

Оглавление

Подготовка к работе, установка стартового ПО.....	2
ПРАКТИЧЕСКАЯ РАБОТА №1. Знакомство с HTML	3
ЛАБОРАТОРНАЯ РАБОТА № 1. Создание сайта средствами HTML	9
ПРАКТИЧЕСКАЯ РАБОТА №2. «Подключение стилей CSS к сайту»	9
ПРАКТИЧЕСКАЯ РАБОТА № 3. «Блочная верстка с использованием CSS»	12
ЛАБОРАТОРНАЯ РАБОТА №2. Создание сайта средствами HTML и CSS.....	18
ПРАКТИЧЕСКАЯ РАБОТА №4. Основы JavaScript.....	18
ЛАБОРАТОРНАЯ РАБОТА №3. Формы и простейшие Java-скрипты в HTML-документах.....	22
ПРАКТИЧЕСКАЯ РАБОТА №5. Изучение языка php.....	24
ЛАБОРАТОРНАЯ РАБОТА №4. Знакомство с Web-сервером и обработкой данных с помощью PHP-программ.....	30
ПРАКТИЧЕСКАЯ РАБОТА №6. Запросы SQL.....	31
ЛАБОРАТОРНАЯ РАБОТА №5. Запросы SQL через PHP	33

Подготовка к работе, установка стартового ПО

Для выполнения лабораторных работ потребуется установить ряд программ:
AMPPS

<https://www.ampps.com/>

Visual Studio Code с расширением «PHPIntelephense»

<https://code.visualstudio.com/download>

ПРАКТИЧЕСКАЯ РАБОТА №1.

Знакомство с HTML

Цель работы: познакомиться со структурой файла HTML, основными тэгами.

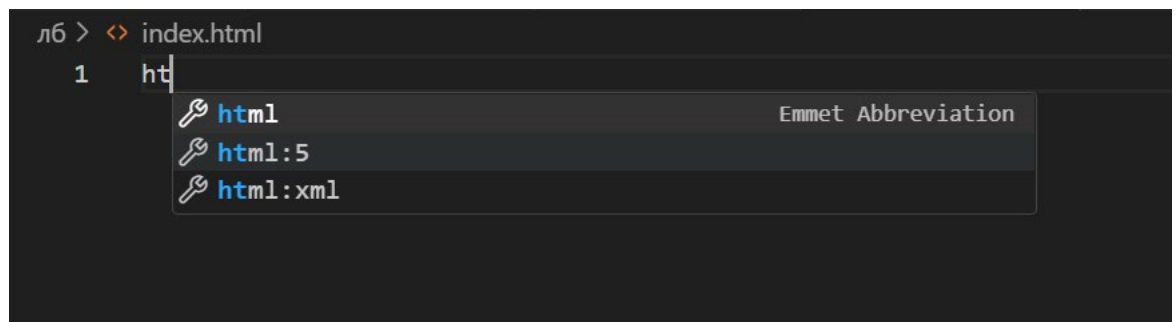
УПРАЖНЕНИЕ 1. Организация структуры веб-документа.

В своей рабочей папке найдите папку «www».

Откройте текстовый редактор «Visual Studio Code»,

Назовите файл «index.html», и укажите слово «html», и программа выдаст варианты макета. Выбираем «html.5», после чего на экране отобразится шаблон кода, в его теле напишем текст при помощи тега «p», который должен вывестись в браузере.

Тег `<>` – состоит из имени, за которым может следовать необязательный список атрибутов. Весь тег (вместе с атрибутами) заключается в угловые скобки `<>`: `<имя_тега [атрибуты]>`. Как правило, теги являются парными и состоят из начального и конечного тегов, между которыми и помещается информация.



```
лб > <> index.html
1  ht
    html
    html:5
    html:xml
```

Emmet Abbreviation



```
лб > <> index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9
10 </body>
```

Настройте автосохранение. Откройте страницу в браузере.

Перейдите в «Visual Studio Code» и изучите между парными тэгами `<head>` и `</head>` строки:

Эти строки служебной информации не видны на странице сайта, но позволяют правильно отображать информацию на любом ПК, в какой стране бы он не находился, независимо от настроек.

Исключением является тег `<title>`, содержание которого отображается в браузере как название вашей страницы (окна документа). Введите название страницы:

`<title> Практическая работа №1</title>`

Сохраните Ваш документ (Ctrl+S), перейдите в окно браузера и нажмите кнопку «Обновить».

Отметьте произошедшие изменения.

УПРАЖНЕНИЕ 2. Оформление фона веб-страницы.

Установка в качестве фона цвета.

В качестве параметра тега `<body>` установите `bgcolor="blue"`. Т.е. у Вас должна получиться следующая строка:

```
<body bgcolor="blue">
```

Сохраните Ваш документ (Ctrl+S), перейдите в окно браузера и нажмите кнопку «Обновить».

Отметьте произошедшие изменения.

Перейдите в «Visual Studio Code» и измените "blue" на "#f05fbb".

Сохраните Ваш документ (Ctrl+S), перейдите в окно браузера и нажмите кнопку «Обновить».

Отметьте произошедшие изменения.

УПРАЖНЕНИЕ 3. Оформление верхней части веб-страницы.

Теперь ВСЕ(!) теги будем размещать между парными тегами `<body>` и `</body>`.

Как правило, в верхней части веб-страницы размещается заголовок сайта в виде текста или логотипа (картинки). Можете использовать любой логотип.

После тега `<body>` введите следующий участок кода:

```

```

Сохраните Ваш документ (Ctrl+S), перейдите в окно браузера и нажмите кнопку «Обновить».

Отметьте произошедшие изменения.

Для того, чтобы разместить логотип по центру страницы, заключите только что введенную Вами строку между тегами `<center>` и `</center>`.

Теперь отделим заголовок от остального содержимого страницы горизонтальной линией. В следующей строке введите текст:

```
<hr width="80%" size="5" color="white">
```

Сохраните Ваш документ (Ctrl+S), перейдите в окно браузера и нажмите кнопку «Обновить».

Отметьте произошедшие изменения.

Самостоятельно: установите цвет, ширину и размер линии на Ваш вкус.

УПРАЖНЕНИЕ 4. Оформление основной части веб-страницы.

Теперь мы выведем данные: о чем, собственно, сайт.

В качестве пропедевтической части отобразим на странице следующий текст (просто наберите

следующей строкой его после тега `<hr...>`):

На нашей планете есть множество естественных и искусственных объектов, которые поражают наше воображение. Это пирамиды в Египте и Мексике, истуканы на острове Пасхи, дворец Тадж-Махал в Индии, Стоунхендж в Великобритании, Ниагарский водопад в Канаде, Большой каньон в Колорадо и многие другие. Их называют «Чудеса света». Ежегодно сотни тысяч туристов приезжают посмотреть на эти диковины, принося ощутимый доход государствам, на территории которых они расположены.

Сохраните Ваш документ (Ctrl+S), перейдите в окно браузера и нажмите кнопку «Обновить».

Отметьте произошедшие изменения.

Теперь будем этот текст форматировать.

Прежде всего, отметим его как абзац. Для этого заключим текст между тегами `<p>` и `</p>`.

Теперь установим выравнивание при помощи параметра `align="justify"` (выравнивание по ширине).

Т.е. должно получиться нижеследующее:

```
<p align="justify"> На нашей...расположены.</p>
```

Измените значение параметра align на "left" затем на "right" и посмотрите, как меняется позиция текста на странице.

Установим теперь атрибуты текста (цвет, размер, гарнитуру). Для этого воспользуемся тегом

. Заключите текст между тегами и . Должно получиться следующее:

```
<p align="justify"><font face="Arial" size="4" color="navy">На нашей...  
...расположены.</font></p>
```

Измените значение параметров face, size и color по Вашему вкусу.

Для акцентирования внимания на отдельных словах используем парные теги и (жирный), <i> и </i> (курсив), <u> и </u> (подчеркнутый). Заключите между этими все названия

объектов, например:

...дворец Тадж-Махал в Индии,...

и

...<u>Большой каньон</u> в Колорадо...

и

... <i>пирамиды</i> в Египте...

Добавим следующий абзац:

Однако не обязательно ехать в далекие страны, чтобы увидеть удивительные природные ландшафты, памятники старины и произведения архитектуры. В нашей стране тоже есть уникальные объекты, которые находятся в одном из самых красивых городов России. Представляем Вам Санкт-Петербург.

Установите необходимые параметры абзаца и шрифта (по аналогии с предыдущим).

Создадим маркированный список чудес:

```
<ul type="circle">  
<li> Медный Всадник</li>  
  <li> Государственный Эрмитаж </li>  
  <li> Крейсер «Аврора» </li>  
  <li> Здание главного Адмиралтейства </li>  
  <li> Петропавловская крепость </li>  
  <li> Львы Санкт-Петербурга </li>  
  <li> Кунсткамера </li>  
  <li> Стрелка Васильевского острова </li>  
  <li> Исаакиевский собор </li>  
  <li> Храм Спаса-на-Крови </li>  
  <li> Смольный монастырь </li>  
  <li> Летний сад </li>  
</ul>
```

Для того, чтобы список выводился по центру страницы, необходимо создать таблицу 1x1, т.е.

добавить следующие теги:

```
<table align="center">  
<tr>  
<td>  
<ul type="circle">  
<li> Медный Всадник</li>  
.....
```

```

.....
<li> Летний сад </li>
</ul>
</td>
</tr>
</table>

```

Используя уже известный Вам тег и его параметры face, size и color, установите подходящие размер, цвет и гарнитуру шрифта списка (можно всего списка сразу – между тегами и или каждого элемента в отдельности - между тегами и).

Теперь создадим таблицу, которая будет содержать изображения чудес. Поскольку их 12, то таблица будет иметь 4 строки и 3 столбца. Введите следующий участок кода:

```

<table border="1" align="center">
  <tr align="center" valign="center">
    <td> Медный Всадник</td>
    <td> Государственный Эрмитаж</td>
    <td> Крейсер «Аврора» </td>
  </tr>
  <tr align="center" valign="center"></tr>
    <td> Здание главного Адмиралтейства</td>
    <td> Петропавловская крепость</td>
    <td> Львы Санкт-Петербурга </td>
  </tr>
  <tr align="center" valign="center">
    <td> Кунсткамера</td>
    <td> Стрелка Васильевского острова</td>
    <td> Исаакиевский собор</td>
  </tr>
  <tr align="center" valign="center"></tr>
    <td> Храм Спаса-на-Крови</td>
    <td> Смольный монастырь</td>
    <td> Летний сад</td>
  </tr>
</table>

```

Теперь названия чудес заменим их изображениями: вместо текста «Медный всадник» между тегами <td> и </td> вставьте участок кода

```



```

Аналогично заменить текст на изображения остальных достопримечательностей.

УПРАЖНЕНИЕ 5. Вставка ссылок.

Внизу главной страницы нашего сайта расположена таблица, содержащая уменьшенные изображения. Сейчас добавим участок кода, позволяющий при щелчке на фотографии открыть отдельную страницу с ее полноразмерным изображением.

Для этого коды в ячейках таблицы заключим в следующие теги <a> и :

```

<a href="images/medvsad.jpg" target="_blank"></a>

```

Аналогично сделайте для каждой ячейки таблицы.

УПРАЖНЕНИЕ 6. Создание меню.

Поскольку наш сайт будет содержать несколько страниц, нам потребуется создать меню для перехода между ними. Для этого после строки <hr WIDTH="80%" SIZE="5" color="white"> (вверху сайта – горизонтальная линия) добавьте следующий участок кода:

```

<table border="2" width="80%" align="center" bgcolor="#f5f5f5">

```

```

<tr>
<th><a href="index.html">Главная</a></th>
<th bgcolor="Lime"><a href="erm.html">Государственный Эрмитаж</a></th>
<th bgcolor="Cyan"><a href="admir.html">Здание главного Адмиралтейства</a></th>
<th bgcolor="DeepPink"><a href="petrop.html">Петропавловская крепость</a></th>
<th bgcolor="Gold"><a href="smoln.html">Смольный монастырь</a></th></tr>
</table>

```

Примечание: тег <th> аналогичен тегу <td>, только текст в ячейке создается полужирный с выравниванием по центру.

Итак, мы создали меню перехода, однако, при попытке перейти по ссылке выдается сообщение об ошибке – страница не найдена.

УПРАЖНЕНИЕ 7. Создание других страниц сайта.

Добавим ссылки на уже готовые страницы.

Добавьте следующие строчки кода:

```

<table border="2" width="80%" align="center" bgcolor="#f5f5f5">
<tr>
<th><a href="index.html">Главная</a></th>
<th bgcolor="Lime"><a href="https://hermitagemuseum.ru/">Государственный
Эрмитаж </a></th>
<th bgcolor="Cyan"><a href="https://www.spb-guide.ru/admiraltejstvo.htm">Здание
главного Адмиралтейства </a></th>
<th bgcolor="DeepPink"><a href="https://petropavlovskaya.org/">Петропавловская
крепость </a></th>
<th bgcolor="Gold"><a href="https://mos-holidays.ru/spb/dostoprimechatelnosti/smolnyj-
monastyr/">Смольный монастырь </a></th></tr>
</table>

```

Теперь верхнее меню позволит осуществлять переходы между страницами.

УПРАЖНЕНИЕ 8. Добавление аудио контента.

После меню создайте с помощью тега <audio> добавьте на страницу Гимн Санкт-Петербурга. Добавьте следующие строчки кода:

```

<p> <font face="SEGOE PRINT" size="4" color="red"> Гимн Санкт-Петербурга
</font><br>
<audio controls
<source src="audio/hymn.mp3">
</audio>
</p>

```

УПРАЖНЕНИЕ 9. Формы.

В конце страницы создадим пользовательскую форму.

Структура любой пользовательской формы описывается тегом-контейнером <form></form>, внутри которого могут располагаться теги и параметры составляющих элементов HTML-формы.

В конце документа в подвале <footer> </footer> добавим теги формы <form></form>.

Добавим в форму поля для ввода логина и пароля, выбор из ниспадающего списка своего города, а также выбор своей возрастной группы с элементом выбора типа radio и увлечений с элементом выбора типа checkbox. В конце формы добавим кнопки для сброса введенных данных в форму и для отправления данных на Web-сервер (пока отправка происходить не будет, но саму кнопку запланируем).

Должен получиться такой фрагмент кода:

```

<footer>
<form method="get">
<h3> Анкета пользователя</h3>

```

```

<table>
  <tr>
    <td> Логин: </td>
    <td> <input type="text" name="login" size="30"></td>
  </tr>
  <tr>
    <td> Пароль: </td>
    <td> <input type="password" name="password" size="10"></td></tr>
  <tr>
    <td> Город: </td>
    <td> <select>
      <option value="a"> Москва
      <option selected=2 value="b"> Санкт-Петербург
      <option value="c"> Казань
      <option value="d"> Мурманск
      <option value="e"> Другой...
    </select> </td></tr>
  <tr>
    <td> Почта: </td>
    <td> <input type="email" name="mail" size="30"></td>
  </tr>
</table>

```

```

<h4> Укажите свою возрастную группу </h4>
<input type="radio" name="age" value="child"> 7-12 лет
<input type="radio" name="age" value="junior" checked> 13-20 лет
<input type="radio" name="age" value="adult"> от 20 лет

```

```

<h4> Укажите свои увлечения </h4>
<input type="checkbox" name="hobby" value="computers"> Компьютеры
<input type="checkbox" name="hobby" value="art"> Литература
<input type="checkbox" name="hobby" value="music"> Музыка
<input type="checkbox" name="hobby" value="avto"> Автомобили
<input type="checkbox" name="hobby" value="sport"> Спорт
<br>

```

```

<input type="submit" value="Отправить">
<input type="reset" value="Очистить">

```

```

</form>
</footer>

```

Сохраните Ваш документ (Ctrl+S), перейдите в окно браузера и нажмите кнопку «Обновить».

Отметьте произошедшие изменения. Попробуйте заполнить поля формы и очистить их.

Оцените общий вид получившейся страницы. Если что-то, на Ваш взгляд, выбивается из общего стиля и требует изменений – правки допускаются.

Итог:

По итогу выполнения данной работы мы рассмотрели структуру HTML-документа, а также основные теги. По окончании работы ответьте на контрольные вопросы для закрепления знаний.

Отчет по практическому заданию должен содержать описание ваших действий при выполнении практического задания с приложенным итоговым скриншотом экрана и фрагментов кода.

Ответить на контрольные вопросы

1. Язык тегов HTML.

2. Структура HTML-документов.
3. Характеристика форматирования абзаца и шрифта.
4. Цветовая гамма HTML-документа.
5. Относительный и абсолютный адреса.
6. Основные типы списков в HTML-документах.
7. Основные методы создания таблиц.
8. Основные теги для создания HTML-форм.

ЛАБОРАТОРНАЯ РАБОТА № 1.

Создание сайта средствами HTML

Цель работы: научиться создавать и редактировать HTML-файл.

Создайте самостоятельно страницу сайта, по аналогии с практической работой №1, на одну из предложенных тем:

- Футбольная команда;
- Порода животных;
- Город/Страна;
- Компьютерная игра;
- Автомобиль;
- Бренд одежды;
- Итальянская/Китайская/Русская/Индийская кухня;
- Своя тема...

Обязательно применить:

- выравнивание текста;
- шрифты;
- цвета;
- таблицы;
- списки;
- ссылки;
- картинки;
- формы.

Итог:

Отчет по лабораторной работе должен содержать описание ваших действий при создании страницы на определенную тему с приложенным итоговым скриншотом экрана и фрагментами кода.

ПРАКТИЧЕСКАЯ РАБОТА №2.

«Подключение стилей CSS к сайту»

Цель работы: Изучить методы подключения стилей CSS.

В своей рабочей папке найдите папку «www».

Откройте текстовый редактор «Visual Studio Code»,

Назовите файл «index2.html, и укажите слово «html», и программа выдаст варианты макета. Выбираем «html.5», после чего на экране отобразится шаблон кода, в его теле напишем текст при помощи тега «p», который должен выводиться в браузере.

Текст:

Польза овощей и фруктов

<p> Значение овощей, фруктов и ягод в питании человека общеизвестно. Они вкусны, питательны, полезны для здоровья. Овощи и фрукты не обладают большой энергетической ценностью, но являются поставщиками важнейших нутриентов (пищевых веществ, необходимых для нормального осуществления обмена веществ), от которых напрямую зависит здоровье человека. Витамины, минеральные вещества, легкоусвояемые углеводы, пищевые волокна – вот далеко не полный перечень полезных веществ, которыми богаты эти дары природы. </p>

<p> Овощи и фрукты – основные поставщики пищевых волокон, играющих важную роль в функционировании желудочно-кишечного тракта, нормализации жирового обмена, выведении из организма холестерина. Пищевые волокна не перевариваются и не усваиваются в желудочно-кишечном тракте, но перерабатываются полезной микрофлорой кишечника. </p>

<p> Рекомендуется потреблять не менее 400 граммов овощей и фруктов ежедневно. Но поскольку овощи и фрукты неравноценны по содержанию различных питательных веществ, желательно составлять недельное меню таким образом, чтобы ежедневно на столе присутствовало широкое разнообразие растительных продуктов. Причем полезны не только свежие овощи и фрукты, а также замороженные и сушеные, которые сохраняют свои уникальные свойства. </p>

В данной лабораторной работе рассматриваются способы подключения стилей CSS3 к сайту. Существует три способа подключения стилей к сайту.

Первый способ используется больше для обучения или для минимального изменения свойств отдельного тега, он увеличивает вес страницы и тем самым увеличивает время загрузки страницы, помимо этого увеличивает количество кода, что существенно усложняет работу с ним. Этот способ предполагает написание стилей в атрибуте тега style.

Второй способ используется чаще чем первый, он подходит если используется небольшое количество стилей, либо, когда возникает необходимость на отдельной странице сайта подключить стили, не используемые больше нигде на сайте. Стили во втором способе подключаются к странице в части с использованием тега

Третий самый распространенный способ подключения стилей - это подключение внешнего файла со стилями, который имеет расширение .css и содержит внутри себя правила и свойства CSS. Данный способ имеет множество достоинств перед предыдущими и существенно облегчает работу с оформлением сайта.

УПРАЖНЕНИЕ 1. Стили через атрибут.

Способ 1. Подключение стилей с использованием атрибута тега style. Как известно из предыдущих работ по HTML, мы уже знаем что каждый тег имеет определенный набор атрибутов. Некоторые атрибуты уникальные и применяются лишь к определенным тегам, другие же универсальные и могут быть применены к большинству тегов. Одним из таких уникальных атрибутов является атрибут style, который позволяет добавить стили CSS к данному тегу. Синтаксис:

Применим к первому абзацу стили через атрибут. Введите следующий участок кода:

```
<p style= "margin: 10px 0; padding: 8px; border: 1px solid #999; color: blueviolet;" >
```

Значение овощей, фруктов ... эти дары природы. </p>

Все стили записываются через точку с запятой. Значение атрибута помещается в кавычки. Данный способ не рекомендуется использовать везде и постоянно, так как он убивает напрочь суть универсальности стилей. Помимо этого, данный способ написания стилей увеличивает вес страницы, что влияет на скорость загрузки сайта. Данный стиль будет применяться в приоритете другим стилям так как обладает большей значимостью.

УПРАЖНЕНИЕ 2. Стили через тег <style>.

Способ 2. Подключение стилей с использованием тега <style> в части <head>.

Добавьте после трёх абзацев еще фрагмент текста, заключенный в контейнер `div` и еще один абзац

`<div>`**Овощи и фрукты – действительно кладезь витаминов и минералов. Они богаты кальцием, магнием, железом, цинком и витаминами А, С, Е и К.**`</div>`

`<p>` Мало кто знает, что, помимо витаминов минеральных веществ и клетчатки, полезные фрукты и овощи содержат еще и минорные компоненты, которые служат защитой организма от вредного воздействия окружающей среды. `</p>`

Элемент `<div>` является блочным элементом и предназначен для выделения фрагмента документа с целью изменения вида содержимого. Как правило, вид блока управляется с помощью стилей.

Второй способ подключения стилей является более распространенным, он позволяет создать стили для каждой отдельной страницы используя парный тег `<style>`. Чтобы не описывать каждый раз стиль внутри тега, можно выделить стиль во внешнюю таблицу стилей, а для тега добавить атрибут `class` или `id` с именем селектора.

Применим ко второму абзацу стили через тег `<style>`.

Введите следующий участок кода:

```
<style>
div{
width: 200px;
background: #cfcf32;
padding: 5px;
padding-right: 20px;
border: solid 1px black;
float: left;
}

#menu {
background-color: #999;
color: azure;
}

.menu2 {
background-color: aquamarine;
color: black;
}
</style>
```

`<p id="menu">` **Овощи и фрукты – основные микрофлорой кишечника.** `</p>`

`<p class="menu2">` **Рекомендуется потреблять уникальные свойства.** `</p>`

`<div>`**Овощи и фрукты Е и К.** `</div>`

Данный способ добавляет некоторую универсальность стилям, т.е. стили, написанные таким способом, будут применяться к соответствующим тегам на всей странице. Данный способ немного увеличивает вес страницы и соответственно увеличивает время загрузки. Данный способ подходит в том случае, если вы хотите оформить какую-либо страницу вашего сайта определенными стилями, которые не будут повторяться на других веб-страницах сайта. Основное что следует запомнить это то что данные стили будут применяться только на той странице, в коде которой в части `HEAD` он будет написан.

УПРАЖНЕНИЕ 3. Стили через подключение файла.

Способ 3. Подключение внешнего файла со стилями CSS. Данный способ является универсальным и самым распространенным. Он позволяет создать стили CSS в отдельном текстовом документе, который имеет расширение `.css` (например: `style.css`). Плюсы данного способа в том, что все стили для всего сайта записываются в одном документе, что существенно облегчает дальнейшую работу с внешним видом сайта. На сайте всегда присутствуют элементы, которые на

всех страницах сайта отображаются одинаково (например, это может быть меню, шапка, подвал, какие либо кнопки и прочие элементы).

Создадим файл style.css и пропишем в нем код:

```
.paragr {  
    margin:5px;  
    padding:20px;  
    background-color:#eee;  
    border-style:dashed;  
    border-width:6px;  
    text-align:justify;  
    text-indent:30px;  
    top: 40px;  
    left: 210px;  
    width: 400px;  
}
```

Применим данный стиль к абзацу, перед этим подключив стиль с помощью строки:

```
<link rel="stylesheet" type="text/css" href="style.css">  
<p class="paragr"> Мало кто .... окружающей среды. </p>
```

Итог:

По итогу выполнения данной работы мы рассмотрели способы подключения стилей CSS к сайту HTML, рассмотрели достоинства и недостатки каждого из способов. По окончании работы ответьте на контрольные вопросы для закрепления знаний.

Отчет по практическому заданию должен содержать описание ваших действий при выполнении практического задания с приложенным итоговым скриншотом экрана и фрагментов кода.

Ответить на контрольные вопросы

1. Если к одному тегу написаны стили всеми тремя способами, то в каком порядке и с каким приоритетом они будут применяться?
2. В чем особенности каждого из способов подключения стилей?

ПРАКТИЧЕСКАЯ РАБОТА № 3. «Блочная верстка с использованием CSS»

Цель работы: Изучить возможность верстки веб-страницы с помощью тегов HTML и CSS.

В своей рабочей папке найдите папку «www».

Откройте текстовый редактор «Visual Studio Code»,

Назовите файл «index3.html, и укажите слово «html», и программа выдаст варианты макета. Выбираем «html.5», после чего на экране отобразится шаблон кода, в его теле напишем текст при помощи тега «p», который должен выводиться в браузере.

Блочная верстка – это верстка страницы с использованием слоев или блоков, в основе используются теги div.

Основной принцип такой верстки в том, что вся страница разделяется на блоки, они могут накладываться друг на друга, им можно задавать отступы с краев, в данные блоки помещаются остальные элементы страницы.

Ко всем блокам применяются стили CSS, которые показывают браузерам размер, расположение, отступы снаружи и внутри блока и много другое.

Стили мы будем прописывать с использованием тега <style> в части <head>.

УПРАЖНЕНИЕ 1. Разработка структуры.

Для начала нам необходимо понять какая структура страницы нам нужна, то есть нарисовать ее структуру.

Мы будем использовать простую структуру, которая встречается на большинстве сайтов. Она будет состоять из:

шапки, горизонтального меню, левого сайдбара, основной части и подвала.

Представим необходимую структуру графически:



После того как мы нарисовали структуру в графическом варианте мы можем определиться с тем какие теги мы будем использовать для создания структуры страницы с использованием HTML5.

```
<!DOCTYPE html>
<head>
  <title>Практическая работа №3</title>
</head>
<body>
  <header>
    Шапка сайта
  </header>
  <nav>
    Меню навигации горизонтальное
  </nav>
  <aside>
    Левый сайдбар
  </aside>
  <div >
    Основная часть
  </div>
```

```
<div></div>
<footer>
Подвал
</footer>
</body>
</html>
```

Если мы посмотрим, как выглядит наш код в браузере то увидим следующее

Шапка сайта
Меню навигации горизонтальное
Левый сайдбар
Основная часть
Подвал

На нужный результат совсем не похоже, кроме надписей. Для того чтобы наша страница приобретала нужный вид добавляем стили CSS файл.

```
body {
background: #f5f5f5;
color: #000000;
font-family: Arial, Times New Roman;
font-size: 16px;
}
header {
background: yellow;
height: 100px;
width: 100%;
}
nav {
background: #330044;
color: white;
width: 100%;
height: 50px;
}
aside {
background: #1baf5d;
float: left;
width: 10%;
height: 400px;
}
#content {
background: gray;
float: left;
width: 90%;
}
#clear {
clear: both;
}
footer {
background: #ff0404;
color: white;
height: 80px;
width: 100%;
}
```

}

После применения стилей видно, что сайт уже стал приобретать вид того, что нам нужно. Теперь разберем код стилей. Итак, начнем по порядку с тега нам необходимо задать ему ширину в 100% от окна, это делается с помощью свойства `width:100%`, и задаем высоту в 100 пикселей с помощью свойства `height:100px`. И соответственно используя свойство `background: yellow`. Значения ширины и высоты могут принимать значения в пикселях и в процентах. Что использовать зависит от конкретных ситуаций. Следующим элементом является меню навигации. При стилизации блока меню используются те же самые свойства что и при стилизации шапки. Плюс добавили свойство `color: white`, оно позволяет указать цвет текста. В качестве значений для свойств, задающих цвета могут выступать названия цветов (например, `red`, `Green`, `black`), так и шестнадцатичные значения цветов (например, `#ff00ff`, `#000000` и т.д.).

Следующим блоком у нас является сайдбар созданный с помощью тега `<aside>`, к нему применяем свойства ширины, высоты, цвет заливки и добавляется свойство `float:left`, оно позволяет задать позиционирование блока по левому краю. При задании размеров блоков нужно правильно рассчитывать ширину блоков, которые находятся рядом. Вся ширина экрана это 100%, то есть для того чтобы нам разместить два блока рядом необходимо одному блоку задать ширину X второму задать ширину Y, но главное следует всегда просчитывать чтобы сумма значений X и Y всегда равнялась 100%. В противном случае может появиться пустое пространство, либо блоки не будут умещаться и будут не структурированы, то есть какой-либо из блоков может переместиться ниже или выше другого, либо наложиться сверху. Следующий блок - это основная часть страницы, его создали с помощью тега `<div>` и присвоили ему идентификатор `#content`. Данному идентификатору мы создали стили заливки фона (`background`), ширину (`width`), высоту (`height`), и обтекаемость (`float`). Последним блоком является подвал, который мы создали с помощью тега `<footer>`. Стили для данного блока практически идентичны стилям, которые мы использовали при стилизации шапки страницы.

Далее дополним страницу контентом, применяя фреймы, а также дополнительные стили `css`: градиент, переходы для подсвечивания, анимацию.

Файл со стилями `css`:

```
header {
    background: #ffb60f;
    background: radial-gradient(circle at 80px 40px, #f9eec7, #ffb60f, #ffb60f);
    padding: 20px;
    height: 40px;
    width: 100%;
    font-weight: bold;
    font-style: italic;
    text-align: center;
    font-size: 40px;
}
nav {
    background: #330044;
    color: white;
    width: 100%;
    height: 70px;
    text-align: center;
}
aside {
    background: #1baf5d;
    float: left;
    width: 10%;
```

```
height: 600px;}
```

```
#content {  
background: #FFE4E1;  
float: left;  
width: 90%;  
height: 600px;  
}  
#clear {  
clear: both;  
}  
footer {  
background-image: url(img/3.jpg);  
color: white;  
height: 60px;  
width: 100%;  
}
```

```
table {  
width: 100%;  
text-align: center;  
}
```

```
th, td {  
padding: 4px;  
}
```

```
th {  
font-weight: bold;  
font-style: italic;  
}
```

```
p {  
animation-duration: 3s;  
animation-name: slidein;  
}
```

```
@keyframes slidein {  
from {  
margin-left: 100%;  
width: 300%;  
}
```

```
to {  
margin-left: 0%;  
width: 100%;  
}  
}
```

```
.menuButton {  
position: relative;  
transition-property: background-color, color;  
transition-duration: 1s;  
transition-timing-function: ease-out;  
text-align: left;  
background-color: grey;
```



```

left: 5px;
top: 5px;
height: 26px;
color: white;
border-color: black;
font-family: sans-serif;
font-size: 20px;
text-decoration: none;
box-shadow: 2px 2px 1px black;
padding: 2px 4px;
border: solid 1px black;
}

.menuButton:hover {
position: relative;
transition-property: background-color, color;
transition-duration: 1s;
transition-timing-function: ease-out;
background-color: white;
color: black;
box-shadow: 2px 2px 1px black;
}

```

Файл html:

```

<!DOCTYPE html>
<head>
  <title>Практическая работа №3</title>
  <link rel="stylesheet" type="text/css" href="name_base.css">
</head>
<body>
  <header>
    Золотая шиншилла
  </header>
  <nav>
    <table>
      <tr>
        <th>
          <td>Главная</td>
          <td>О породе</td>
          <td>Особенности ухода</td>
          <td>Питомники</td>
          <td>Справка</td>
        </th>
      </tr>
    </table>
  </nav>
  <aside>
    <h3 class="menuButton">Содержание</h3><br>
    Особенности характера<br>
    Как выбрать?<br>
    Возможные болезни<br>
    Разведение<br>
    Отзывы владельцев<br>
  </aside>
  <div id="content">
    <iframe
      src="https://c-cats.ru/info-british-chinchilla-description/"
      width="1270px" >
    </iframe>
  </div>

```

```
<div id="clear"></div>
<footer>
  <p>Практическую работу выполнил студент группы ИКПИ-__ ФИО</p>
</footer>
</body>
</html>
```

Итог:

По итогу выполнения данной работы мы рассмотрели различные стили CSS, а также блочную верстку. По окончании работы ответьте на контрольные вопросы для закрепления знаний.

Отчет по практическому заданию должен содержать описание ваших действий при выполнении практического задания с приложенным итоговым скриншотом экрана и фрагментов кода.

Ответить на контрольные вопросы

1. Анимация в CSS.
2. Градиент в CSS.
3. Подсвечивание в CSS.
4. Фреймы.
5. Тег div.

ЛАБОРАТОРНАЯ РАБОТА №2.

Создание сайта средствами HTML и CSS

Вам необходимо сверстать макет страницы используя HTML и CSS. Тему использовать лабораторной работы №1. Стили необходимо присоединить с помощью внешнего файла style.css. Макет должен содержать шапку сайта, сайдбар (левый или правый, допускается применение обоих), основная часть страницы, подвал.

Применить фреймы, анимацию, градиент, подсвечивание.

Итог:

По итогу выполнения данной работы мы рассмотрели различные стили CSS, а также блочную верстку. По окончании работы ответьте на контрольные вопросы для закрепления знаний.

Отчет по практическому заданию должен содержать описание ваших действий при выполнении практического задания с приложенным итоговым скриншотом экрана и фрагментов кода.

ПРАКТИЧЕСКАЯ РАБОТА №4.

Основы JavaScript

Цель работы: Изучить основы JavaScript.

Подключение JavaScript может быть реализовано несколькими способами. И как всегда каждый из них имеет свои плюсы и свои минусы.

Тег <script>

JavaScript программы могут быть встроены в любое место документа HTML. Для этого используется парный тег <script></script>

Код внутри данного тега выполнится в тот момент, когда браузер его прогрузит и обработает.

Пример:

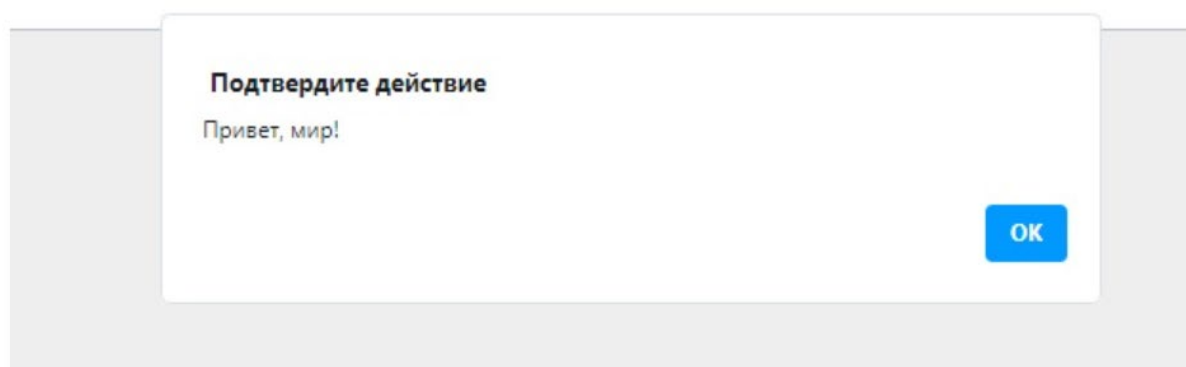
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Мой первый сайт</title>
  </head>
  <body>

    <H1 align="center"> Привет МИР!!!</H1>

    <script>
      alert( 'Привет, мир!' );
    </script>

    <p>Дорогой друг я рад видеть тебя на моем первом сайте!</p>
  </body>
</html>
```

Код из примера при загрузке HTML документа создаст всплывающее окно ALERT.



Подключение внешнего скрипта

В большинстве случаев скрипты JavaScript подключаются с помощью внешнего файла. Подключение внешних скриптов очень напоминает подключение стилей CSS к документу HTML. Данный способ используется если у вас много JavaScript кода и, если он должен выполняться не только на одной странице, а на нескольких. Файл скрипта подключается с помощью тега `<script>` и его атрибута `src`, который содержит путь к файлу JS.

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

```
<script src="site.com/mood.js"></script>
```

Атрибут `src` может содержать не только локальный путь к файлу скриптов, но и путь к файлу скриптов на стороннем ресурсе. Это помогает уменьшить место, занимаемое сайтом на сервере или хостинге. Но немного может увеличиваться время загрузки сайта, так как ему придется подгружать файл скриптов с другого сайта.

В качестве достоинства можно выделить то что большинство браузеров отдельно подгружают внешний файл со скриптами JS и может хранить его в своем кэше. А другие страницы, использующие этот же файл, будут использовать тот который уже сохранен в кэше. Это сокращает расход трафика на сервере и позволяет ускорить загрузку страниц.

УПРАЖНЕНИЕ 1. Alert и формы.

Создайте HTML-документ, в нем форму, сообщающую информацию о студенте. Форма будет содержать кнопку. Нажатие на кнопку вызывает появление окна с сообщением о студенте «ФИО Группа ИКПИ__ Практическая №__». Использовать функцию, написанную на языке JavaScript, открывающую окно с сообщением. Добавьте стили в отдельный файл css для форматирования кнопки и ее расположения на странице.

Файл: js

```
function show_information()
{ alert("ФИО Группа ИКПИ__ Практическая №__"); }
```

Файл html:

```
<form name="button1" class="all_center">
    <input type="button" value="ИНФО" onclick="show_information();">
</form>
```

УПРАЖНЕНИЕ 2. Alert и формы (ввод данных).

В том же документе html, создайте еще одну форму, в которую включим следующие элементы: окно ввода с поясняющим текстом и кнопка для вывода среднего балла. Окно ввода служит для ввода среднего балла, окно сообщения будет выводить это значение. Если окно ввода пустое или данные не попадают в диапазон от 3.00 до 5.00, нажатие на кнопку должно вызывать сообщение «Нет данных!». Добавьте стили в отдельный файл css для форматирования объектов и их расположения на странице.

Файл: js

```
function show_information2(elem)
{ let x = parseInt(elem.value);
  if (!isNaN(x) && 3.00 <= x && x <= 5.00) {
    alert("Ваш средний балл – " + elem.value + " балла"); }
  else { alert("Нет данных!"); }
```

Файл html:

```
<form name="button2" class="all_center">
```

```

        <input type="number" min="3.00" max="5.00" style="width: 150px;" name="sqr"
placeholder="Средний балл">

        <input type="button" value="Показать средний балл"
onclick="show_information2(button2.sqr);">

    </form>

```

УПРАЖНЕНИЕ 3. Анкета.

В том же документе html, создайте еще одну форму, представляющую собой анкету/опрос. Анкета будет представлять собой форму и содержать средство для ввода фамилии, три вопроса, ответы на которые выбираются с помощью радиокнопок (не менее трёх в каждой группе), и кнопки, «обрабатывающей» анкету. Обработка анкеты заключается в определении «рейтинга», представляющего собой средний балл за ответы на вопросы. Добавьте стили в отдельный файл css для форматирования анкеты и ее расположения на странице.

Файл: js

```

function send(family, radio1, radio2, radio3)

{
    let markProg = parseInt(radio1.value), markMath = parseInt(radio2.value); markPhyz =
parseInt(radio3.value);

    document.write("<h2>Фамилия: " + family.value + "</h2>");

    document.write("<h2>Оценка по дисциплине «Программирование»: " + markProg +
"</h2>");

    document.write("<h2>Оценка по дисциплине «Математика»: " + markMath + "</h2>");

    document.write("<h2>Оценка по дисциплине «Физика»: " + markPhyz + "</h2>");

    document.write("<h2>Рейтинг: " + ((markProg + markMath + markPhyz) / 3) + "</h2>");

}

```

Файл html:

```

<h1 class="radio_button">Опрос</h1>

    <form name="questionnaire" class="radio_button">

        <input type="text" name="family" placeholder="Фамилия"><br>

        <label> Оценка по дисциплине «Программирование»:</label><br>

        <input type="radio" name="radio1" value="5"
checked="checked"><label>Отлично</label><br>

        <input type="radio" name="radio1" value="4"><label>Хорошо</label><br>

        <input type="radio" name="radio1" value="3"><label>Удовлетворительно</label><br>

        <label>Оценка по дисциплине «Математика»:</label><br>

```

```

        <input                type="radio"                name="radio2"                value="5"
checked="checked"><label>Отлично</label><br>

        <input type="radio" name="radio2" value="4"><label>Хорошо</label><br>

        <input                type="radio"                name="radio2"
value="3"><label>Удовлетворительно</label><br>

        <label>Оценка по дисциплине «Физика»:</label><br>

        <input                type="radio"                name="radio3"                value="5"
checked="checked"><label>Отлично</label><br>

        <input type="radio" name="radio3" value="4"><label>Хорошо</label><br>

        <input                type="radio"                name="radio3"
value="3"><label>Удовлетворительно</label><br>

        <input type="button" value="Отправить" onclick="send(questionnaire.family,
questionnaire.radio1, questionnaire.radio2, questionnaire.radio3);">

    </form>

```

Итог:

По итогу выполнения данной работы мы рассмотрели основы JavaScript. По окончании работы ответьте на контрольные вопросы для закрепления знаний.

Отчет по практическому заданию должен содержать описание ваших действий при выполнении практического задания с приложенным итоговым скриншотом экрана и фрагментов кода.

Ответить на контрольные вопросы

1. Объявление переменных в JS.
2. Условные операторы в JS.
3. Функции и методы в JS.

ЛАБОРАТОРНАЯ РАБОТА №3. Формы и простейшие Java-скрипты в HTML-документах

Цель работы: Изучить основы написания простых Java-скриптов.

В лабораторной работе №3 разработать форму, которая проверяет, правильно ли введен адрес электронной почты. При этом выводится сообщение об ошибке или наоборот, о правильности ввода данных. Разместить эту форму в слое, фон которого при двойном щелчке будет меняться на любой по выбору и наоборот.

Примеры неверно введенных почтовых адресов:

Введите адрес электронной почты и нажмите кнопку «Проверить»

123 Адрес неверный

Проверить

Введите адрес электронной почты и нажмите кнопку «Проверить»

ПрИвЕт@ Адрес неверный

Проверить

Введите адрес электронной почты и нажмите кнопку «Проверить»

my_mail@qq,! Адрес неверный

Проверить

Пример верно введенного почтового адреса:

Введите адрес электронной почты и нажмите кнопку «Проверить»

my_mail@mail.com Адрес правильный

Проверить

Пример изменения фона по двойному щелчку мышки:

Введите адрес электронной почты и нажмите кнопку «Проверить»

my_mail@mail.com Адрес правильный

Проверить

Дважды щелкните в этом поле для изменения цвета фона.

Введите адрес электронной почты и нажмите кнопку «Проверить»

my_mail@mail.com

Адрес правильный

Проверить

Дважды щелкните в этом поле для изменения цвета фона.

Итог:

По итогу выполнения данной работы мы рассмотрели основы JavaScript. По окончании работы ответьте на контрольные вопросы для закрепления знаний.

Отчет по практическому заданию должен содержать описание ваших действий при выполнении практического задания с приложенным итоговым скриншотом экрана и фрагментов кода.

ПРАКТИЧЕСКАЯ РАБОТА №5. Изучение языка php

Цель работы: Изучить основы языка серверного скриптования PHP.

Общие правила построения php-программы

PHP-программа должна быть выделена в HTML-тексте специальными тегами. Существует четыре стиля оформления PHP-кода:

Стиль	Открывающий тег	Закрывающий тег
Сокращенный	<?	?>
XML (стандартный)	<?php	?>
ASP	<%	%>
SCRIPT (программный)	<SCRIPT LANGUAGE="php">	</SCRIPT>

Из перечисленных тегов только стандартные и программные работают в любой конфигурации PHP.

УПРАЖНЕНИЕ 1. PHP-оператор echo

<html>

<head>


```
<title>Документ, состоящий из HTML-текста и PHP-кода</title>
```

```
</head>
```

```
<body>
```

```
<?php
```

```
echo "А вот и PHP!<br>";
```

```
?>
```

Всё очень просто!

```
</body>
```

```
</html>
```

Здесь содержится один PHP-оператор **echo**. Этот оператор работает на стороне сервера и встраивает в HTML строку-аргумент "**А вот и PHP!
**" в HTML-страницу. При этом тег **
** сделает свое дело, т.е. осуществит переход на новую строку.

Сервер "узнает" о том, что это исполняемая программа по специальным тегам:

```
<? ... ?>
```

Отличие языка PHP от языков клиентского программирования (таких, как JavaScript), состоит в том, что программный код выполняется на стороне сервера, т.е. еще до того, как страницу получит браузер пользователя.

В одном документе можно чередовать HTML-текст и блоки PHP-операторов сколько угодно раз. При этом все переменные, функции и классы, определенные в первом блоке, будут доступны и в последующих блоках.

В примере заменим вызов функции **echo** на вызов встроенной функции **phpinfo()**, которая выдаст список параметров PHP-среды.

УПРАЖНЕНИЕ 2. Встроенная функция `phpinfo()`

```
<html>
```

```
<head>
```

```
<title>Информационная страница PHP</title>
```

```
</head>
```

```
<body>
```

```
<?php
```

```
phpinfo();
```

```
?>
```

```
</body>
```

```
</html>
```

Выданная справка содержит большое количество разнообразной информации о среде PHP (обратите внимание на бегунок справа). Функция **phpinfo()** бывает весьма полезна при диагностировании поведения создаваемых PHP-программ.

Все страницы, содержащие программный код, должны иметь расширение *.phtml* или *.php*, однако, если вы имеете доступ к администрированию сервера, вы можете настроить препроцессор PHP таким образом, чтобы на содержание скрипта проверялись все страницы, включая *.html*.

УПРАЖНЕНИЕ 3. Вывод на экран и переменные в PHP.

Создайте файл и поместите в начало страницы следующий код:

```
<html>
```

```
<head>
```

```
<title>Вывод на экран и переменные в PHP</title>
```

```
<?php
```

```
echo "Привет, мир!";
```

```
?>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

УПРАЖНЕНИЕ 4. Введите в скрипт переменные каждого типа:

```
<html>
```

```
<head>
```

```
<title>Вывод на экран и переменные в PHP</title>
```

```
</head>
```

```
<body>
```

```
<?php

$i = 6; // целое

$d = 4.89; // дробное

$str = "PHP для всех!"; // строка

echo ($i + $d);

echo "<br>Привет, мир! ".$str;

?>

</body>

</html>
```

Обратите внимание, что все переменные в PHP должны начинаться с символа dollar (\$), тип переменной не задается в явном виде, он вычисляется препроцессором PHP в зависимости от контекста. В результате работы скрипта на экран первой строкой будет выведена сумма значений двух переменных **\$i** и **\$d** и результат конкатенации двух строк **"
Привет, мир!"** и значения переменной **\$str**.

Поддерживаются все арифметические операции и функции, многоуровневые скобки, логические операции, операции увеличения или уменьшения на единицу и многое другое. Кроме того, очень просто и естественно организуется сравнение **if-else**.

УПРАЖНЕНИЕ 5. Логический оператор if-else

```
<html>

<head>

<title>Вывод на экран и переменные в PHP</title>

</head>

<body>

<?php

$i = 6; // целое

$v = 7;

$d = 4.89; // дробное

$str = "PHP для всех!"; // строка

echo ($i + $d);
```

```

echo "<br>Привет, мир!".$str;

if($i == $v)

{

echo $i + $v;

}

else

{

echo $i.$v;

}

?>

</body>

</html>

```

При сравнении на истину применяется два знака равенства для того, чтобы интерпретатор мог без труда отличить сравнение от присваивания. Результат работы скрипта — **67**, т.к **\$i** не равно **\$v**, а команда **echo \$i.\$v**; объединяет две строки: **"6"** и **"7"**. Неравенство обозначается символами **!=**, допустимы все остальные арифметические и логические символы и операторы (например, **or**, **and**, **>**, **<=** и т.д.).

В PHP есть средства быстрого изменения переменной на единицу в сторону увеличения или уменьшения. Для этого нужно указать имя переменной и за ней, без знака равенства, — подряд два плюса или минуса соответственно. Например, **\$a++**; — переменная **\$a** будет увеличена на единицу.

Поддерживается одновременное присваивание одного значения нескольким переменным — **\$a = \$b = 4;**. Введите на странице следующий код:

УПРАЖНЕНИЕ 6. Быстрое изменение переменной на единицу.

```

<html>

<head>

<title>Вывод на экран и переменные в PHP</title>

</head>

<body>

<?php

```

```
$b = $a = 5;

echo "<br>переменная a=$a, b=$b";

$c = $a++;

echo "<br>переменная a=$a, c=$c";

$d = $d = ++$b;

echo "<br>переменная e=$e, d=$d, b=$b";

$g = 10;

$h = $g += 10;

echo "<br>переменная g=$g, h=$h";

?>

</body>

</html>
```

Обратите внимание, что операция присваивания тоже дает свой результат, так операция **\$a = 5** дала результат 5, именно поэтому переменная **b** была проинициализирована в 5.

В операции **\$c = \$a++**; сначала было произведено присваивание, а затем переменная **\$a** была увеличена на 1.

В операции **\$d = ++\$b**; сначала была увеличена на 1 переменная **\$b**, а уже затем присвоено получившееся значение.

Итог:

По итогу выполнения данной работы мы рассмотрели основы PHP. По окончании работы ответьте на контрольные вопросы для закрепления знаний.

Отчет по практическому заданию должен содержать описание ваших действий при выполнении практического задания с приложенным итоговым скриншотом экрана и фрагментов кода.

Ответить на контрольные вопросы

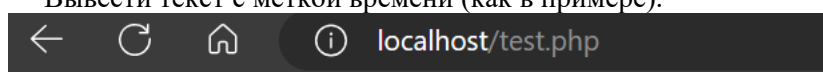
1. Переменные в PHP.
2. Операторы в PHP.

ЛАБОРАТОРНАЯ РАБОТА №4.

Знакомство с Web-сервером и обработкой данных с помощью PHP-программ

Цель работы: познакомиться с настройками Web-сервера, с простейшими инструкциями языка PHP, использованием переменных, методами пересылки данных между клиентом и сервером, научиться использовать PHP-скрипты в HTML-документах.

1. Настроить Web-сервер для работы с PHP-программами.
2. Познакомиться с информацией о Web-сервере, появившейся на экране. Определить операционную систему, установленную на сервере, название и версию Web-сервера, версию PHP, IP-адреса сервера, удаленного компьютера, адрес каталога для хранения документов.
3. Вывести текст с меткой времени (как в примере).



4. Создать файл, выполняющий сложение двух числовых переменных. Исходные данные для вычисления передать как параметры при вызове php-файла: localhost/test.php?var1=3&var2=5

5. Создать файл, отображающий в окне браузера параметры Web-сервера. Варианты задания:

1. IP-адрес удаленного компьютера, метод пересылки данных;
2. Программное обеспечение сервера, протокол передачи данных;
3. Каталог для хранения документов на сервере, IP-адрес сервера;
4. Почтовый адрес администратора сети, имя хост-компьютера.

6. Создать Web-страницу с формой HTML, которая содержит поля ввода и кнопку подтверждения. При нажатии кнопки подтверждения форма отправляет данные на сервер для обработки. Обработку данных выполняет php-сценарий. Результат выводится на отдельной странице вместе с исходными данными.

Задание: Фамилия, экзаменационные оценки по трем предметам – вычислить средний балл и назначить на стипендию.

7. Создать Web-страницу с формой HTML, содержащей поле ввода и кнопку подтверждения. В поле ввода вводится строка, состоящая из нескольких слов. Сценарий php обрабатывает строку в соответствии с заданием. Варианты задания:

Задание: Вычислить количество слов в строке.

Итог:

По итогу выполнения данной работы мы рассмотрели обработку данных с помощью PHP-программ. По окончании работы ответьте на контрольные вопросы для закрепления знаний.

Отчет по практическому заданию должен содержать описание ваших действий при выполнении практического задания с приложенным итоговым скриншотом экрана и фрагментов кода.

Ответить на контрольные вопросы

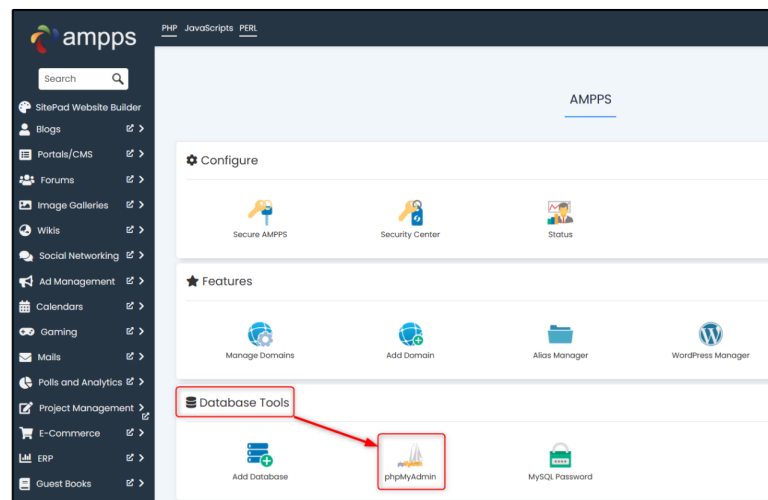
1. Переменные в PHP.
2. Операторы в PHP.

ПРАКТИЧЕСКАЯ РАБОТА №6. Запросы

Цель работы: получить представление о работе запросов.

УПРАЖНЕНИЕ 1.

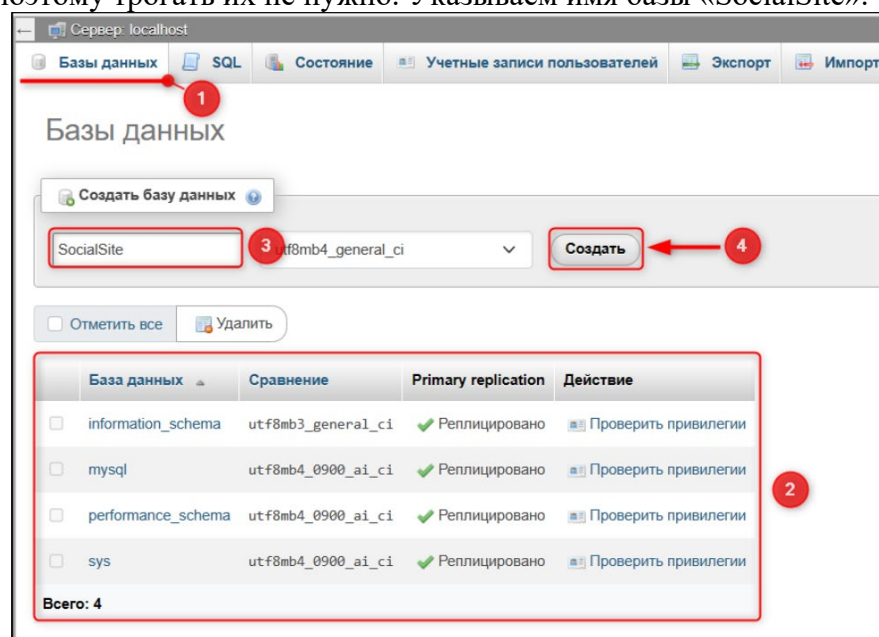
Ознакомимся с MySQL и посмотрим на базу данных phpMyAdmin. Перейдем в пункт управления. Нас переносит на домашнюю страницу Apache AMPPS, где находится весь пункт управления. Находим раздел «Database Tools»: внутри него лежит нужная ссылка «phpMyAdmin», нажимаем на нее.



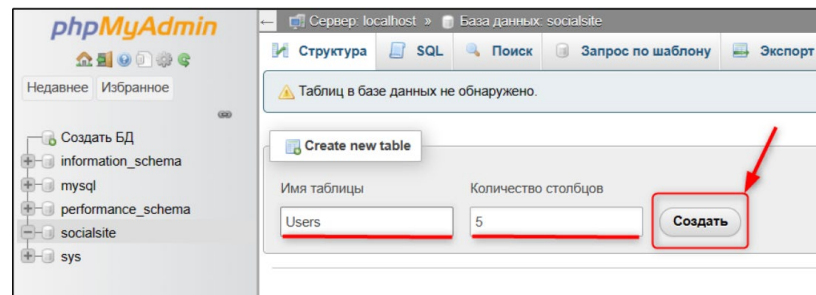
Запускается приветственное окно, в котором потребуется ввести пользователя (root) и пароль (mysql). Попадаем на главную страницу разработки.

УПРАЖНЕНИЕ 2. Создание базы данных для сайта.

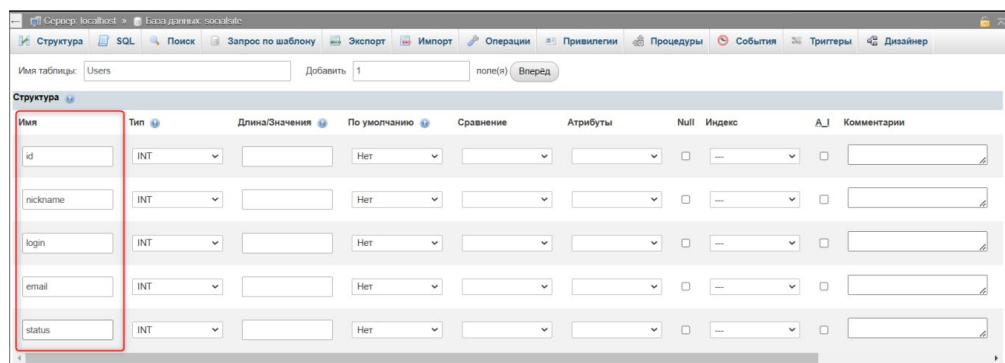
Для ее создания перейдем во вкладку «Базы данных» в верхней панели сайта, откроется ее настройка. Сразу можно заметить, что создано 4 базы данных, они являются стандартными, поэтому трогать их не нужно. Указываем имя базы «SocialSite».



После создания база данных появится в левом трейе, можно увидеть, что ее структура пустая. Создадим тестовую таблицу, чтобы посмотреть, как все работает. Указываем имя таблицы «Users» и количество столбцов – 5. Создалась структура таблицы, которую необходимо заполнить.

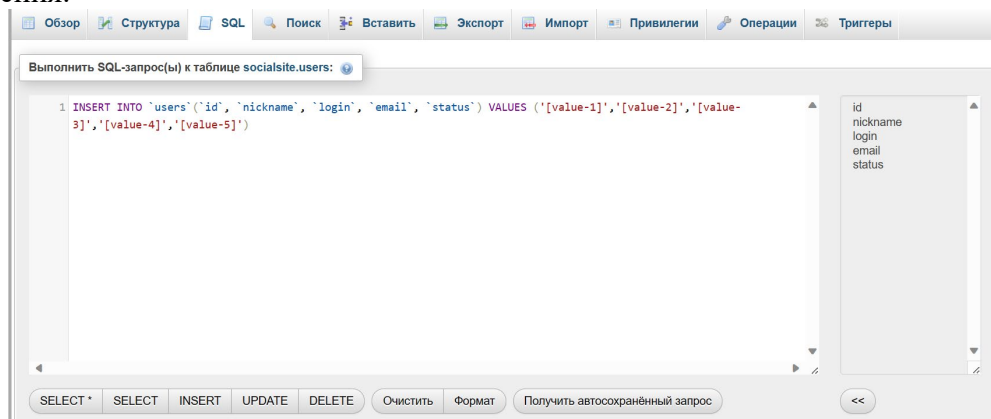


В столбце «Имя» заполним все пустые поля: id (ключ), nickname, login, email и status. Структура может изменяться в зависимости от логики работы, поэтому на данный момент она является тестовой, далее мы будем ее изменять под наш сайт.

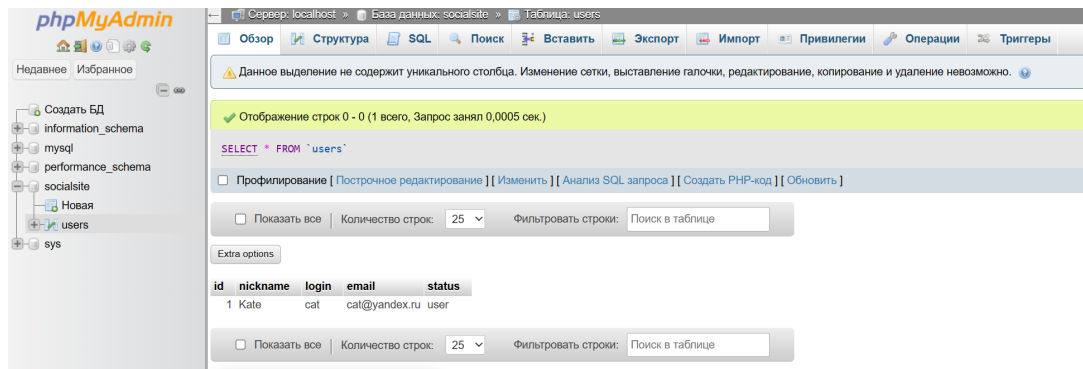


УПРАЖНЕНИЕ 3. Написание запроса.

Рассмотрим создание запроса на localhost. Для этого перейдем на вкладку «SQL», на экране отобразится уже созданный стандартный запрос. Под окном запроса находятся команды с шаблонами, которые можно применить и посмотреть, как они будут выглядеть. Введем значения.



Результат:



Добавьте еще несколько строк в таблицу, проверьте результат.

Итог:

По итогу выполнения данной работы мы рассмотрели запросы SQL. По окончании работы ответьте на контрольные вопросы для закрепления знаний.

Отчет по практическому заданию должен содержать описание ваших действий при выполнении практического задания с приложенным итоговым скриншотом экрана и фрагментов кода.

Ответить на контрольные вопросы

1. Как обозначаются комментарии на PHP?
2. Что такое select *?

ЛАБОРАТОРНАЯ РАБОТА №5. Запросы SQL через PHP

Цель работы: Написание запросов через PHP.

1. Подключиться к базе данных через users PHP, созданной в практической работе №6, вывести id, никнейм, логин.

Результат:

Номер пользователя 1

Имя: Kate

2. Добавьте еще несколько строк в базу данных, проверьте результат.

Результат:

Номер пользователя 1

Имя: Kate

Логин: cat

Номер пользователя 2

Имя: Bob

Логин: Bobby123

3. Создать базу данных users_1lab, подключиться к базе данных через PHP. Через форму, созданной в практической работе №1, записать данные в базу данных users_1lab.

Результат:

login	password	country	mail	age	hobby
Clara	12345678	Мурманск	876clar@yandex.ru	adult	music
Bob	BobBoBy	Москва	bob.b.b@mail.ru	junior	computers
Olga	QwEr567	Другой...	popolg@mail.ru	child	sport

4. Добавить хэширование паролей.

Результат:

login	password	country	mail	age	hobby
Clara	12345678	Мурманск	876clar@yandex.ru	adult	music
Bob	BobBoBy	Москва	bob.b.b@mail.ru	junior	computers
Olga	QwEr567	Другой...	popolg@mail.ru	child	sport
Jack	1ece33f8258a7ea8d35b8ffaef329859	Санкт-Петербург	jjaaaa@mail.ru	adult	sport

5. Добавить проверку на длину логина и пароля:

Логин должен состоять не менее чем из 3 символов и не более чем из 15 символов.

Пароль должен состоять не менее чем из 10 символов и не более чем из 15 символов.

Итог:

По итогу выполнения данной работы мы рассмотрели запросы SQL через PHP. По окончании работы ответьте на контрольные вопросы для закрепления знаний.

Отчет по практическому заданию должен содержать описание ваших действий при выполнении практического задания с приложенным итоговым скриншотом экрана и фрагментов кода.