

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)**

**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ
ИМ. Б.Л. РОЗИНГА (ФИЛИАЛ) СПбГУТ
(АКТ (Ф) СПбГУТ)**

**Составил
Ю.С. Маломан**

УП.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

Методические указания по выполнению практических работ

по специальностям:

09.02.07 – Информационные системы и программирование

Архангельск 2021

Маломан, Ю.С. УП.01 Разработка программных модулей. Методические указания по выполнению практических работ. – Архангельск: АКТ (ф) СПбГУТ, 2021

Методические указания по выполнению практических работ содержат описания работ, предусмотренных рабочей программой по УП.01 «Разработка программных модулей». Каждая работа рассчитана на 2 часа, общий объем составляет 72 часа. Методические указания по выполнению практических работ предназначены для студентов очной формы обучения по специальности 09.02.07 – Информационные системы и программирование.

Рассмотрено и одобрено на заседании цикловой комиссии Информационных технологий и математических дисциплин АКТ (ф) СПбГУТ.

Содержание

Практическая работа №1 Верстка веб-страниц	5
Практическая работа №2 Вывод и форматирование текста на веб-страницах	8
Практическая работа №3 Отображение медиа-контента на веб-страницах	10
Практическая работа №4 Реализация переходов между веб-страницами	12
Практическая работа №5 Настройка стилей веб-страниц	14
Практическая работа №6 Отображение списков на веб-страницах	16
Практическая работа №7 Отображение таблиц на веб-страницах	18
Практическая работа №8 Разработка и подключение веб-сценариев на JavaScript	20
Практическая работа №9 Вывод данных в JavaScript-сценарии	22
Практическая работа №10 Отладка с помощью средств разработчика в браузере	24
Практическая работа №11 Отладка JavaScript-сценариев средствами класса console	27
Практическая работа №12 Разработка JavaScript-сценариев, содержащих операторы ветвления	29
Практическая работа №13 Разработка JavaScript-сценариев, содержащих циклические операторы	31
Практическая работа №14 Разработка функций на JavaScript	33
Практическая работа №15 Разработка программ с использованием массивов на JavaScript	35
Практическая работа №16 Разработка программ обработки строковых данных на JavaScript	37
Практическая работа №17 Разработка объектов на JavaScript	39
Практическая работа №18 Разработка веб-приложений с помощью библиотеки jQuery	41
Практическая работа №19 Настройка стилей веб-приложений с помощью библиотеки jQuery	45
Практическая работа №20 Обработка событий таймера на JavaScript	47
Практическая работа №21 Обработка событий мыши на JavaScript	49
Практическая работа №22 Обработка событий клавиатуры на JavaScript	51
Практическая работа №23 Разработка интерфейса интерактивного игрового веб-приложения	53
Практическая работа №24 Разработка интерактивного игрового веб-приложения	55
Практическая работа №25 Установка и настройка локального сервера	57
Практическая работа №26 Встраивание RHP-сценариев в веб-страницы	59
Практическая работа №27 Включение внешних модулей в RHP-сценарии	62
Практическая работа №28 Вывод данных в RHP-сценариях	65
Практическая работа №29 Разработка RHP-сценариев, содержащих операторы ветвления	67
Практическая работа №30 Разработка RHP-сценариев, содержащих циклические операторы	69
Практическая работа №31 Разработка функций на RHP	71
Практическая работа №32 Разработка программ с использованием массивов на RHP	73

Практическая работа №33 Разработка программ обработки строковых данных на РНР	75
Практическая работа №34 Разработка объектов на РНР	77
Практическая работа №35 Разработка веб-форм	79
Практическая работа №36 Передача данных между веб-страницами	81

Практическая работа №1

Верстка веб-страниц

1 Цель работы

1.1 Научиться применять семантические тэги html5, применяющиеся для формирования веб-страниц.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.2-3.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Настройка тэга head (заголовок и кодировки)

Создать html-документ для отображения новостей:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <title>Заголовок</title>
```

```
  </head>
```

```
  <body>
```

```
    <p>Содержание документа HTML5</p>
```

```
  </body>
```

```
</html>
```

Поменять заголовок на «Новости» и проверить результат в браузере.

5.2 Создание шапки сайта

Добавить на страницу после открывающегося тэга body тэг header и внутри него разместить текст, который будет отображаться в начале странице как заголовок верхнего уровня:

```
<h1>Последние новости</h1>
```

5.3 Создание подвала сайта

Добавить на страницу перед закрывающимся тэгом body тэг footer и внутри него разместить текст, который будет отображаться в конце страницы знак копирайта, имя автора и текущий год:

```
<p>
```

```
  Фамилия Имя<br>
```

</p>

5.4 Создание навигационного меню:

Добавить в шапку сайта после закрытия тэга </h1> тэг nav в котором в виде списка должны отображаться пункты меню:

```
<nav>
  <ul>
    <li>Главная
    <li>Вперед
    <li>Назад
  </ul>
</nav>
```

5.5 Добавление содержимого из трех статей

Добавить между закрывшимся тэгом шапки сайта и открывшимся тэгом подвала сайта тэг main (основное содержимое страницы) со следующим текстом:

```
<main>
  <h2>Статьи</h2>

  <article>
    <h3>Новость 1</h3>
    <p>текст статьи о новости 1.</p>
  </article>

  <article>
    <h3>Новость 2</h3>
    <p>текст статьи о новости 2.</p>
  </article>
</main>
```

6 Порядок выполнения работы

- 6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Какова структура html-страницы?
- 8.2 Что указывается в тэге head?
- 8.3 Что указывается в тэге body?
- 8.4 Для чего используется тэг header?

8.5 Для чего используется тэг footer?

8.6 Для чего используется тэг nav?

8.7 Для чего используется тэг main?

Практическая работа №2

Вывод и форматирование текста на веб-страницах

1 Цель работы

- 1.1 Научиться использовать html-тэги для вывода текста на страницу;
- 1.2 Научиться выполнять форматирование текста на странице, используя html-тэги.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.3-4, гл.11.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Используя тэги заголовков от <h1> до <h6> вывести на странице 6 строк и сравнить полученный результат.

Для разделения использовать одиночные тэги <hr> (горизонтальная линия) и
 (перевод на новую строку).

5.2 Добавить после каждого заголовка по 2-3 абзаца, состоящих из нескольких строк текста. Для задания абзаца использовать тэг <p>

5.3 Используя тэги <sub> и <sup> вывести на экран формулу воды (H₂O) и единицу измерения площади (м²).

5.4 Используя тэг и вывести на странице текст, написанный курсивом, полужирным и полужирным курсивом.

5.5 Используя один тэг <pre>...</pre>, вывести на экран многострочный текст, содержащий большое количество подряд идущих пробелов, пробелы в начале строки.

6 Порядок выполнения работы

- 6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Для чего используется тэг p?

- 8.2 Для чего используется тэг `hr`?
- 8.3 Для чего используется тэг `br`?
- 8.4 Для чего используются тэги `h1...h6`?
- 8.5 Для чего используются тэги `em` и `strong`?

Практическая работа №3

Отображение медиа-контента на веб-страницах

1 Цель работы

1.1 Научиться использовать тэги html5, применяющиеся для формирования веб-страниц, содержащих мультимедиа-контент.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.5.

2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.23.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Реализовать отображение на html-странице изображения с замещающим текстом «Пример изображения», используя тэг `img`.

5.2 Реализовать вставку аудиофайла на html-страницу, используя тэг `audio`. При загрузке страницы должно включаться автовоспроизведение по кругу аудиофайла. Также у аудиофайла должны отображаться управляющие воспроизведением кнопки.

5.3 Реализовать вставку видеофайла на html-страницу, используя тэг `video`. У видеофайла должны отображаться управляющие воспроизведением кнопки.

5.4 Добавить тэгу `video` атрибуты для задания ширины и высоты видеоролика и изображения, отображаемого пока видеоролик не запущен.

5.5 Реализовать вставку видеофайла на html-страницу, используя тэг `video`. Аудиодорожка и видеодорожка должны подключаться отдельными файлами в тэгах `source`, вложенных в тэг `video`.

6 Порядок выполнения работы

6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

- 8.1 Для чего используется тэг `img`?
- 8.2 Для чего используется тэг `audio`?
- 8.3 Для чего используется тэг `video`?
- 8.4 Какой атрибут позволяет отобразить управляющие аудио и видео на странице?
- 8.5 Какой атрибут используется для указания ссылки на мультимедиафайл?

Практическая работа №4

Реализация переходов между веб-страницами

1 Цель работы

1.1 Научиться использовать html-тэги, применяющиеся для перехода и навигации.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.7.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Создать 3 html-страницы. У каждой разместить в шапке по 2 текстовые гиперссылки, чтобы с каждой страницы можно было переключиться на любую другую.

5.2 Добавить на одну из страниц почтовую гиперссылку с подписью «Связаться с владельцем сайта».

5.3 Добавить на одну из страниц 3 больших блока текста, у каждого указать свой заголовок верхнего уровня. У каждого заголовка задать атрибут id. В начале страницы разместить меню для перехода к каждому из блоков текста с использованием якорей.

5.4 Добавить на одну из страниц вместо текстовый гиперссылки гиперссылку в виде изображения.

5.5 Добавить в конце каждого блока текста из п.5.3 гиперссылку «Наверх» и реализовать переход в начало страницы, используя якоря.

6 Порядок выполнения работы

6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Какой тэг применяется для создания гиперссылки?

8.2 Как создать текстовую гиперссылку?

- 8.3 Как создать гиперссылку в виде изображения?
- 8.4 Как создать почтовую гиперссылку?
- 8.5 Как перейти к «якорю»?

Практическая работа №5

Настройка стилей веб-страниц

1 Цель работы

1.1 Научиться оформлять веб-страницы, используя CSS.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.9-11, гл.13, гл.16.

2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.18.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

Стили описывать во внешнем файле и подключать к html-страницы из ПР№1.

5.1 Используя css, сменить фон страницы на линейный градиент и изменить цвет заголовков.

5.2 Используя css, изменить шрифт у абзацев.

5.3 Используя css, добавить одному из заголовков тень.

5.4 Используя css, сделать один из абзацев невидимым.

5.5 Используя css, настроить блочный дизайн элементов страницы (шапка — по верхнему краю, подвал — по нижнему краю, панель навигации в оставшейся части страницы слева, все остальное — контент, основная часть).

6 Порядок выполнения работы

6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Как подключить стиль к html-странице?

8.2 Что такое «блочный дизайн»?

8.3 Как указать ширину и высоту элементов, используя css?

8.4 Как указать цвет фона элементов, используя css?

8.5 Как указать настройки текста, используя css?

Практическая работа №6

Отображение списков на веб-страницах

1 Цель работы

1.1 Научиться использовать тэги html5, применяющиеся для размещения списков на веб-страницах.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – п.3.1, гл.12.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Используя тэги ul и li, создать маркированный список из пяти пунктов.

5.2 Используя тэги ol и li, создать нумерованный список из пяти пунктов.

5.3 Используя тэги ul и li, описать расписание занятий на дни с понедельника по субботу. Для каждого дня занятия представляют вложенный список.

5.4 Используя css, изменить нумерацию элементов списка из п.5.2 с арабских на римские числа, написанные заглавными буквами.

5.5 Используя css, изменить маркеры элементов списка из п.5.1 на квадраты.

6 Порядок выполнения работы

6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Для чего используется тэг ul?

8.2 Для чего используется тэг ol?

8.3 Для чего используется тэг li?

8.4 Как изменить стиль элементов списка, используя css?

Практическая работа №7

Отображение таблиц на веб-страницах

1 Цель работы

1.1 Научиться использовать тэги html5, применяющиеся для размещения таблиц на веб-страницах.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.6, гл.15.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Используя тэги table, thead, tr, th, создать первую строку таблицы для отображения расписания занятий: первая ячейка пустая, остальные ячейки заполнены днями недели с понедельника по субботу.

5.2 Используя тэги tbody, tr, td заполнить расписание занятий в таблице из п.5.1. В первом столбце — номер пары по расписанию, в ячейках на пересечении дня и пары — название предмета.

5.3 Добавить в таблицу caption (заголовок) «Расписание занятий»

5.4 Выполнить объединение находящихся в одном столбце находящихся рядом пустых ячеек.

5.5 Используя css, изменить выравнивание таблицы, чтобы она находилась по центру, границы сделать толщиной 3 пикселя, таблица должна занимать 80% старинцы, высота первой строки — 30 пикселей.

6 Порядок выполнения работы

6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Для чего используется тэг table?

8.2 Для чего используется тэг thead?

- 8.3 Для чего используется тэг `tbody`?
- 8.4 Для чего используется тэг `tr`?
- 8.5 Для чего используется тэг `th`?
- 8.6 Для чего используется тэг `td`?
- 8.7 Для чего используется тэг `caption`?

Практическая работа №8

Разработка и подключение веб-сценариев на JavaScript

1 Цель работы

1.1 Научиться создавать и подключать сценарии на JavaScript к веб-страницам.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.18.

2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.13.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Подключение js-сценария в тэге body

Создать html-страницу со следующим кодом (в тэге script пишется код js-сценария):

```
<html>
<head>
<title>Hello World</title>
</head>
<body>
<script type="text/javascript">
  document.write("Hello World")
</script>
<noscript>
  Ваш браузер не поддерживает JS, или его поддержка отключена
</noscript>
</body>
</html>
```

Проверить результат в браузере.

5.2 Подключение js-сценария в тэге head

Создать копию html-страницы из п.5.1 и перенести тэг script со всем содержимым внутрь тэга head.

Проверить результат в браузере.

5.3 Включение внешнего js-сценария в тэге head

Создать js-файл с названием script.js со следующим текстом:

```
document.write("Hello World")
```

Создать копию html-страницы из п.5.2, удалить содержимое между тэгами script и добавить в тэг script атрибут со ссылкой на js-файл:

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

Проверить результат в браузере.

5.4 Вывод на экран переменных

Добавить в js-файл с названием script.js объявление переменных:

```
var a = 10;
```

```
var b = 20;
```

Используя функцию document.write(), вывести на странице сумму переменных a и b.

Проверить результат в браузере.

6 Порядок выполнения работы

6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Где может быть указан тэг script?

8.2 Как подключить внешний js-скрипт?

8.3 Как описать js-скрипт внутри html-страницы?

Практическая работа №9

Вывод данных в JavaScript-сценарии

1 Цель работы

1.1 Научиться выводить данные на html-странице, используя JavaScript.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.22.

2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.13.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Вывод данных на странице с использованием write

Создать и подключить к html-странице js-скрипт, выводящий на экран переменные a и b и сумму этих переменных, используя функцию document.write(). Каждое значение вывести на отдельной строке. Для перевода строки использовать
.

5.2 Вывод данных с использованием alert

Создать и подключить к html-странице js-скрипт, выводящий на экран «hello world», используя команду alert:

```
window.alert(текст);
```

5.3 Ввод данных с использованием prompt

Создать и подключить к html-странице js-скрипт, выводящий на экран сообщение «Введите ваше имя» и выводящий на странице «Добро пожаловать, имя» (вместо имя — введенный пользователем текст), используя команду prompt:

```
var переменная = window.prompt(текст);
```

5.4 Ввод данных с использованием confirm

Создать и подключить к html-странице js-скрипт, выводящий на экран сообщение «Вывести максимум?» и выводящий на странице максимум из двух числовых переменных, если пользователь нажал «ОК», иначе — минимум. Для выполнения проверять, равна ли переменная result значению true:

```
var result = confirm(текст);
```

6 Порядок выполнения работы

- 6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Что выполняет функция write()?
- 8.2 Что выполняет функция alert()?
- 8.3 Что выполняет функция prompt()?
- 8.4 Что выполняет функция confirm()?

Практическая работа №10

Отладка с помощью средств разработчика в браузере

1 Цель работы

- 1.1 Научиться проводить отладку JavaScript-сценариев;
- 1.2 Научиться применять средства разработчика в браузере;
- 1.3 Закрепить навык разработки JavaScript-сценариев.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.22.

2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.13.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Изучение содержимого инструментов разработчика

5.1.1 Создать файл factorial.js, в котором прописать функцию вычисления факториала (для вычисления использовать цикл со счетчиком, для отрицательных чисел возвращать -1):

```
function factorial(n) {  
  let answer = 1;  
  if (n == 0 || n == 1) {  
    return answer;  
  }  
  else {  
    for(var i = n; i >= 1; i--) {  
      answer = answer * i;  
    }  
    return answer;  
  }  
}
```

```
var x = factorial(5);  
alert(x);  
document.write(x);
```

Добавить в функцию многострочный комментарий с указанием того, что делает функция, имени разработчика, даты создания функции перед объявлением функции

и добавить однострочные комментарии, поясняющие каждую строчку кода функции вычисления факториала.

5.1.2 Создать файл index.html, подключить к нему созданный файл factorial.js:

Создать html-файл

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="UTF-8">
</head>
<body>
  <script src="factorial.js"></script>
</body>
</html>
```

1.3 Открыть файл index.html в браузере Google Chrome и перейти к инструментам разработчика (F12 или меню Инструменты – Инструменты Разработчика). Ознакомиться с содержанием вкладки Sources.

5.2 Просмотр ошибок в консоли браузера

Внести в javascript-сценарий ошибки. Проверить, что сообщения об ошибках выводятся в консоль браузера. Исправить ошибки.

5.3 Управление точками останова

Добавить точки останова в тело цикла функции factorial и добавить в отслеживаемые (Watch) переменную-счетчик цикла.

Используя кнопки управления выполнением, провести пошаговую отладку кода. Изучить, как меняется содержимое зоны информации и контроля (разделы Watch, Scope, Breakpoints) в процессе трассировки кода.

Изменить условие для точки останова в цикле так, чтобы остановка происходила при четных значениях i. Переход к настройкам точки останова осуществляется через ее контекстное меню.

5.4 Форматирование скриптов

Создать в файле factorial.js копию функции factorial, переименовать ее в unformattedFactorial и удалить из нее все переносы на новую строку и вспомогательные пробелы. Провести форматирование кода, используя кнопку Формат (пиктограмма {} во вкладке Sources).

5.5 Отображение сайта на устройствах с различным разрешением экрана

Создать файл со стилями, в котором прописать стили для двух блоков div (первый должен быть с шириной и высотой 760x1000, второй — 240x760, цвета блоков должны быть разными) и подключить их к файлу index.html, в теле которого создать два блока, у каждого из которых должен быть задан свой стиль из описанных в файле со стилями.

Перейти к инструментам разработчика и ознакомиться с возможностями по просмотру внешнего вида страниц на различных экранах (Toggle device mode):

- выбор устройства из стандартных;

- задание произвольного (отзывчивого) разрешения экрана;
- масштабирование;
- отображение линеек,
- запуск мобильной / десктопной версии.

6 Порядок выполнения работы

- 6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Как запустить отладку в браузере?
- 8.2 Как поставить точки останова в браузере?
- 8.3 Как просмотреть содержимое консоли в браузере?
- 8.4 Как просмотреть внешний вид сайта на устройствах с различным разрешением, используя средства разработчика в браузере?

Практическая работа №11

Отладка JavaScript-сценариев средствами класса console

1 Цель работы

- 1.1 Научиться проводить отладку JavaScript-сценариев;
- 1.2 Научиться применять отладочные функции JavaScript для работы с консолью;
- 1.3 Закрепить навык разработки JavaScript-сценариев.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.22.

2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.13.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

Работу выполнять на основе файлов из ПРН№10

5.1 Применение debugger

Добавить в javascript-сценарий следующую строку, являющуюся альтернативой точке останова:

`debugger;`

Сделать существующие точки останова неактивными. Провести трассировку кода.

5.2 Вывод данных в консоль с использованием console.log

Добавить в javascript-сценарий следующую команду для вывода значения `i` в консоль:

`console.log("i=", i);`

5.3 Вывод данных в консоль с использованием console.info

Реализовать вывод информации в следующем виде (вместо названий переменных должны отображаться их значения):

`i! = result`

Сравнить вывод `console.info` с выводом `console.log`

5.4 Вывод предупреждений и сообщений об ошибке в консоль

Добавить в тело javascript-сценария вычисление результата деления двух чисел.

Если делитель равен нулю, выводить сообщение об ошибке с помощью функции `console.error` и предупреждение о невозможности вывода результата с помощью функции `console.warn`.

6 Порядок выполнения работы

- 6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Для чего используется команда `debugger`?
- 8.2 Какие функции в JavaScript-сценариях позволяют вывести текст в консоль и для чего используется каждая функция?
- 8.3 Для чего используется функция `console.assert()` и какие параметры она принимает?

Практическая работа №12

Разработка JavaScript-сценариев, содержащих операторы ветвления

1 Цель работы

- 1.1 Научиться разрабатывать на JavaScript программы, содержащие ветвления;
- 1.2 Закрепить навык составления программ методами структурного программирования.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.18.

2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.14.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Написать js-сценарий, определяющий, какое из трех чисел минимально и выводящее его.

5.2 Написать js-сценарий, определяющую по номеру месяца сезон (зима, весна, лето, осень). Если номер месяца некорректный, вывести сообщение, что такого месяца нет.

5.3 Написать js-сценарий, определяющую по году и номеру месяца (считать, что год и номер месяца корректны):

- является ли введенный пользователем год високосным (т.е. кратен 4 и не кратен 100 или кратен 400);
- количество дней в месяце.

6 Порядок выполнения работы

- 6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Для чего используется условный оператор?

8.2 Какие существуют формы оператора if и каков их вид?

8.3 Для чего используется тернарный оператор?

8.4 Для чего используется оператор множественного выбора?

Практическая работа №13

Разработка JavaScript-сценариев, содержащих циклические операторы

1 Цель работы

- 1.1 Научиться разрабатывать на JavaScript программы, содержащие циклы;
- 1.2 Закрепить навык составления программ методами структурного программирования.

2 Литература

- 2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.18.
- 2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.14.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Разработать js-сценарий, определяющий и выводящий на экран количество цифр и сумму всех цифр в натуральном числе n. Количество цифр можно подсчитать путем многократного деления числа на 10.
- 5.2 Разработать js-сценарий, выводящий на экран таблицу умножения.
- 5.3 Разработать js-сценарий, вычисляющий и выводящий на экран результат возведения a в степень x (a — вещественное число, x — целое).

6 Порядок выполнения работы

- 6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Какие виды операторов цикла имеются в JavaScript и каков их синтаксис?
- 8.2 Какие операторы прерывания имеются в JavaScript и каков синтаксис их вызова?

Практическая работа №14

Разработка функций на JavaScript

1 Цель работы

1.1 Научиться разрабатывать на JavaScript программы, содержащие пользовательские функции (подпрограммы);

1.2 Закрепить навык составления программ методами процедурного и структурного программирования.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.18.

2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – п.15.1.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Написать и протестировать рекурсивную функцию вычисления факториала.

5.2 Написать функцию, возвращающую факториал указанного числа, вычисленный при помощи цикла со счетчиком.

5.3 Написать функцию, переводящую переданный параметр (секунды) в минуты:секунды и выводящую результат. Данные выводятся на экран в виде м:сс, например: «128 – 2:08».

5.4 Написать функцию, выводящую значение текущей даты в виде «дд.мм.гггг, день недели», например: 11.03.2021, вторник.

6 Порядок выполнения работы

6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

- 8.1 Как в JavaScript объявить и вызвать функцию, возвращающую значение?
- 8.2 Как в JavaScript объявить и вызвать функцию, не возвращающую значение?
- 8.3 В какой части программы могут быть объявлены подпрограммы?
- 8.4 Как подключить к html-странице файл с JavaScript-подпрограммами?

Практическая работа №15

Разработка программ с использованием массивов на JavaScript

1 Цель работы

- 1.1 Научиться создавать и обрабатывать массивы в программах на языке JavaScript;
- 1.2 Научиться применять стандартные методы класса Array для обработки массивов в программах на языке JavaScript;
- 1.3 Закрепить навык разработки и тестирования приложений на языке JavaScript.

2 Литература

- 2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.18.
- 2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – п.15.3.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Написать программу, в которой создать два массива:
 - а) первый заполнить N случайными целыми числами от 0 до 100;
 - б) второй заполнить нечетными числами из первого.Массивы вывести, используя названиеМассива.join(", ").

5.2 Написать программу, в которой вставить все элементы из одного массива в конец другого массива, используя стандартную функцию push().

Полученный массив выводить поэлементно с указанием индексов элементов (каждый элемент с новой строки).

5.3 Написать программу, в которой удалить N элементов массива, используя стандартную функцию pop().

Результаты вывести, используя названиеМассива.join(", ").

5.4 Написать программу, в которой отсортировать массив чисел, используя стандартную функцию sort():

- а) по возрастанию;
- б) по убыванию.

Результаты вывести, используя названиеМассива.join(", ").

6 Порядок выполнения работы

- 6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Что такое «массив»?
- 8.2 Каким образом определяются переменные типа массив (одномерный и двумерный) в JavaScript?
- 8.3 Как осуществляется доступ к отдельному элементу массива в JavaScript?
- 8.4 Какой класс является родительским для массивов в JavaScript?

Практическая работа №16

Разработка программ обработки строковых данных на JavaScript

1 Цель работы

- 1.1 Научиться создавать и обрабатывать строковые данные в программах на языке JavaScript;
- 1.2 Научиться применять стандартные методы класса String для обработки строк в программах на языке JavaScript;
- 1.3 Закрепить навык разработки и тестирования приложений на языке JavaScript.

2 Литература

- 2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.18.
- 2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.15.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Написать программу, в которой вывести символы строки — каждый на отдельной строке — с указанием кода символа.

Код символа можно получить, используя следующую функцию строка.charCodeAt(индекс символа)

- 5.2 Написать программу, в которой подсчитать количество вхождений подстроки в строку.

Подстроку можно найти, используя следующую функцию: строка.indexOf("строка", начальный индекс)

- 5.3 Написать программу, в которой подсчитать количество вхождений подстроки в строку без учета регистра.

Регистр можно изменить, используя следующую функцию: строка.toLowerCase()

- 5.4 Написать функцию для калькулятора. На входе — два параметра и операция (например, "+"), на выходе — строка (например: "1.2 * 6 = 7.2", "3 x 8 = 24", "Операция abc неизвестна"). Учесть, что некоторые операции могут обозначаться разными способами.

6 Порядок выполнения работы

- 6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 К какому типу переменных относятся переменные типа `string`?
- 8.2 Какие операции допустимы над строковыми данными?
- 8.3 В чем отличие переменной типа `string` от массива символов?
- 8.4 Для чего используются регулярные выражения?

Практическая работа №17

Разработка объектов на JavaScript

1 Цель работы

- 1.1 Научиться описывать прототипы объектов и создавать объекты на JavaScript;
- 1.2 Закрепить навык составления программ методами процедурного и объектно-ориентированного программирования.

2 Литература

- 2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.18-19.
- 2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.15.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Создать функцию-конструктор User с двумя параметрами для присваивания полям объекта this значения логина (login) и пароля (password).
- 5.2 Создать объект класса User, изменить его логин и пароль.
- 5.3 Добавить в функцию User функцию hello, выводящую на экран “Добро пожаловать, login”. Вызвать метод hello у созданного объекта.
- 5.4 Проверить, что логин и пароль объекта совпадают с admin и qwerty соответственно. Результат сравнения вывести на экран.

6 Порядок выполнения работы

- 6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Как программно разместить элемент управления на странице?
- 8.2 Каков синтаксис функции создания объекта на JavaScript?
- 8.3 Как обратиться к свойствам (полям) объекта, используя JavaScript?

Практическая работа №18

Разработка веб-приложений с помощью библиотеки jQuery

1 Цель работы

1.1 Научиться разрабатывать веб-приложения и настраивать их внешний вид с помощью библиотеки jQuery.

1.2 Закрепить навык верстки веб-страниц и программирования на стороне клиента.

2 Литература

2.1 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.21.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Подключение и применение jQuery

Изучить, записать и проверить работу следующего примера:

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
  <h2>Заголовок страницы</h2>
  <p>Первый абзац</p>
  <p>Второй абзац</p>
  <button>Скрыть все абзацы</button>

  <script src="jquery-3.6.0.min.js "></script>
  <script>
    $(document).ready(function () {
      $("button").click(function () {
        $("p").hide();
      });
    });
  </script>
</body>
</html>
```

Вынести последний js-скрипт из html-страницы в отдельный файл.

5.2 Применение селекторов для скрытия элементов

Добавить на страницу из п.5.1 следующие кнопки:

- “Скрыть кнопку” (при нажатии должна скрываться кнопка, на которую нажали.

При реализации использовать селектор `this`);

- “Скрыть первый абзац” и “Скрыть второй абзац” (при нажатии должен скрываться соответствующий абзац). При реализации использовать селектор вида `#идентификаторЭлемента`);

- “Скрыть все элементы класса `test`” (при нажатии должны скрываться соответствующие объекты). При реализации указать у заголовка и второго абзаца класс `test`, использовать селектор вида `.классЭлемента`).

5.3 Применение селекторов для скрытия элементов

Добавить на страницу из п.5.1-5.2 следующие кнопки:

- “Показать абзацы” (при нажатии должны показываться оба абзаца. При реализации использовать метод `show()`).

- “Показать все элементы” (при нажатии должны показываться все элементы. При реализации использовать метод `show()` и селектор `*` для всех элементов тэга `body`).

5.4 Отображение и скрытие элементов на странице

Создать `html`-документ с абзацем и двумя кнопками “Скрыть быстро” и “Скрыть медленно”, подключить к нему `jQuery`. При реализации для отображения и скрытия абзаца вместо методов `hide` и `show` использовать метод `toggle()` в первой кнопке и `toggle(1000)` во второй кнопке.

6 Порядок выполнения работы

6.1 Используя `Notepad++` и браузер `Google Chrome`, выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Для чего предназначена библиотека `jQuery`?

8.2 Каков синтаксис обращения к различным типам элементов `html`-страницы с помощью `jQuery`?

8.3 Как подключить `jQuery`?

8.4 Как обработать действие пользователя с помощью `jQuery`?

9 Приложение

Базовая команда для библиотеки выглядит как:

`$(селектор).action(параметры, если требуются);` где

- `$` - предписание использовать jQuery;
- (селектор) - это запрос или элементы поиска в HTML элементах страницы;
- `action()` - это функция, которая вызывается при определенном действии или выполняет определенное действие над найденными элементами, которые удовлетворяют условиям селектора.

Селекторы:

- `window` - окно
- `this` - текущий элемент
- `тэг` - все элементы с указанным тэгом
- `.класс` - все элементы с указанным классом
- `#идентификатор` - элемент с указанным идентификатором

Селекторы могут быть скомбинированы (в этом случае они указываются через пробел)

В большинстве примеров jQuery-методы находятся внутри события документа `ready()`:

```
$(document).ready(function(){  
    // jQuery-методы размещаем здесь...  
});
```

Это необходимо для предотвращения любых срабатываний jQuery-кода до того, как документ полностью загрузится.

Это хорошая практика, чтобы дожидаться, пока документ будет полностью загружен и готов к работе с ним. Это также позволяет подключать свой JavaScript код в заголовочной части документа, а не в конце тела документа.

Синтаксис обработчика событий:

```
$(селектор).action(function(){  
    // описание действий в функции при наступлении указанного action  
});
```

Данные конкретного элемента можно получить следующими способами:

- атрибут элемента: `$("#идЭлемента").attr("название атрибута")` - значение указанного атрибута указанного элемента
- значение элемента: `$("#идЭлемента").val()`
- текст, находящийся между тэгами элемента: `$("#идЭлемента").text()`
- html-код, находящийся между тэгами элемента: `$("#идЭлемента").html()`
- значение css-свойства: `$("#идЭлемента").css("название свойства")`

Практическая работа №19

Настройка стилей веб-приложений с помощью библиотеки jQuery

1 Цель работы

1.1 Научиться разрабатывать веб-приложения и настраивать их внешний вид с помощью библиотеки jQuery.

1.2 Закрепить навык верстки веб-страниц и программирования на стороне клиента.

2 Литература

2.1 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.21.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Изменение цветов строк таблицы

Создать html-документ с таблицей из двух столбцов и пяти строк, подключить к нему jQuery:

- при загрузке страницы нечетные строки таблицы должны окрашиваться в серый цвет (при реализации использовать код `$("tr:even").css("background-color", "код или название цвета")`),

- при нажатии на кнопку “Зеленые строки” четные строки таблицы должны окрашиваться в зеленый цвет (при реализации использовать селектор `tr:odd`),

- при нажатии на строку таблицы она должна окрашиваться в желтый цвет (при реализации использовать селектор `this`),

- при нажатии на строку таблицы, если она окрашена в желтый цвет, изменять цвет на белый (при реализации использовать селектор `this`). При определении текущего цвета учесть, что цвет хранится в RGB.

5.2 Инвертирование цветовой схемы

Создать html-документ с тремя абзацами текста, подключить к нему jQuery и файл стилей. Добавить для абзацев стиль зеленый текст, черный фон. При нажатии на абзац цвета его фона и текста должны меняться местами (у цвета фона - исходный цвет текста, у цвета текста - исходный цвет фона).

При реализации вместо использования условий можно обменивать между собой значения цвета фона и цвета текста.

5.3 Изменение настроек шрифта

Создать html-документ с текстом, подключить к нему jQuery и файл стилей. Добавить два блока размером 50x50 со сплошной черной границей и с текстом А- и А+, при нажатии на которые должно выполняться уменьшение и увеличение текста абзацев на 2 пикселя соответственно.

5.4 Настройка стиля темы из трех предустановленных

Создать html-документ с текстом, подключить к нему jQuery и файл стилей. Создать три стиля уровня класса:

- день (текст: black, фон: white);
- ночь (текст: rgb(250,250,250), фон: rgb(50,50,50));
- сепия (текст: #704214, фон: #efc9a4).

Добавить три блока размером 50x50 со сплошной черной границей с цветом (белый, темно-синий, бежевый для дня, ночи и сепии соответственно) и идентификаторами.

При нажатии на соответствующий блок должна меняться цветовая схема страницы путем получения идентификатора, сбрасывания текущих классов и задания нового согласно выбранному.

Для установки класса используется addClass, для сброса — removeClass, для поочередного сброса и установки — toggleClass.

6 Порядок выполнения работы

- 6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Как настроить цвет фона с помощью jQuery?
- 8.2 Как настроить размер текста с помощью jQuery?
- 8.3 Как получить содержимое html-элемента с помощью jQuery?

Практическая работа №20

Обработка событий таймера на JavaScript

1 Цель работы

- 1.1 Научиться обрабатывать события таймера в программах на JavaScript;
- 1.2 Закрепить навык разработки и тестирования приложений на языке JavaScript;
- 1.3 Закрепить навык составления программ методами процедурного и событийно-ориентированного программирования.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.22.

2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.20..

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Отображение текущего времени

Реализовать отображение текущего времени на странице при открытии страницы:

```
var now = new Date();  
var formatted = now.getHours() + ":" + now.getMinutes() + ":" + now.getSeconds();  
Изменить вывод так, чтобы данные отображались в формате чч:мм:сс.
```

Для того, чтобы время на странице менялось каждую секунду, использовать функцию:

```
setInterval(function() {  
    // описание действий в функции,  
    // выполняющихся каждый заданный интервал  
}, интервал в миллисекундах);
```

- 5.2 Реализовать рисование квадрата и его перемещение сверху вниз в пределах canvas по таймеру.

5.3 Реализовать рисование круга и его перемещение в пределах canvas по аналогии с движением бильярдного шара (т.е. шар должен отталкиваться от стен при столкновении). Скорость по x и y должны генерироваться случайным образом.

5.4 Реализовать рисование кругов и их перемещение в пределах canvas по аналогии с движением бильярдного шара (т.е. шар должен отталкиваться от стен при столкновении). Количество шаров указывается пользователем в поле ввода, движение

начинается после нажатия кнопки. Скорость по x и y для каждого шара должны генерироваться случайным образом.

6 Порядок выполнения работы

- 6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Какова общая форма написания обработчиков событий?
- 8.2 Какие виды событий таймера можно обработать в программах на JavaScript?
- 8.3 Как можно запустить и остановить таймер?
- 8.4 Как изменить период срабатывания таймера?

Практическая работа №21

Обработка событий мыши на JavaScript

1 Цель работы

- 1.1 Научиться обрабатывать события мыши в программах на JavaScript;
- 1.2 Закрепить навык разработки и тестирования приложений на языке JavaScript;
- 1.3 Закрепить навык составления программ методами процедурного и событийно-ориентированного программирования.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.20.

2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.20.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Добавить на страницу три блока (ширину, высоту и цвет задать через стили). Реализовать вывод на странице и в консоль координат курсора мыши при движении относительно страницы и относительно текущего блока в формате (x;y), используя обработку события `mousemove`.

5.2 Изменять цвет одного из блоков в зависимости от того, какая клавиша мыши была нажата, используя обработку события `mousedown`. Например, при нажатии на левую кнопку — красный цвет, на правую — желтый.

5.3 Добавить на страницу изображение и реализовать его перетаскивание, используя обработку события `ondragstart`.

5.4 Пропорционально изменять размер изображения в зависимости от того, вверх или вниз прокручено колесико мыши, используя обработку события `mousewheel`. При прокрутке вверх размер должен увеличиваться (на % или количество пикселей), при прокрутке вниз — уменьшаться.

6 Порядок выполнения работы

- 6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Как добавить прослушиватель на события мыши?

8.2 Какие виды событий мыши можно обработать в программах на JavaScript?

8.3 Как получить новые координаты мыши?

Практическая работа №22

Обработка событий клавиатуры на JavaScript

1 Цель работы

- 1.1 Научиться обрабатывать события клавиатуры в программах на JavaScript;
- 1.2 Закрепить навык разработки и тестирования приложений на языке JavaScript;
- 1.3 Закрепить навык составления программ методами процедурного и событийно-ориентированного программирования.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.20.

2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.20.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Реализовать отображение в консоли кода нажатой клавиши и словесное уточнение, если были нажаты стрелки, Enter, Esc, Alt, Control, Shift.

5.2 Добавить на страницу изображение четырех кнопок: вверх, вниз, влево, вправо. Расположить их так же, как они расположены на клавиатуре (в один ряд по порядку «влево», «вниз», «вправо», над кнопкой «вниз» - кнопку «вверх»). При нажатии на клавишу-стрелку внешний вид соответствующей кнопки должен изменяться (например, вокруг нее появляется тень или меняется цвет кнопки), при отжатии клавиши-стрелки — возвращаться к исходному.

5.3 Реализовать на странице рисование изображения из внешнего файла (например, танк, корабль, пакман, другой игровой персонаж). Считать, что отображается вид сбоку на объект. При нажатии на клавиши-стрелки объект должен сдвигаться в указанную сторону.

5.4 Реализовать перемещение мяча для пинг-понга по таймеру подобно движению бильярдного шара. Если мяч касается правой границы, то он продолжает двигаться за нее, а пользователь проиграл. Пользователю требуется перемещать вдоль правой границы при помощи клавиш вверх и вниз прямоугольник, от которого будет отталкиваться мяч.

6 Порядок выполнения работы

- 6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Как добавить прослушиватель на события клавиатуры?
- 8.2 Какие виды событий клавиатуры можно обработать в программах на JavaScript?
- 8.3 Как обработать нажатие спецклавиш?

Практическая работа №23

Разработка интерфейса интерактивного игрового веб-приложения

1 Цель работы

1.1 Научиться проектировать и создавать интерфейс интерактивных веб-приложений..

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.5-6.

2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.13.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Создать веб-страницу с экраном приветствия. На экране приветствия должны отображаться инструкция игры «Кликер», поле ввода имени игрока и кнопка «Играть» (для перехода к игровому экрану).

5.2 Создать веб-страницу с игровым экраном. На игровом экране должно отображаться имя игрока, время в формате мм:сс и счет, кнопка «Выход» (для перехода к экрану результатов).

5.3 На игровом экране должен быть задан фон и отображаться круг с радиальной заливкой от центра или внешнее изображение (на этот объект игрок должен будет нажимать).

5.4 Создать веб-страницу с экраном результатов. На экране результатов должна отображаться таблица рекордов, в ней должны отображаться 5 результатов с указанием места.

6 Порядок выполнения работы

6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

- 8.1 Как изменить фон веб-страницы?
- 8.2 Как отобразить поле ввода на странице?
- 8.3 Как отобразить кнопку на странице?
- 8.4 Как подключить изображение к веб-странице?
- 8.5 Какой тэг используется для отображения таблиц?

Практическая работа №24

Разработка интерактивного игрового веб-приложения

1 Цель работы

- 1.1 Научиться создавать интерактивные приложения на языке JavaScript;
- 1.2 Закрепить навык составления программ методами процедурного и событийно-ориентированного программирования.

2 Литература

- 2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.18-19.
- 2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.13.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

Все задания выполняются на игровом экране из ПР№23

5.1 Внести изменения в игровой экран: на экране игры должно отображаться время, прошедшее со старта игры в формате мм:сс. Для реализации разработать js-сценарий.

5.2 Внести изменения в игровой экран: на экране игры должен отображаться текущий счет (увеличивается на 5 при нажатии на левую кнопку мыши, на 10 — при нажатии на правую). Щелчок должен выполняться по кругу/изображению.

5.3 Внести изменения в игровой экран: реализовать возможность постановки игры на паузу (останавливается время и по центру выводится сообщение «Пауза», игровой процесс становится недоступен). Постановка должна выполняться при нажатии пробела.

5.4 Внести изменения в игровой экран: реализовать возможность запуска игры после паузы (запускается время и пропадает сообщение «Пауза», игровой процесс становится доступен). Запуск должен выполняться при нажатии пробела после паузы.

6 Порядок выполнения работы

- 6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Какие приложения называются интерактивными?

8.2 Какие приложения называются мультимедийными?

8.3 Какие преимущества у написания объектно-ориентированных приложений?

Практическая работа №25

Установка и настройка локального сервера

1 Цель работы

- 1.1 Научиться устанавливать и применять локальный WAMP-сервер;
- 1.2 Изучить настройки php.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.24.

2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.2.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Скопировать папку с локальным сервером Denwer в папку C:\Temp\[название группы].

Открыть папку «denwer» и запустить на выполнение файл Run.exe, дождаться запуска всех служб локального сервера.

5.2 Открыть браузер, ввести в адресной строке:

`http://localhost/`

В случае корректной установки в браузере будет выведена страница с текстом «Ура, заработало!».

5.3 Для вывода подробной информации о настройках php в папке Denwer\home\localhost\www создать файл info.php со следующим содержимым:

```
<?php
// Показывать всю информацию, по умолчанию INFO_ALL
phpinfo();
// Показывать информацию только о загруженных модулях.
phpinfo(INFO_MODULES);
?>
```

5.4 В браузере открыть страницу `http://localhost/info.php` и изучить полученный результат.

5.5 Перейти в папку Denwer\usr\local\php5 и изучить файл конфигурации php.ini.

6 Порядок выполнения работы

- 6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Для чего применяется локальный WAMP-сервер?
- 8.2 Из каких компонентов состоит WAMP-сервер?
- 8.3 Какая функция применяется для просмотра настроек php?
- 8.4 Для чего применяется Apache-сервер?
- 8.5 Каковы особенности разработки с использованием WAMP-сервера?
- 8.6 Какие WAMP-серверы существуют?

Практическая работа №26

Встраивание PHP-сценариев в веб-страницы

1 Цель работы

- 1.1 Научиться разрабатывать php-сценарии;
- 1.2 Научиться изолировать php-код от html-кода;
- 1.3 Закрепить навык тестирования и отладки приложений.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.25.

2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.3.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Создание php-сценария

Создать php-страницу со следующим содержимым:

```
<html>
<head>
  <title>Тест PHP</title>
</head>
<body>
  <?php
    echo '<p>Привет, мир!</p>';
  ?>
</body>
</html>
```

Протестировать результат в браузере. Открыть код страницы и убедиться, что php-сценарий транслировался в html-код.

- 5.2 Изолирование php-сценария

Модифицировать созданную php-страницу, заменив php-вставку на следующий код:

```
<?php if ($expression == true): ?>
```

Это будет отображено, если выражение истинно.

```
<?php else: ?>
```

В ином случае будет отображено это.

```
<?php endif; ?>
```

Перед запуском в браузере задать переменной `$expression` произвольное значение. Протестировать результат в браузере с различными значениями переменной `$expression`.

5.3 Применение условия в php-сценарии

Заменить в php-сценарии изолирование с использованием условий на один тэг `<?php ... ?>` с командой `echo` и операторами `if-else`. Протестировать результат в браузере.

5.4 Объявление переменных

Добавить в сценарий две переменных `$a` и `$b`, присвоить им числовые значения и вывести результат сложения этих переменных на экран, используя `echo`.

5.5 Добавление комментариев

Добавить в сценарий комментарий к каждой строке php-кода.

6 Порядок выполнения работы

6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Как встроить php-сценарий в html-код?

8.2 Какой способ является более эффективен при выводе больших блоков текста: изолирование с использованием условий или отправка текста с помощью функций `echo` или `print`?

8.3 Какая общая форма изолирования php-сценариев с помощью условий?

8.4 Какие пары тэгов могут применяться для обозначения PHP-кода?

8.5 Что такое «интерпретатор»?

8.6 Как добавить комментарии в php-сценарий?

Практическая работа №27

Включение внешних модулей в PHP-сценарии

1 Цель работы

- 1.1 Научиться разрабатывать php-сценарии;
- 1.2 Научиться создавать и встраивать в php-сценарий внешние php-модули;
- 1.3 Закрепить навык тестирования и отладки приложений.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.25.

2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – п.5.2.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Создание и подключение внешнего php-сценария

5.1.1 Создать файл vars.php со следующим сценарием:

```
<?php
$color = 'зеленое';
$fruit = 'яблоко';
?>
```

5.1.2 Подключение php-сценария

Создать файл test.php со следующим сценарием:

```
<?php
echo "До подключения: $color $fruit<br>";
include 'vars.php';
echo "После подключения: $color $fruit<br>";
?>
```

5.1.3 Открыть файл test.php и объяснить полученный результат.

5.2 Для проверки того, что внешний модуль удалось подключить, добавить в test.php следующий код:

```
if ((include 'vars.php') == TRUE) {
    echo 'Модуль успешно подключен';
    ...
}
```

```
}
```

5.3 Самостоятельно изучить в отдельных файлах подключение внешних модулей при помощи команд `include_once`, `require`, `require_once`. Объяснить отличия в выводе результата и подключении, если подключаемый файл будет отсутствовать.

5.4 Реализовать подключение внешнего модуля внутри функции:

```
<?php
```

```
function foo()
```

```
{
```

```
    global $color;
```

```
    include 'vars.php';
```

```
    echo "Тест: $color $fruit";
```

```
}
```

```
/* vars.php в той же области видимости, что и foo(), *
```

```
* поэтому $fruit НЕ доступен снаружи этой области      *
```

```
* $color доступен, поскольку переменная глобальная */
```

```
foo();           // A green apple
```

```
echo "Тест: $color $fruit"; // A green
```

```
?>
```

5.5 Создать следующий набор php-сценариев и объяснить полученную при выводе разницу:

Файл `return.php`

```
<?php
```

```
$var = 'PHP';
```

```
return $var;
```

```
?>
```

Файл `noreturn.php`

```
<?php
```

```
$var = 'PHP';
```

```
?>
```

Файл `testreturns.php`

```
<?php
```

```
$foo = include 'return.php';
```

```
echo $foo; // выведет 'PHP'
```

```
$bar = include 'noreturn.php';
```

```
echo $bar; // выведет 1
```

```
?>
```

6 Порядок выполнения работы

- 6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 В чем преимущество разработки и применения модулей?
- 8.2 В чем отличие между командами `include` и `require`?
- 8.3 В чем отличие между командами `include` и `include_once`?

Практическая работа №28

Вывод данных в PHP-сценариях

1 Цель работы

- 1.1 Научиться выводить данные в php-сценарии, используя echo;
- 1.2 Научиться выводить данные в php-сценарии, используя print, printf;
- 1.3 Закрепить навык тестирования и отладки приложений.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.25.

2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.3.3, гл.7.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Используя команду echo, вывести результат деления двух чисел в следующем виде (вместо a, b, x подставить значения соответствующих переменных): $a/b=x$

5.2 Используя команду print, вывести результат деления двух чисел в следующем виде (вместо a, b, x подставить значения соответствующих переменных): $a/b=x$

5.3 Используя команду printf, вывести результат деления двух чисел с точностью до трех знаков после запятой в следующем виде (вместо a, b, x подставить значения соответствующих переменных): $a/b=x$

5.4 Используя команду printf, вывести текущую дату в следующем виде (количество букв соответствует количеству знаков): dd:mm:yyyy

5.5 Используя команду printf, вывести значение строковой переменной, строка должна дополняться пробелами, если она длиной меньше 20 символов.

6 Порядок выполнения работы

- 6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

- 8.1 В чем отличие между командами `echo` и `print`?
- 8.2 Каков синтаксис команды `echo`?
- 8.3 Каков синтаксис команды `print`?
- 8.4 Каков синтаксис команды `printf`?
- 8.5 Как указать формат переменной в команде `printf`?

Практическая работа №29

Разработка PHP-сценариев, содержащих операторы ветвления

1 Цель работы

1.1 Научиться разрабатывать php-сценарии методами структурного и модульного программирования;

1.2 Научиться создавать и встраивать в php-сценарий внешние php-модули.

2 Литература

2.1 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – п.4.3.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Разработать php-сценарий, определяющий, можно ли вычислить корень указанной степени из вещественного числа X , и выводящий результат, если это возможно (корень четной степени невозможно вычислить, если под корнем отрицательное число).

Подключить разработанный сценарий к основному модулю и протестировать результат в браузере.

5.2 Разработать php-сценарий, вычисляющий значение функции y :

$y = a/x$, если $x < 0$

$y = 1/\sqrt{a-x}$, если $x \geq 0$.

Подключить разработанный сценарий к основному модулю и протестировать результат в браузере.

5.3 Разработать php-сценарий, определяющий по месяцу и году является ли год високосным и количество дней в месяце. Год является високосным, если он кратен 4 и не кратен 100 или если он кратен 400.

Подключить разработанный сценарий к основному модулю и протестировать результат в браузере.

6 Порядок выполнения работы

6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 В каких случаях применяется тернарный оператор?

8.2 В чем отличие между операторами if-else и switch?

8.3 В каких случаях целесообразно заменять оператор if-else оператором switch?

8.4 Каков формат тернарного оператора?

8.5 Является ли обязательной часть else условного оператора?

8.6 Какие логические операторы могут применяться при составлении условий?

Практическая работа №30

Разработка PHP-сценариев, содержащих циклические операторы

1 Цель работы

1.1 Научиться разрабатывать php-сценарии методами структурного и модульного программирования;

1.2 Научиться создавать и встраивать в php-сценарий внешние php-модули.

2 Литература

2.1 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – п.4.4.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Разработать php-сценарий, определяющий в натуральном числе n количество цифр и сумму всех его цифр. Количество цифр можно подсчитать путем многократного деления числа на 10.

Подключить разработанный сценарий к основному модулю и протестировать результат в браузере.

5.2 Разработать php-сценарий, выводящий на экран таблицу умножения.

Подключить разработанный сценарий к основному модулю и протестировать результат в браузере.

5.3 Разработать php-сценарий, возвращающий результат возведения a в степень x (a — вещественное число, x — целое).

Подключить разработанный сценарий к основному модулю и протестировать результат в браузере.

6 Порядок выполнения работы

6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

- 8.1 Чем отличаются циклы с постусловием и предусловием?
- 8.2 В каком случае выполнится тело цикла с предусловием?
- 8.3 В каком случае выполнится тело цикла с постусловием?
- 8.4 Какое минимальное количество раз выполняется цикл с предусловием?
- 8.5 Какое минимальное количество раз выполняется цикл с постусловием?
- 8.6 В каких случаях следует применять оператор `for`?

Практическая работа №31

Разработка функций на PHP

1 Цель работы

1.1 Научиться разрабатывать php-сценарии методами структурного и модульного программирования;

1.2 Научиться создавать и использовать функции в php-сценарии.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.25.

2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – п.5.1.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Разработать php-сценарий MathLibrary.php, содержащий следующие функции пользователя: сложение, вычитание, умножение, деление, возведение в степень, факториал. У всех функций должны быть входные параметры, каждая функция должна возвращать результат.

5.2 Подключить разработанную библиотеку к основному модулю и протестировать вызов функций библиотеки.

5.3 Дополнить php-сценарий функцией help, которая должна по названию функции, переданному в параметрах, возвращать информацию о функции. Протестировать вызов функции в основном модуле.

5.4 Дополнить php-сценарий функцией info, которая должна выводить на экран информацию о библиотеке и разработчике. Протестировать вызов функции в основном модуле.

6 Порядок выполнения работы

6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое «библиотека»?

8.2 Что такое «модуль»?

8.3 В чем преимущества модульного подхода к разработке?

8.4 Как вернуть результат из функции в php?

Практическая работа №32

Разработка программ с использованием массивов на PHP

1 Цель работы

- 1.1 Научиться создавать и обрабатывать массивы и ассоциативные массивы в программах на языке php;
- 1.2 Закрепить навык разработки и тестирования приложений на языке php.

2 Литература

- 2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – п.25.5.
- 2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.6.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Создать массив, который заполнить пятью строками. Вывести элементы массива на экран, используя цикл. Каждый элемент массива должен быть выведен с отдельной строки с указанием его индекса.
- 5.2 Создать массив, который заполнить пятью строками. Вывести элементы массива на экран, используя `print_r`.
- 5.3 Создать ассоциативный массив, в котором индексами будут логины, а значениями — пароли. Вывести элементы массива на экран, используя `print_r`.
- 5.4 Определить, есть ли в ассоциативном массиве пользователь с определенным логином и паролем. Результат вывести на экран.
- 5.5 Создать массив, который заполнить случайным количеством случайных чисел. Вывести на экран отсортированный массив, количество элементов массива и сумму его элементов.

6 Порядок выполнения работы

- 6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Что такое «массив»?
- 8.2 Каким образом определяются переменные типа массив в php?
- 8.3 В чем отличие между индексированными и ассоциативными массивами?
- 8.4 Как узнать количество элементов в массиве, используя функцию php?

Практическая работа №33

Разработка программ обработки строковых данных на PHP

1 Цель работы

- 1.1 Научиться создавать и обрабатывать строки в программах на языке php;
- 1.2 Закрепить навык разработки и тестирования приложений на языке php.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.38.

2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.16.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Вывести на экран длину строки, используя функцию `strlen`.
- 5.2 Вывести на экран подстроку из исходной строки, используя функцию `substr`.
- 5.3 Удалить из строки пробелы по краям строки, используя функцию `trim` и привести ее к нижнему регистру, используя функцию `strtolower`.
- 5.4 Вывести на экран информацию о том, есть ли требуемая подстрока в строке, используя функцию `strpos`.
- 5.5 Вывести на экран количество вхождений требуемой подстроки в строке, используя функцию `strpos`.

6 Порядок выполнения работы

- 6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Как определить длину строки?
- 8.2 Как обрезать пробелы по краям строки?

8.3 Что возвращает функция `strpos`?

Практическая работа №34

Разработка объектов на PHP

1 Цель работы

- 1.1 Научиться разрабатывать php-модули средствами ООП;
- 1.2 Закрепить навык составления программ методами процедурного и объектно-ориентированного программирования.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – п.25.6.

2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – п.5.4.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Создать класс User со следующими полями:

- login (открытое поле);
- password (закрытое поле);
- email (защищенное поле).

Добавить в класс функцию hello, выводящую на экран “Добро пожаловать, login”.

5.2 Создать объект класса, изменить его логин и вызвать метод hello.

5.3 Добавить в класс User конструктор, позволяющий задать значения всем полям класса. В конструкторе должны быть параметры по умолчанию ‘user’, ‘12345’, ‘’ для логина, пароля и email соответственно.

Создать объект класса с помощью конструктора, указывая параметры и не указывая параметры.

5.4 Добавить в класс User статическую переменную zagolovok (по умолчанию: ‘h1’) и константу welcome (по умолчанию: ‘Добро пожаловать’). Использовать переменную и константу для в методе hello для указания заголовка и текста перед логином.

Перед вызовом hello изменить значение переменной zagolovok на h3 (для обращения к статической переменной и константе используется синтаксис класс::объект).

6 Порядок выполнения работы

- 6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Что такое «ООП»?
- 8.2 Каков синтаксис объявления класса в php?
- 8.3 Как получить доступ к данным объекта в php?
- 8.4 Как получить доступ к функциям объекта в php?
- 8.5 Какова общая форма конструктора в php?

Практическая работа №35

Разработка веб-форм

1 Цель работы

- 1.1 Научиться разрабатывать веб-приложения, содержащие веб-формы;
- 1.2 Закрепить навык верстки веб-страниц и программирования на стороне клиента.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.8.

2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.11.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Создать веб-форму авторизации, запрашивающую логин и пароль. На форме должны быть поля ввода логина и пароля, снабженные метками, и кнопка с подписью «ОК».

5.2 Создать веб-форму регистрации, запрашивающую логин, пароль и подтверждение пароля. На форме должны быть поля ввода логина, пароля и подтверждения пароля, снабженные метками, флажок checkbox «Я согласен с правилами» и кнопка с подписью «ОК».

5.3 Создать веб-форму, содержащую кнопку «Показать результат» (type: button), кнопку «Очистить» (type: reset), поле ввода только на чтение для вывода количества правильных ответов.

Добавить в веб-форму 4 абзаца, в каждом из которых должен быть указан вопрос и предложены 3 варианта ответа в виде переключателей (т.е. корректный ответ — только один для каждого вопроса). У переключателей задать одинаковые name в рамках каждого абзаца и value с номером ответа.

5.4 Создать веб-форму, содержащую кнопку «Показать результат» (type: button), кнопку «Очистить» (type: reset), поле ввода только на чтение для вывода количества правильных ответов.

Добавить в веб-форму 4 абзаца, в каждом из которых должен быть указан вопрос и предложены 3 варианта ответа. Выбор варианта должен быть реализован следующим образом:

- с помощью переключателей radio у первого вопроса;
- с помощью переключателей radio у второго вопроса. Вместо текста ответа должны использоваться картинки;
- с помощью выпадающего списка select у третьего вопроса;
- с помощью набора флажков checkbox у четвертого вопроса для возможности выбора двух правильных ответов на вопрос.

6 Порядок выполнения работы

6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Какие тэги применяются для создания веб-форм?

8.2 Какие тэги применяются для добавления различных элементов управления на веб-форму?

8.3 Как создать переключатель в веб-форме?

8.4 Как создать флажок в веб-форме?

8.5 Как создать поле ввода в веб-форме?

8.6 Как создать выпадающий список в веб-форме?

Практическая работа №36

Передача данных между веб-страницами

1 Цель работы

1.1 Научиться передавать данные между формами, используя http-запросы POST и GET.

2 Литература

2.1 Дронов, В. А. PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2016. – 688 с. – URL: <https://ibooks.ru/bookshelf/353560/reading>. – гл.31.

2.2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 5-е изд. / Р. Никсон. – Санкт-Петербург : Питер, 2019. – 816 с. – URL: <https://ibooks.ru/bookshelf/359215/reading>. – гл.11.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Указать у веб-формы авторизации из п.5.1 ПР№35 метод передачи данных post.

Создать php-сценарий, проверяющий, есть ли в ассоциативном массиве пользователь с указанным логином и паролем, при нажатии на кнопку «ОК».

В случае, если есть, приветствовать пользователя на странице, иначе — сообщить, что указанный пользователь не найден.

5.2 Указать у веб-формы регистрации из п.5.2 ПР№35 метод передачи данных post.

Создать php-сценарий, проверяющий, что пользователь указал все данные и совпадают пароль и подтверждение, при нажатии на кнопку «ОК».

В случае корректного ввода данных сценарий должен выводить сообщение: «Вы зарегистрированы, логин!», иначе — «Данные некорректны».

5.3 Указать у веб-формы тестирования из п.5.3 ПР№35 метод передачи данных get.

Создать php-сценарий, подсчитывающий количество правильных ответов, указанных пользователем на странице тестирования, при нажатии на кнопку «Показать результаты».

6 Порядок выполнения работы

6.1 Используя Notepad++ и браузер Google Chrome, выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Какие методы служат для отправки данных с одной веб-страницы на другую?
- 8.2 В чем отличие между методами GET и POST?
- 8.3 Как указать в форме метод, который будет использоваться для передачи данных?
- 8.4 Где отображаются параметры, переданные методом GET?