при наведении курсора на которую меняется			Создать изображение с надписью на нем и подписью	
	шрифт	цвет шрифта	цвет фона	При щелчке на изображении меняется	Цвет при щелчке на подписи
1	Увеличить до 48 pt	Белый	Голубой	Изображение и надпись	Синий
2	Увеличить до 64 pt	Синий	Белый	Цвет шрифта и надпись	Красный
3	Увеличить до 36 pt	Красный	Голубой	Цвет фона и надпись	Зеленый
4	Увеличить до 42 pt	Зеленый	Белый	Цвет шрифта и изображение	Голубой
5	Увеличить до 52 pt	Синий	Зеленый	Цвет фона и изображение	Желтый
6	Увеличить до 56 pt	Белый	Голубой	Фон и цвет надписи	Зеленый
7	Уменьшить на 12 pt	Синий	Белый	Изображение и надпись	Синий
8	Уменьшить на 16 pt	Красный	Голубой	Цвет шрифта и надпись	Желтый
9	Уменьшить на 24 pt	Зеленый	Белый	Цвет фона и надпись	Красный
10	Уменьшить на 20 pt	Синий	Зеленый	Цвет шрифта и изображение	Голубой
11	Уменьшить на 10 pt	Красный	Голубой	Цвет фона и изображение	Зеленый
12	Уменьшить на 14 pt	Зеленый	Белый	Фон и цвет надписи	Синий

Контрольные вопросы

1 Какие виды событий используются в JavaScript?
2 Какие события мыши Вы знаете?
3 Как назначить обработчик событий?
4 Какие события JavaScript относятся к системным?
5 Какие события клавиатуры Вы знаете?
6 Что понимается под типом события?
7 Что означают типы событий mousemove и keydown?
8 Что понимается под целью события?
9 Что такое обработчик событий?
10 Что означает тип события load?

1.11 Лабораторная работа № 11. Основные методы JQuery

Методы jQuery позволяют манипулировать содержимым веб-страницы. Они присваивают элементам, отобранным в jQuery-объектах, заданные действия. В результате этого происходит динамическое изменение элементов и их содержимого.

Каждый метод jQuery либо сам что-либо возвращает, либо получает параметр и выполняет указанные в параметре действия.

В общем виде синтаксис для вызова метода jQuery имеет следующий вид:

```
$("селектор").имяМетода(параметры);
```

Динамическое изменение элементов веб-страниц. Библиотека jQuery упрощает процесс отбора элементов html-страниц. С помощью методов jQuery производятся манипуляции с объектной моделью документа DOM. Чтобы отобрать группу элементов, нужно передать селектор функции jQuery. В качестве селектора элемента может выступать сам элемент, его идентификатор или класс, а также комбинация селекторов: $("a"); $("#some-id"); $(".someclass"); $("header > ul:has(a)").

Функция $() возвращает объект jQuery, содержащий массив элементов DOM – так называемый обернутый набор, соответствующий указанному селектору. Большинство методов по завершении действий возвращает первоначальный набор элементов.

JQuery – это одна из наиболее известных библиотек, написанная на языке JavaScript для упрощения программирования веб-страниц. Это файл с расширением .js, который подключается к веб-странице как фрагмент скрипта и загружается в браузер вместе с веб-страницей.

Чтобы включить JQuery в веб-страницу, достаточно скачать последнюю версию библиотеки, например, файл jquery-1.9.1.js с сайта jquery.com, положить его в ту же папку, где лежит текст веб-страницы, а в текст веб-страницы вставить <script src="jquery-1.9.1.js"></script>

Файл jquery-1.9.1.js при этом имеет объем более 200 Кбайт, что может

замедлять загрузку веб-страницы в браузере пользователя.

Если веб-страница маленькая и важно, чтобы все «летало» и загрузка библиотеки JQuery ничего не замедляла, то существует альтернативный метод загрузки JQuery – с сайта Google:

```
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">
</script>
```

Второе важное характерное для jQuery применение состоит в создании AJAX элементов, т. е. тех элементов страницы, которые отсылают на сервер данные и получают ответ без перезагрузки страницы. К таким элементам можно отнести форму «Управление корзиной для интернет-магазина», пагинацию (нумерацию страниц сайта), вывод информера погоды и многое другое.

Порядок выполнения работы.

Используя библиотеку Jquery, создать сайт. Варианты заданий указаны в таблице 1.11.

Таблица 1.11 – Варианты заданий

Номер варианта	Используя библиотеку Jquery, создать сайт, содержащий
1	Выпадающее меню
2	Плавающее меню
3	Фото галерею
4	Всплывающие окна
5	Слайдеры (блоками на странице, в пределах которых с установленной периодичностью происходит демонстрация анонсов новостей, статей или изображений)
6	Перемещающиеся блоки
7	Изменение прозрачности элементов
8	Подсвечивание текста
9	Переливание цвета текста разными оттенками
10	Плавающее меню
11	Всплывающие окна
12	Изменение прозрачности элементов

Контрольные вопросы

1 Что такое JQuery?
2 Что такое слайдер?
3 Какие действия выполняют методы JQuery?
4 Прокомментируйте синтаксис вызова методов jQuery.
5 Что понимается под селектором jQuery?
6 Какие основные правила использования jQuery Вы знаете?
7 Как подключить библиотеку jQuery к веб-странице?
8 Где найти файл библиотеки jQuery?
9 Что такое альтернативный метод загрузки jQuery?
10 Какие основные методы jQuery Вы знаете?

1.12 Лабораторная работа № 12. Основные события JQuery

События jQuery, представляющие собой момент, в который что-либо происходит, например щелчок кнопки мыши, помогают сделать веб-страницы интерактивными, реагирующими на простейшие действия пользователя.

Момент, в который произошло событие, называется запуском события. События могут срабатывать при выполнении различных операций с веб-страницей. Помимо этого, и сам браузер может стать источником событий.

Управление веб-страницей производится с помощью следующих событий: мыши; документа/окна; форм; клавиатуры; jQuery.

События мыши:

.click() – запускается при нажатии и отпускании кнопки мыши, применяется к ссылкам, картинкам, кнопкам, абзацам, блокам и т. д.;

.dblclick() – запускается при двойном нажатии и отпускании кнопки мыши, например, при открытии какой-либо папки;

.mousedown() – происходит во время нажатия кнопки мыши, например, при перетаскивании элементов;

.mousemove() – запускается при перемещении указателя мыши по элементу;

.mouseout() – запускается при отпускании кнопки мыши.

События документа/окна:

.load() – запускается, когда браузер загрузит все файлы веб-страницы: html-файлы, внешние css- и Javascript-файлы, медиафайлы;

.resize() – запускается, когда пользователь изменяет размер окна браузера;

.scroll() – запускается, когда пользователь использует полосы прокрутки, либо прокручивает веб-страницу с помощью колесика мыши, либо использует для этих целей клавиши клавиатуры (pgup, pgdn, home, end);

.unload() – запускается, когда пользователь собирается покинуть страницу, щелкая по ссылке для перехода на другую страницу, закрывает вкладку страницы или окно браузера.

События форм:

.blur() – запускается, когда поле формы выводится из фокуса, например, при переходе в другое поле формы;

.change() – запускается при изменении статуса поля формы, например при выборе пункта из выпадающего меню;

.focus() – запускается при переходе в поле формы при щелчке на нем кнопкой мыши или клавишей табуляции;

.reset() – позволяет вернуть форму в первоначальное состояние, отменив сделанные изменения;

.select() – запускается при выделении текста внутри текстового поля формы;

.submit() – запускается при отправлении заполненной формы с помощью щелчка по кнопке «Отправить» или нажатии клавиши «Enter», когда курсор помещен в текстовом поле.

События клавиатуры:

.keydown() – запускается при нажатии клавиши перед событием keypress.

.keypress() – запускается при нажатии на клавишу до тех пор, пока клавиша не будет отпущена;

.keyup() – запускается при отпускании клавиши.

События jQuery:

.hover() – позволяет одновременно решить две задачи, связанные с событием наведения указателя мыши и событием снятия указателя мыши в отношении выбранного объекта;

.toggle() – работает аналогично событию hover() с разницей в том, что оно запускается от щелчка кнопкой мыши. Например, можно открыть выпадающее меню одним щелчком и скрыть вторым.

Объект события: при запуске события браузер сохраняет информацию о нём в объекте события, который содержит данные, собранные в момент, когда событие произошло. Обработка события происходит с помощью функции, при этом объект передается функции как аргумент-переменная evt.

Объект события имеет различные свойства, наиболее распространённые из которых следующие:

pageX – расстояние (px) от указателя мыши до левого края окна браузера;

pageY – расстояние (px) от указателя мыши до верхнего края окна браузера;

screen – расстояние (px) от указателя мыши до левого края монитора;

screenY – расстояние (px) от указателя мыши до верхнего края монитора;

shiftKey – TRUE, если была нажата клавиша SHIFT, когда происходило событие;

which – используется для определения числового кода нажатой клавиши (вместе с shiftKey);

target – означает, что по объекту события щелкнули кнопкой мыши (например, для события click());

data – объект, использованный с функцией bind() для передачи данных функции, управляющей событием.

Порядок выполнения работы.

Используя события, указанные в таблице 1.12, и библиотеку Jquery, создать сайт.

Таблица 1.12 – Варианты заданий

Номер варианта	Используя библиотеку Jquery, создать сайт, содержащий
1	События мыши .click() и .mousedown()
2	События мыши .dblclick() и .mousemove()
3	События документа/окна .load() и .scroll()
4	События документа/окна .resize() и .unload()
5	События форм. blur(), .focus() и .select()
6	События форм .change(), .reset() и .submit()
7	События клавиатуры .keydown() и .keyup()
8	События клавиатуры .keypress() и .keyup()

Окончание таблицы 1.12

Номер варианта	Используя библиотеку Jquery создать сайт, содержащий
9	Событие jQuery .hover()
10	Событие jQuery .toggle()
11	События мыши .click() и .mousemove()
12	События форм. blur(), .focus() и .submit()

Контрольные вопросы

1 Что представляют собой события jQuery?
2 С помощью каких событий производится управление веб-страницей?
3 Какие события мыши Вы знаете?
4 Что такое объект события?
5 Какие события документа/окна Вы знаете?
6 Какие события клавиатуры Вы знаете?
7 Какие события форм Вы знаете?
8 Какие события jQuery Вы знаете?
9 В каком объекте браузер сохраняет информацию о событии при его запуске?
10 Какие свойства объекта события Вы знаете?

2 Серверное программирование

2.1 Лабораторная работа № 13. Установка локального сервера

При разработке и отладке серверных приложений используются локальные сервера, наиболее популярным среди которых является веб-сервер Apache.

Связь внешней программы с веб-сервером выполняется с использованием CGI (Common Gateway Interface – общий интерфейс шлюза), а программу, позволяющую использовать консоль ввода и вывода для взаимодействия с клиентом и работающую по интерфейсу CGI, принято называть шлюзом, но используется также и название «скрипт» (сценарий) или «CGI-программа».

Для разработки серверных приложений на языке PHP используется интерпретатор PHP, который представляет собой либо внешнюю CGI-программу, либо динамическую библиотеку, которую необходимо подключить к веб-серверу, чтобы вместо кода PHP-скриптов клиенту выдавались результаты ее выполнения.

Традиционно совместно с веб-сервером Apache и интерпретатором PHP используется СУБД MySQL.

Так как и язык программирования PHP, и веб-сервер Apache, и MySQL-сервер первоначально разработаны для UNIX-подобных операционных систем, их настройка и администрирование сводятся к редактированию конфигура-

ционных файлов. Такой подход часто сбивает с толку программистов, не имеющих опыта работы в UNIX. Чтобы не завязнуть в многочисленных настройках серверов, правилах их связывания друг с другом, веб-разработчики прибегают к готовым пакетам, где вся настройка выполнена профессиональными администраторами. Остается только загрузить и установить готовый пакет, представляющий собой CMS (Content Management System – система управления содержимым), после чего можно сразу приступить к работе с языком программирования. Популярными пакетами, объединяющими PHP, веб-сервер Apache и СУБД MySQL, являются WordPress, Drupal, Joomla, DataLife Engine. Wamp, Mamp и другие.

Самой популярной CMS, распространяемой по открытому лицензионному соглашению, является WordPress. По данным веб Technology Surveys, на этом движке по состоянию на ноябрь 2018 г. разработано 32,3 % от общего числа существующих сайтов, а также 59,5 % сайтов, использующих CMS. С помощью WordPress можно создать интернет-магазин, личный блог, корпоративный сайт, информационный портал, отраслевой ресурс, галерею мультимедиа и др.

Принципы построения сайтов в среде WordPress понятны на интуитивном уровне. После создания и настройки сайта необходимо опубликовать контент, а чтобы сайт был эффективным, контент должен быть качественным и полезным для аудитории, поэтому его необходимо регулярно обновлять, что является самой сложной и ответственной работой.

Одним из наиболее простых в использовании и легко настраиваемых CMS является также MAMP, который доступен по адресу https://www.mamp.info/en/downloads/

После скачивания MAMP необходимо кликнуть на иконку установочного пакета, чтобы запустить процесс распаковки и установки mamp на компьютер. После всех успешных действий установки появится диалоговое окно локального сервера (рисунок 2.1).

Рисунок 2.1 – Диалоговое окно MAMP

Особого внимания здесь заслуживает ссылка с шестеренкой и надписью Preferences – Настройки и привилегии, активизация которой выводит окно с пятью вкладками (рисунок 2.2).

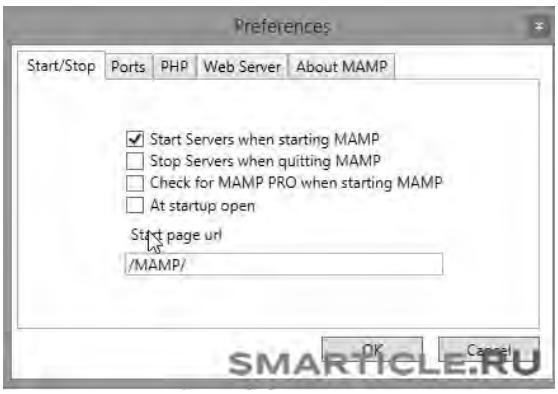

Рисунок 2.2 – Меню Preferences

После успешных настроек должны загореться два пункта зеленым цветом – Apache и MySql (см. рисунок 2.1), подтверждающих, что сервер работает.

Далее следует перейти на стартовую страницу, нажав на ссылку Open Start page, после чего должен открыться браузер, в адресной строке которого появится локальный путь localhost/MAMP, по которому будет выполняться обращение к файлам сайта.

После этого следует перейти в навигационное меню, в котором необходимо отметить только один раздел Tools (Инструментарий), где расположена ссылка для доступа в phpMyAdmin.

Вторая важная вкладка необходима для разрешения конфликта между Скайпом – Ports (Порты): порт Апач – 80, MySql-порт – 3306.

Остальные вкладки можно не редактировать.

Далее, после проведенных настроек, можно запустить сервер, нажав на ссылку Start Servers или на ссылку Open Start page, перейти на стартовую страницу и открыть браузер (рисунок 2.3).

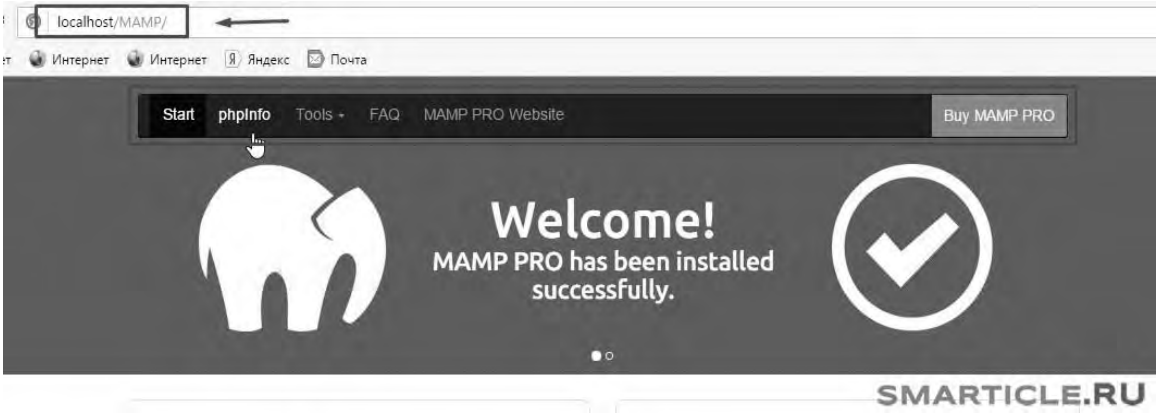

Рисунок 2.3 – Окно браузера

Здесь в адресной строке прописан локальный путь, по которому будет выполняться обращение к файлам сайта – localhost/MAMP.

Далее следует навигационное меню, в котором интересен только один раздел Tools (Инструментарий). Именно здесь расположена ссылка для доступа в phpMyAdmin.

Порядок выполнения работы.

Выполнить установку и настройку веб-сервера Apache, интерпретатора PHP и СУБД MySQL на свой компьютер. Для этого ознакомиться с установкой и настройкой CMS MAMP,

Контрольные вопросы

1 Что такое локальный веб-сервер?
2 Для чего используется локальный веб-сервер Apache?
3 Что такое CMS?
4 Как установить CMS WordPress?
5 Как создать статическую страницу в WordPress?
6 Для чего используются пакеты WAMP и MAMP?
7 Как настроить CMS WordPress и тему?
8 Какие пакеты, объединяющие PHP, веб-сервер Apache и СУБД MySQL Вы знаете?
9 Можно ли создать сайт без знания html и языка PHP?
10 Как выбрать хостинг и зарегистрировать домен?

2.2 Лабораторная работа № 14. Изучение строковых функций языка PHP

В языке PHP используются три способа задания строк: с помощью одинарных кавычек, двойных кавычек и с использованием heredoc-синтаксиса.

Строки, содержащие заключенные в одинарные кавычки переменные и управляющие последовательности специальных символов, не обрабатываются.

Важнейшим свойством строк в двойных кавычках является обработка содержащихся в них переменных.

Определение строк с использованием heredoc-синтаксиса начинается с символа <<<, после которого следует идентификатор. Заканчивается строка этим же идентификатором, который должен начинаться с первой позиции новой строки.

Heredoc-текст ведет себя так же, как и строка в двойных кавычках. Это означает, что в heredoc нет необходимости экранировать кавычки, но можно использовать управляющие последовательности. Переменные внутри heredoc также обрабатываются.

Для работы со строками в PHP имеется более ста функций (таблица 2.1).

Таблица 2.1 – Некоторые функции обработки строк

Номер функции	Функция
1	2
1	chunk_split – разбивает строку на фрагменты
2	echo – выводит одну или более строк

Продолжение таблицы 2.1

1	2
3	explode – разбивает строку с помощью разделителя
4	implode – объединяет элементы массива в строку
5	lcfirst – преобразует первый символ строки в нижний регистр
6	ltrim – удаляет пробелы или другие символы из начала строки
7	print – выводит строку
8	printf – выводит отформатированную строку
9	rtrim – удаляет пробелы или другие символы из конца строки
10	similar_text – вычисляет степень похожести двух строк
11	sprintf – возвращает отформатированную строку
12	sscanf – разбирает строку в соответствии с заданным форматом
13	stripos – возвращает позицию первого вхождения подстроки без учета регистра
14	strlen – возвращает длину строки
15	str_pad – дополняет строку другой строкой до заданной длины
16	str_replace – заменяет все вхождения строки поиска на строку замены
17	str_shuffle – переставляет символы в строке случайным образом
18	str_split – преобразует строку в массив
19	str_word_count – возвращает информацию о словах, входящих в строку
20	strip_tags – удаляет теги html и PHP из строки
21	strpbrk – ищет в строке любой символ из заданного набора
22	strpos – возвращает позицию первого вхождения подстроки
23	strrchr – находит последнее вхождение символа в строке
24	strrev – переворачивает строку задом наперед
25	stripos – возвращает позицию первого вхождения подстроки без учета регистра
26	strlen – возвращает длину строки
27	strpbrk – ищет в строке любой символ из заданного набора
28	strpos – возвращает позицию первого вхождения подстроки
29	strrchr – находит последнее вхождение символа в строке
30	strrev – переворачивает строку задом наперед
31	strripos – возвращает позицию последнего вхождения подстроки без учета регистра
32	strrpos – возвращает позицию последнего вхождения подстроки в строке
33	strspn – возвращает длину участка в начале строки, соответствующего маске
34	strstr – находит первое вхождение подстроки
35	strtok – разбивает строку на токены
36	strtolower – преобразует строку в нижний регистр
37	strtoupper – преобразует строку в верхний регистр
38	strtr – преобразует заданные символы или заменяет подстроки
39	substr_count – возвращает число вхождений подстроки
40	substr_replace – заменяет часть строки

Окончание таблицы 2.1

1	2
41	substr – возвращает подстроку
42	trim – удаляет пробелы или другие символы из начала и конца строки
43	ucfirst – преобразует первый символ строки в верхний регистр
44	ucwords – преобразует в верхний регистр первый символ каждого слова строки
45	vfprintf – записывает отформатированную строку в поток
46	vprintf – выводит отформатированную строку
47	vsprintf – возвращает отформатированную строку
48	wordwrap – переносит строку по указанному количеству символов
49	serialize() – создает строковое представление текущего состояния массива или объекта
50	unserialize() – восстанавливает состояние массива / объекта из строки

Порядок выполнения работы.

1 Работа со строками:

– определить строку с использованием синтаксиса одинарных кавычек;

– определить строку с использованием синтаксиса двойных кавычек;

– определить строку с использованием heredoc-синтаксиса;

– создать массив из трех-пяти элементов, вывести его с использованием echo, print, print_r, serialize и пояснить полученные результаты.

2 Составить программу на языке PHP с использованием функций, указанных в таблице 2.2, согласно варианту.

Таблица 2.2 – Варианты заданий

Номер варианта	Номер функции	Номер варианта	Номер функции	Номер варианта	Номер функции
1	1, 13, 25, 37	5	5, 17, 29, 41	9	9, 21, 33, 45
2	2, 14, 26, 38	6	6, 18, 30, 42	10	10, 22, 34, 46
3	3, 15, 27, 39	7	7, 19, 31, 43	11	11, 23, 35, 47
4	4, 16, 28, 40	8	8, 20, 32,44	12	12, 24, 36, 48

Контрольные вопросы

1 Каковы особенности строк, записанных в одинарных кавычках?
2 В чем особенности строк, записанных в двойных кавычках?
3 Что понимается под heredoc-синтаксисом?
4 Как объединить элементы массива в строку?
5 Как удалить пробелы или другие символы из конца строки?
6 Как преобразовать строку в массив?
7 Какая функция переворачивает строку задом наперед?

8 Какие функции преобразуют символы строки в верхний регистр?
9 Как удалить пробелы из начала и конца строки?
10 Что такое heredoc-текст?

2.3 Лабораторная работа № 15. Изучение операторов цикла языка PHP

В языке PHP существует несколько конструкций, позволяющих выполнять повторяющиеся действия в зависимости от условия. Это циклы while, do ...while, foreach и for.

while – это простой цикл. Он имеет две формы записи:

while (выражение) { блок_выполнения }

либо

while (выражение): блок_выполнения endwhile;

Циклы do..while похожи на циклы while, но в них истинность выражения проверяется в конце цикла. Форма записи:

do {блок_выполнения} while (выражение);

Цикл for со счетчиком используется для выполнения тела цикла определенное число раз. Синтаксис цикла for:

for (инициализирующие_команды; условие; команды_после_итерации) {тело_цикла;}

Цикл for начинает свою работу с выполнения инициализирующих_команд, которые выполняются только один раз. Затем проверяется условие_цикла, и если оно истинно (true), то выполняется тело_цикла. После того как будет выполнен последний оператор тела, выполняются команды_после_итерации. Затем снова проверяется условие_цикла. Если оно истинно (true), выполняется тело_цикла и команды_после_итерации и т. д. Например:

```
<?php
for ($x=0; $x<10; $x++) echo $x;
?>                  // выводит: 0123456789
```

Если необходимо указать несколько команд, то их можно разделить запятыми, например:

```
<?php
for ($x=0, $y=0; $x<10; $x++, $y++) echo $x;
?>                  // Выводит 0123456789
```

Пример использования нескольких команд в цикле for:

```
<?php
for($i=0,$j=0,$k="Точки"; $i<10; $j++,$i+=$j) { $k=$k."."; echo $k; }
                    // Выводит Точки.Точки..Точки...Точки....
?>
```

Цикл for имеет альтернативный синтаксис:

```
for(инициализирующие_команды; условие; команды_после_итерации);
операторы;
endfor;
```

Цикл foreach перебора массивов имеет синтаксис:

```
foreach (массив as $ключ=>$значение)
команды;
```

Пример цикла foreach:

```
<?php
$names["Иванов"] = "Андрей";          $names["Петров"] = "Борис";
$names["Волков"] = "Сергей";          $names["Макаров"] = "Федор";
foreach ($names as $key => $value) {
echo "<b>$value $key</b><br>";
}
?>
```

выводит: Андрей Иванов
Борис Петров
Сергей Волков
Федор Макаров

Порядок выполнения работы.

Написать и отладить скрипт, выполняющий действия, указанные в таблице 2.3, согласно варианту.

Таблица 2.3 – Варианты заданий

Номер вари-анта	С помощью цикла
1	2
1	Найти сумму корней чисел от 1 до 15. Результат округлить до двух знаков в дробной части
2	Найти сумму тех чисел от 1 до 100, которые делятся на 7
3	Создать строку из шести символов, состоящую из случайных чисел от 1 до 9
4	Дан массив с числами, найти сумму квадратов его элементов
5	Дан массив с числами, найти корень из суммы квадратов элементов этого массива, а результат округлить в меньшую сторону до целых
6	Дан массив с числами, найти сумму тех его чисел, которые больше 0 и меньше 10
7	Заполнить двумерный массив, содержащий 10 подмассивов, случайными числами от 1 до 10. В каждом подмассиве должно быть по 10 элементов
8	Преобразовать строку 'var_text_hello' в 'varTextHello'. Скрипт должен работать с любыми строками такого типа
9	Дан массив с произвольными числами. Сделать так, чтобы элемент в массиве повторился количество раз, соответствующее его значению. Например, [1, 3, 2, 4] должен превратиться в [1, 3, 3, 3, 2, 2, 4, 4, 4, 4]

Окончание таблицы 2.3

1	2
10	Дана строка, удалить из этой строки четные символы
11	Дана строка, поменять ее первый символ на второй и наоборот, третий на четвертый и наоборот, пятый на шестой и наоборот и т. д., т. е. из строки '12345678' необходимо сформировать строку '21436587'
12	Написать скрипт, который проверяет, являются ли заданные числа простыми, т. е. делящимися только на единицу и сами на себя

Контрольные вопросы

1 Какие операторы цикла Вы знаете?
2 Какие формы записи конструкции while Вы знаете?
3 Прокомментируйте работу цикла do..while.
4 Для чего используется конструкция for?
5 Какие формы записи конструкции for Вы знаете?
6 Для чего используется оператор break в цикле for?
7 Прокомментируйте работу конструкции foreach.
8 Какие конструкции цикла используют для работы с массивами?
9 Как прервать выполнение цикла?
10 Для чего используется оператор break?

2.4 Лабораторная работа № 16. Изучение приемов работы с массивами на языке PHP

В языке PHP в одном массиве допускается хранение переменных различных типов, а также массивов и объектов. Для обращения к элементу массива используется его индекс (ключ).

PHP поддерживает работу с индексными и ассоциативными массивами, индексами которых являются строки.

Для обращения к элементам индексных массивов используются числовые индексы, а ассоциативных – строковые.

Для создания массивов можно использовать конструкцию array() или способ приведения скалярной переменной типа int, float, string или boolean к типу array, а также специализированные функции:

array([…]) – создает массив из значений, переданных конструкции в качестве параметров array_fill($start_index, $num, $value), которая возвращает массив, содержащий $num элементов, имеющих значение $value. Нумерация индексов при этом начинается со значения $start_index;

range($low, $high [, $step]) – создает массив со значениями из интервала от $iow до $high и шагом $step;

explode($delimiter, $str [, $limit) – возвращает массив из строк, каждая из которых соответствует фрагменту исходной строки $str, находящемуся между разделителем, определяемым аргументом $delimiter. Необязательный параметр

$limit определяет максимальное количество элементов в массиве, при этом последний элемент будет содержать остаток строки $str.

В качестве элементов массива могут выступать другие массивы, в этом случае говорят о многомерных массивах. Массивы можно создавать, обращаясь к элементам или используя вложенные конструкции array(). Для вывода массива используется функция print_r().

Работу с ассоциативными массивами удобно выполнять с использованием специализированного оператора цикла foreach.

При манипуляции с массивами и их элементами часто возникает необходимость определения количества элементов в массиве. Для решения этой задачи используются следующие функции:

count($array [,$mode]) – возвращает количество элементов массива $array. Если $mode принимает значение count_recursive, функция рекурсивно обходит многомерный массив, в противном случае подсчитывается количество элементов только на текущем уровне;

sizeof() – синоним для функции count();

array_count_values($input) – подсчитывает количество уникальных значений среди элементов массива и возвращает ассоциативный массив, ключами которого являются значения массива, а значениями – количество их вхождений в массив $input.

Порядок выполнения работы.

Написать и отладить скрипт, выполняющий действия, указанные в таблице 2.4, согласно варианту.

Таблица 2.4 – Варианты заданий

Номер варианта	Задание
1	2
1	В массиве из n строк проверить, начинается ли каждая строка символом "*", а строки без "*" перенести в другой массив
2	В массиве из n строк проверить, содержит ли k-я строка символ @. Если не содержит, то вставить этот символ в конец строки
3	В массиве строк удалить все html-теги, заключенные в скобки < >
4	В массив случайным образом поместить строки, содержащие «цитата дня». Для выбора строки из массива случайным образом можно использовать функции Shuffle(array arr) или arrayrand(array arr, int num)
5	Создать многомерный массив: Факультет, Курс, Группа, Студенты. Вывести список студентов в алфавитном порядке
6	Создать многомерный массив: Факультет, Кафедра, Преподаватель, Ученое_звание. Вывести список преподавателей в алфавитном порядке
7	Создать двухмерный массив, в первой строке которого записаны номера и названия месяцев года в произвольном порядке. Во второй строке рассортировать месяцы года в алфавитном порядке, а в третьей – в порядке возрастания номера месяца
8	Заполнить элементы квадратной матрицы возрастающими числами начиная с единицы по спирали, начиная с элемента [1, 1] по часовой стрелке

Окончание таблицы 2.4

1	2
9	Заполнить элементы квадратной матрицы возрастающими числами начиная с единицы по спирали, начиная с элемента [n, n] против часовой стрелки
10	Перемножить две числовые матрицы, размеры и значения матриц выбрать самостоятельно
11	Найти максимальную сумму диагональных элементов квадратной матрицы, размер и значения матрицы выбрать самостоятельно
12	Найти суммы элементов двух квадратных матриц и выбрать матрицу с большей суммой, размеры и значения матриц выбрать самостоятельно

Контрольные вопросы

1 Допускается ли хранение в одном массиве значений разных типов?
2 Что такое ассоциированный массив?
3 Какие конструкции используются для создания массивов?
4 Что такое индексный массив?
5 Как создать двухмерный массив?
6 Для чего используется функция Shuffle(array arr)?
7 Прокомментируйте назначение функции arrayrand(array arr, int num).
8 Можно ли хранить в массиве другие объекты?
9 Какие специализированные функции создания массивов Вы знаете?
10 Может ли быть элементом массива другой массив?

2.5 Лабораторная работа № 17. Изучение условных операторов PHP

К условным операторам языка PHP относятся операторы if с расширениями else, elseif, а также конструкция switch, позволяющая проверить условие и выполнить в зависимости от его истинности определенные действия.

Структура оператора if:

```
if (выражение) блок_выполнения
```

Здесь выражение – это любое правильное PHP-выражение, которое в процессе обработки скрипта преобразуется к логическому типу. Если в результате преобразования значение выражения истинно, то выполняется блок_выполнения, в противном случае блок_выполнения игнорируется. Если блок_выполнения содержит несколько команд, то он заключается в фигурные скобки.

Структура оператора switch:

```
switch (выражение или переменная){
case значение1:
   блок_действий1
break;
case значение2:
   блок_действий2
break;
```

```
...
default:
   блок_действий_по_умолчанию
}
```

Для конструкции switch, как и для if, возможен альтернативный синтаксис, где открывающая switch фигурная скобка заменяется двоеточием, а закрывающая – endif и endswitch соответственно.

Порядок выполнения работы.

Написать и отладить скрипт, выполняющий действия, указанные в таблице 2.5, согласно варианту.

Таблица 2.5 – Варианты заданий

Номер варианта	Задание
1	В массиве $a хранятся целые числа и слова. Определить количество и сумму положительных, отрицательных чисел, а также количество слов в массиве
2	В массиве $x хранятся целые и вещественные числа. Определить произведение всех целых и сумму вещественных чисел, больших 25,5
3	Переменная $s содержит строку, состоящую из положительных и отрицательных чисел и слов, разделенных запятыми. Сформировать массив чисел и определить среднее их абсолютных значений, а также строку слов
4	Переменная $min содержит строку разделенных пробелом чисел из интервала от 0 до 59. Определить, сколько чисел попадает в какую четверть часа, а также суммы чисел в каждой четверти
5	Переменная $lang может принимать два значения: "ru" и "en". Если она имеет значение "ru", то в переменную $arr записать массив дней недели на русском языке, а если "en" – то на английском. Задачу решить с использованием if, switch-case и многомерного массива без if и switch
6	В переменной $a содержатся целые числа, разделенные пробелами. Если выделяемые значения равны или меньше пяти, то в переменную $b записать их сумму, а если больше или равны девяти, то разность
7	Если i-я переменная массива $a больше двух и меньше девяти или переменная $b больше или равна шести и меньше девяти, то вывести "Верно", в противном случае "Неверно"
8	В массиве $day содержатся числа из интервала от единицы до 31. Определить, в какую декаду месяца попадают эти числа
9	Переменная $month содержит числа из интервала от единицы до 12. Определить, в какую пору года попадает этот месяц: зима, весна, лето или осень
10	В переменной $year хранится номер год. Определить, является ли он високосным. Год является високосным, если он делится на 4, но при этом не делится на 100 либо делится на 400
11	Дана строка с цифрами, например, "12345". Проверить, является ли первым символом этой строки цифра 1, 2 или 3. Если является, то вывести "да", в противном случае вывести "нет"
12	Дана строка из шести цифр. Проверить, равняется ли сумма первых трех цифр сумме вторых трех цифр. Если это так, вывести "да", в противном случае "нет"

Контрольные вопросы

1 Какие операторы в языке PHP относятся к условным?
2 Какие расширения используются в операторе if?
3 Для чего используется оператор switch?
4 Прокомментируйте структуру оператора switch.
5 Для чего используется расширение else в операторе if?
6 Для чего используется расширение elseif в операторе if?
7 Что понимается под альтернативным синтаксисом оператора if?
8 Для чего используется break в операторе switch?
9 Как оформить несколько команд в блоке выполнения оператора if?
10 Что понимается под альтернативным синтаксисом оператора switch?

2.6 Лабораторная работа № 18. Изучение технологии работы с функциями PHP

В PHP существуют две основные формы функций: встроенные и пользовательские.

Полный список встроенных PHP функций можно просмотреть в окне редактора кода, нажав кнопку «*Поиск*» в правой колонке при пустой строке поиска «PHP-поиск». Для просмотра подробного описания с примером конкретной PHP-функции необходимо указать ее имя в строке PHP-поиск.

Пользовательская функция создается с помощью оператора function, после которого через пробел указывается имя функции и круглые скобки, которые могут быть пустыми либо содержать параметры, переменные PHP, принимаемые функцией.

После объявления функции в фигурных скобках записывается код, выполняемый функцией. Общий синтаксис пользовательской функции имеет вид:

```
function имя_функции (параметры)
{
  //тело функции
}
```

Вызывается функция по ее имени, после которого обязательны круглые скобки, даже если они пустые.

Для возврата значения, являющегося результатом работы функции, используется оператор return, который прекращает выполнение текущей функции и возвращает ее значение. При этом оператор return может быть расположен в любом месте функции.

В PHP допускается использование **динамических функций**. Это означает, что если некоторой переменной присвоено имя функции, то с этой переменной можно обращаться точно так же, как с самой функцией.

В функциях допускается использование глобальных переменных, созданных с помощью инструкции **global** вне функции.

Чтобы переменная сохраняла свое значение между вызовами функции, нужно объявить ее статической с помощью инструкции **static**.

Порядок выполнения работы.

Написать и отладить скрипт, выполняющий действия, указанные в таблице 2.6, согласно варианту.

Таблица 2.6 – Варианты заданий

Номер вари-анта	Задания
1	2
1	1 Создать пользовательскую функцию, которая принимает два аргумента и возвращает их произведение. Вызвать функцию, передав ей в качестве аргументов два числа, и результат вывести на экран. 2 Создать пользовательскую функцию, формирующую массив делителей (чисел, на которое оно делится без остатка) заданного числа
2	Создать три переменные, присвоить им числовые значения и вывести на экран их произведение. Создать пользовательскую функцию, принимающую два аргумента по ссылке и один по значению, которая должна присваивать переменным другие числовые значения. Вызвать функцию и вывести на экран произведение всех переменных
3	Создать две переменные, присвоить им числовые значения и создать пользовательскую функцию, принимающую два аргумента со значениями по умолчанию и выводящую произведение своих аргументов. Вызвать функцию, передав ей в качестве аргументов сначала значения двух переменных, затем значение одной из переменных и, наконец, вообще без аргументов
4	Создать пользовательскую функцию, принимающую аргументы в массив переменной длины и выводящую их на экран. Для доступа к элементам массива использовать цикл foreach. Вызвать функцию, передав ей в качестве значения две строки и число
5	Создать пользовательскую функцию, которая вычисляет корень из заданного четырехзначного числа и округляет его в большую и меньшую стороны. В массив $arr первым элементом записать заданное число, вторым – корень из этого числа, третьим – округление в меньшую сторону, четвертым – округление в большую сторону
6	Заполнить массив 30 случайными числами от 1 до 10. Найти квадратный корень из каждого числа. Округлить результаты в большую и меньшую сторону и записать результаты округления в ассоциативный массив с ключами 'floor' и 'ceil'
7	Дано целое двухзначное число. Создать функцию, определяющую делители этого числа, т. е. числа, на которое оно делится без остатка. Сформировать массив делителей заданного числа
8	Задать режим строгой типизации, использовать инструкцию declare(strict_types=1), после этого создать пользовательскую функцию, которая будет принимать два целочисленных аргумента и выводить на экран их сумму. Вызвать функцию, передав ей в качестве аргументов сначала два целых числа, а затем одно из них в виде строки

Окончание таблицы 2.6

1	2
9	Задать режим строгой типизации, используя инструкцию declare(strict_types=1), затем создать пользовательскую функцию my_func(), которая будет принимать два целочисленных аргумента и возвращать их произведение. Создать переменную $count_apples и присвоить ей строку с именем функции. Обратиться к функции через переменную и вывести на экран общую массу яблок, зная, что имеется 25 корзин по семь килограмм яблок в каждой
10	Создать переменную и присвоить ей целое число. Создать еще одну переменную и присвоить ей анонимную функцию, наследующую эту переменную и выводящую на экран ее инкрементированное значение. Выполнить вызов функции, затем изменить значение внешней переменной и снова вызвать функцию. Изменить скрипт, задав наследование переменной по ссылке
11	1 Создать функцию, вычисляющую квадратный корень из заданного числа. Результат округлить в большую и меньшую сторону, результаты округления записать в ассоциативный массив с ключами "floor" и "ceil". 2 Создать функцию, определяющую, сколько первых элементов массива [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] нужно сложить, чтобы сумма получилась больше 10
12	Создать пользовательскую функцию, вычисляющую корень квадратный из заданного четырехзначного числа. Результат округлить в большую и меньшую стороны. В массив $arr записать первым элементом корень из числа, вторым – округление в меньшую сторону, третьим – округление в большую сторону. Для решения задачи можно использовать функции: sqrt – корень из числа, floor – округление в меньшую сторону и ceil– округление в большую сторону

Контрольные вопросы

1 Какие две основные формы PHP-функций Вы знаете?
2 Как посмотреть полный список встроенных PHP-функций?
3 Как просмотреть подробное описание конкретной PHP-функции?
4 Как создаются пользовательские PHP-функции?
5 Что записывается в фигурных скобках в пользовательской PHP-функции?
6 Прокомментируйте общий синтаксис PHP-функции.
7 Как вызвать PHP-функцию?
8 Как вернуть результат пользовательской PHP-функции?
9 Что такое динамические PHP-функции?
10 Как использовать глобальные переменные в PHP-функциях?

2.7 Лабораторная работа № 19. Изучение технологии работы с файлами PHP

В PHP-документ с помощью инструкции include() можно включать файлы кода. Аргументом этой инструкции является путь к файлу. Чтобы содержимое этого файла обрабатывалось как PHP-программа, его необходимо обрамлять открывающим и закрывающим тегами PHP.

PHP содержит множество функций управления файлами, наиболее

употребительными из которых являются:

touch() – создает пустой файл с заданным именем, например: touch("ex1.txt"). Если такой файл уже существует, то функция изменит дату модификации;

copy() – копирует файл. Для копирования файлов в PHP используется функция copy($source, $result), в которой используются два параметра – источник $source и имя файла-копии $result. При этом следует указывать полные адреса к файлам;

unlink() – удаляет заданный файл. Например:

```
<?php
if (unlink('filename.txt'))
   { echo "Файл удален"; }
else
   { echo "Ошибка при удалении файла"; }
?>
```

fopen() – открывает локальный или удаленный файл и возвращает указатель на него. Указатель используется во всех операциях с содержимым файла. Аргументами функции fopen() являются имя файла и режим открытия;

fclose() – закрывает файл. Аргумент: указатель файла, полученный ранее от функции fopen();

feof() – проверяет конец файла. Аргумент: указатель файла;

fgetc() – читает очередной символ из файла. Аргумент: указатель файла;

fgets() – читает очередную строку файла. Аргументы: указатель файла и длина считываемой строки. Операция прекращается после считывания заданного количества символов или при обнаружении конца строки или файла;

fread() – общая функция чтения из файла. Аргументы: указатель файла и количество считываемых символов;

fseek() – отступ от начала файла. Аргументы: указатель файла и смещение;

fputs() – записывает строку в файл. Аргументы: указатель файла и строка;

fwrite() – полный аналог функции fputs( );

flock() – блокирует файл, т. е. не позволяет другим пользователям читать этот файл или писать в него, пока тот, кто наложил блокировку, не закончит работу с данным файлом. Аргументы: указатель файла и номер режима блокировки.

Порядок выполнения работы.

Написать и отладить скрипт, выполняющий действия, указанные в таблице 2.7 согласно варианту.

Таблица 2.7 – Варианты заданий

Номер варианта	Задание
1	Дан массив со строками. Создать папку 'test', а в ней – папки, названиями которых служат элементы этого массива
2	Найти все файлы из папки 'test' и вставить в начало каждого файла полный путь к нему. Текст файла должен остаться в нем и начинаться с новой строки после пути

Окончание таблицы 2.7

Номер варианта	Задание
3	Вывести на экран имена всех папок из папки 'test' и их подпапок. Уровень вложенности папок может быть любым
4	Вывести на экран содержимое всех файлов из папки 'test' и ее подпапок. Уровень вложенности папок может быть любым
5	Найти все файлы из папки 'test' и ее подпапок любого уровня вложенности и вставить в начало каждого файла полный путь к нему (текст файла должен остаться в нем и начинаться с новой строки после пути)
6	Удалить из папки 'test' все файлы размером более 1 Мбайт
7	Имеется папка с файлами, определить и вывести на экран размер этой папки
8	Имеется папка с подпапками, определить размеры всех подпапок папки и вывести их на экран
9	Вывести на экран название всех файлов с расширением txt из папки 'test'
10	В папке 'test' есть файлы и подпапки. Вывести на экран содержимое всех файлов, которые лежат непосредственно в папке 'test'
11	Вывести на экран название всех файлов, но не подпапок из папки 'test'
12	Вывести на экран название всех файлов и подпапок из папки 'test'

Контрольные вопросы

1 С помощью какой инструкции можно включать файлы кода в PHP-документ?
2 С помощью какой PHP-функции можно создать пустой файл?
3 Какая PHP-функция используется для открытия файла?
4 Какие параметры используются в PHP-функции copy()?
5 Какая PHP-функция используется для чтения очередной строки файла?
6 Какая PHP-функция используется для записи строки в файл?
7 Для чего используется PHP-функция fputs()?
8 Как в качестве элемента массива создать другой массив?
9 Как вывести на экран имена всех папок из папки?
10 Как вывести на экран название всех файлов из папки?

Список литературы

1 **Никольский, А. П.** JavaScript на примерах / А. П. Никольский. – Москва: Наука и техника, 2017. – 274 с.

2 **Симпсон, К.** ES6 & Beyond / ES6 и не только / К. Симпсон. – Санкт-Петербург: Питер, 2017. – 336 с.

3 **Кузнецов, М.** Самоучитель PHP 7 / М. Кузнецов, И. Симдянов. – Санкт-Петербург: БХВ-Петербург, 2018. – 450 с.

4 **Скляр, Д.** Изучаем PHP 7. Руководство по созданию интерактивных веб-сайтов: пер. с англ. / Д. Скляр. – Санкт-Петербург: Альфа-книга, 2017. – 464 с.: ил.