МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФГБОУ ВО «Брянский государственный технический университет»
(БГТУ)

Политехнический колледж (ПК БГТУ)

УТВЕРЖДАЮ
Директор ПК БГТУ

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 3E0E6C00DDAD69AC496A79E33A58FAAB
Владелец: Федонин Олег Николаевич
Действителен: с 11.11.2021 до 11.02.2023

_____ В.М. Малашенко

« » 2019 г.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по выполнению практических работ
по учебной дисциплине
ОП.11 WEB Дизайн

Специальность:	**09.02.03 Программирование в компьютерных системах**
Уровень образования выпускника:	среднее профессиональное образование (СПО)
Программа подготовки специалиста среднего звена (ППССЗ):	базовая
Присваиваемая квалификация:	Техник-программист
Форма обучения:	очная
Срок получения СПО по ППССЗ:	3 года 10 месяцев
Уровень образования, необходимый для приема на обучение по ППССЗ:	основное общее образование
Год приема на обучение на 1-й курс:	2019

Брянск 2019

Методические указания по выполнению практических работ

по учебной дисциплине

ОП.11 WEB-дизайн

(далее — МУ)

для специальности ***09.02.03 Программирование в компьютерных системах***

Разработал(и):

– преподаватель ПК БГТУ Е.И.Пунин

МУ рассмотрены и одобрены на заседании предметно-цикловой комиссии «Программирование в компьютерных системах» ПК БГТУ (далее — ПЦК)

от « » 2019 г., протокол №

Председатель ПЦК Е.С. Трошина

Согласовано:

Заместитель директора ПК БГТУ
по учебно-методической работе Т.Е.Балашова

Введение

Главной задачей среднего профессионального образования является подготовка компетентных специалистов. В процессе формирования профессиональных и общих компетенций практические занятия занимают промежуточное положение между теоретическим и производственным обучением и служат одним из важнейших средств осуществления связи теории и практики.

Практические занятия являются неотъемлемым этапом изучения по учебной дисциплине ОП.15. «Web дизайн» и проводятся с целью:

- формирования практических умений в соответствии с требованиями к уровню подготовки обучающихся, установленными рабочей программой;
- обобщения, систематизации, углубления, закрепления полученных теоретических знаний;
- готовности использовать теоретические знания на практике.

Выполнение практических работ призвано способствовать закреплению теоретических знаний, формированию умений и способов действий через самостоятельную деятельность студентов. Ведущей дидактической целью практических занятий является формирование практических (профессиональных) умений – выполнение определённых действий, операций, необходимых в последующей профессиональной деятельности. Основная задача практических работ - научить студентов применять теоретические знания в практических ситуациях.

Выполнению практических заданий на уроке предшествует проверка знаний студентов, их теоретической готовности к выполнению практической работы.

Структура и содержание практических работ включает в себя следующие элементы: тема, цель выполнения работы, оборудование, программное обеспечение, методические указания по выполнению работы, контрольные вопросы. По каждой работе необходимо оформить отчет в соответствии с требованиями, сделать выводы, ответить на контрольные вопросы. Отчет о выполненной работе представляется студентом преподавателю для проверки, в

том числе с защитой результатов, и оценивания. Отчет может быть представлен как в письменном, так и печатном виде.

Практические занятия по учебной дисциплине ОП.15. «Web дизайн» способствуют формированию следующих общих и профессиональных компетенций:

ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.

ОК 2. Организовывать собственную деятельность, выбирать типовые методы решения профессиональных задач, оценивать их эффективность и качество.

ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.

ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.

ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.

ОК 6. Работать в коллективе и команде, эффективно общаться с коллегами, руководством, людьми, находящимися в зонах пожара.

ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), результат выполнения заданий.

ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.

ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности.

В результате освоения учебной дисциплины обучающийся должен знать:

- основные понятия HTML - кода;
- существующие способы построения Интернет страниц;

- основных средства создания и редактирования Интернет страниц с помощью средств операционной системы или специализированных программ.

ПК 1.1. Выполнять разработку спецификаций отдельных компонент.

ПК 1.2. Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.

ПК 1.6. Разрабатывать компоненты проектной и технической документации с использованием графических языков спецификаций.

Для успешного выполнения практических работ обучающиеся обязаны ознакомиться с порядком их проведения и изучить соответствующие разделы теоретического курса.

Обучающиеся должны четко представлять задачу, уметь проводить необходимые расчеты.

Общие указания к выполнению работ

Перед началом выполнения работы внимательно ознакомьтесь с инструкцией, заданием к практической работе.

Отчет оформляется на листах со штампом. В отчет впишите тему, цель работы, оборудование, программное обеспечение. При выполнении работы следуйте приведенным пунктам плана. Номер пунктов и их названия должны соответствовать друг другу.

По мере выполнения работы необходимо сформулировать вывод.

Отчеты оформляются в журнале. На титульном листе должны быть указаны: название предмета, группа и фамилия студента. Титульный лист оформляется на формате А4

Пример оформления титульного листа:

Министерство науки и высшего образования РФ

ФГБОУ ВО «Брянский государственный технический университет»

«Политехнический колледж»

ЖУРНАЛ

практических работ по учебной дисциплине

Web - дизайн

Студент_____

Группа:_____

Вариант №_____

Преподаватель_____

Практическая работа № 1
Разработка проекта сайта

Цель работы: изучение и освоение принципов разработки сайта.

Оборудование: IBM PC

Программное обеспечение: Microsoft Windows

Указания к выполнению работы

1. Определение основной темы разрабатываемого сайта.

После того, как будет определена основная тема разрабатываемого сайта, необходимо сформулировать, что должно присутствовать на сайте, представляющем компанию или проект.

Сформулировать информационную архитектуру сайта. Любая информационная система – будь то книга или интранет-сеть предприятия – имеют свою информационную архитектуру, которая планируется с самого начала. Существует достаточно много способов организовывать информацию в систему. Хорошо продуманная архитектура сайта помогает пользователям тем, что использует некоторый набор из всех этих способов, т. е. набор, который больше всего подходит для сайта, исходя из целей сайта, информационных потребностей посетителей сайта, материалов сайта, бюджетных ограничений.

Продуманная информационная архитектура гарантирует, что пользователи потратят меньше времени на поиск нужной информации, и почти никогда не скажут, что они чего-то не нашли. При хорошей архитектуре они всегда обнаружат, что один документ связан ссылками с другими документами по той же теме. Они всегда смогут легко переключаться с поиска документов на их просмотр и обратно. Они лучше будут понимать, какую информацию сайт может им предложить.

2. Разработка содержания, структуры и дизайна сайта.

После того, как вы определились с целями и задачами сайта, следует приступить к *разработке его содержания.* На этом этапе необходимо составить перечень информационных разделов сайта и по каждому разделу установить состав входящей в него информации и сформулировать содержание массивов информации («контент»). Содержание сайта – основа любого сайта, и от этого, во многом зависит его успех. Так что же должно размещаться на сайте, какая информация, и главное – как, чтобы посетители, раз зашедшие к вам на сайт, заходили на него снова и снова?

Далее необходимо *разработать структуру сайта* – один из важнейших, после *цели*, вопросов создания сайта. Дизайн, стилистика, компо-

новка, программные средства, технологии и прочие разрабатываются и/или выбираются исходя из структуры – можно даже сказать, что они являются структурными элементами сайта. Именно выбор правильной структуры обеспечивает «высокое юзабилити» сайта (удобство пользования) и гарантирует доступ ко всей информации – быстроту и легкость нахождения, адекватность материалов, находящихся в разделах, их тематике.

Структурность – многоплановое понятие, которое применяется как к *информации*, так и к визуальным *элементам.*

Нельзя назвать верным такое решение, когда все содержательное наполнение сайта, традиционно называемое «информация», или «контент», расположено, что называется, «внавалку». Как правило, вся информация рассортировывается, в первую очередь, по тематическим разделам.

Например, для *web*-представительства небольшой фирмы информационная структура может выглядеть так:

1. Index (Главная страница – это страница, которая загружается при наборе доменного имени сайта в строке браузера. В подавляющем боль-шинстве сайтов главная страница содержит колонку News);
2. About (О нас, О компании и т. д.);
3. News (Новости – периодически и регулярно дополняемая информа-
4. ция);
5. Products (Наши услуги, Наши продукты, Наши туры);
6. Price (Цены, прайс-лист);
7. Support (Техническая и/или сервисная поддержка, горячая линия и/или e-mail связь);
8. Search (Поиск – модуль поиска по введенному слову или словосочетанию).

Расположение информации в самих разделах тоже должно придерживаться какого-либо принципа: хронологический, алфавитный, по ID, по тематической важности и т. п.

Структура сайта – это не только то, что мы видим, за этим стоит программный продукт с разделением содержания и представления(рис. 1.1).

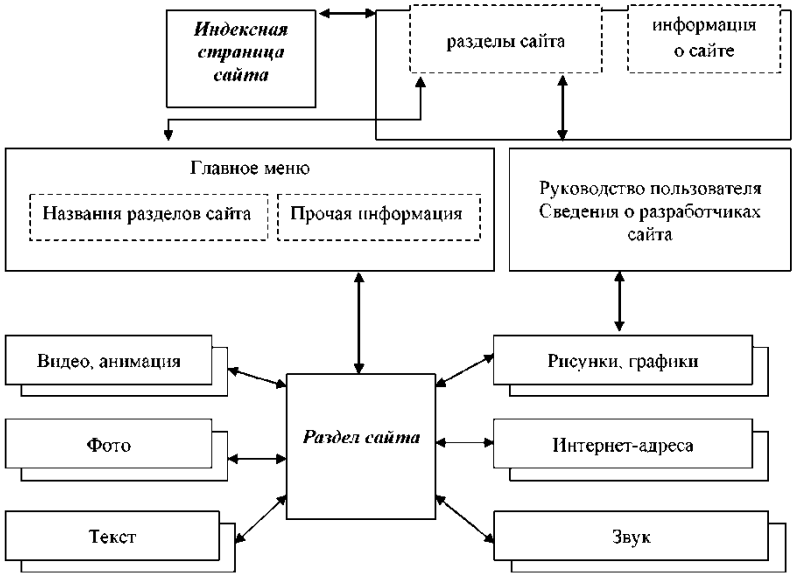


Рис. 1.1. Пример общей структуры сайта

Если рассматривать *web*-страницу не с визуальной стороны, а с точки зрения структурного подхода, то можно выделить несколько групп со-ставляющих ее элементов, таких, как:

1. *текст* – основное содержание страницы, так называемый контент;
2. *акцентуированный текст* – специальным образом форматированный текст, чаще всего с использованием CSS (каскадная таблица стилей), использующийся и для оформления контента, и как элемент интерфейса сайта – меню, разделы, заголовки;
3. *фон* – цвет страницы или ячейки таблицы;
4. *ссылки* – ссылки гипертекста;
5. *картинки* – фотографии и графика контента, графические элементы оформления сайта.

После этого можно приступить непосредственно *к разработке дизайна* проектируемого сайта,

Подготовка технического задания (сценария) для создания сайта.

Сайт с информацией создается на *web*-страницах с помощью языка гипертекстовых разметок и программы, которые сопровождаются и управ-

ляются *техническим заданием* (*сценарием*), обеспечивающих активность гипертекстовых страниц (представляет методы обработки данных и компоновку страниц). При создании сайта техническое задание (сценарий) является основным документом, который конкретизирует общие положения и требования к сайту.

Разработка макета сайта.

Выбор варианта компоновки сайта (или используемых шаблонов сайтов) определяется условиями технического задания на разрабатываемый *web*-сайт. Существуют сайты с фиксированной шириной или «резиновые» сайты, изменяющие ширину в зависимости от разрешения монитора пользователя (рис. 1.2).

Рис. 1.2. Сайт с фиксируемой шириной

Верхняя часть страницы («шапка», header) используется, как правило, для размещения логотипа, эмблемы, названия фирмы и другой важной информации (рис. 1.3).

Нижняя часть страницы («подвал», footer) используется для контактной информации или данных, структурно требующих разделения с данными, размещенными в верхней части сайта (рис. 1.4).

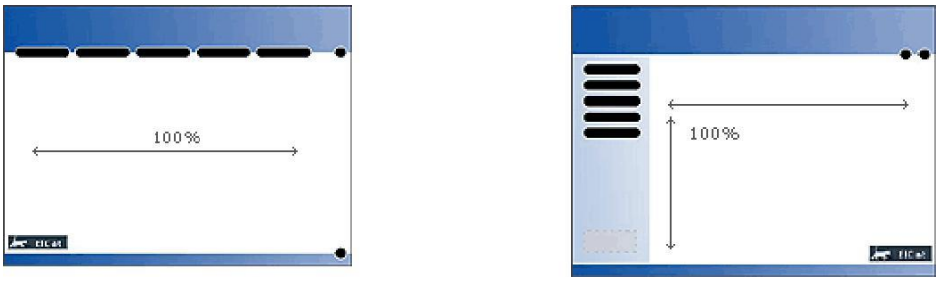


Рис. 1.3. Верхняя часть страницы

Рис. 1.4. Нижняя часть страницы

Панель навигации (главное меню) сайта должна размещаться в удоб-ном для частого использования месте, так как количество ссылок может повлиять на компоновку сайта (вертикальное или горизонтальное размещение панели навигации) (рис. 1.5).

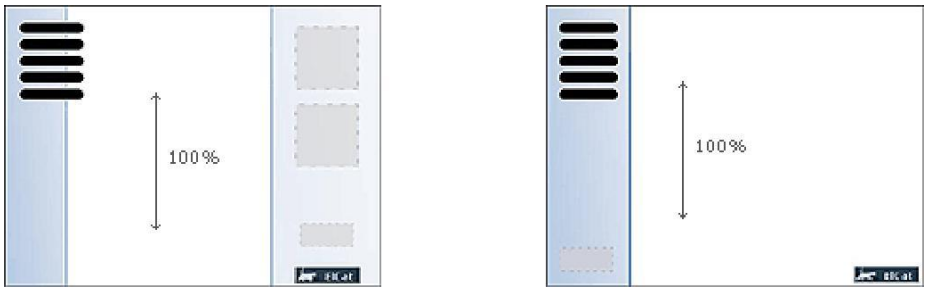


Рис. 1.5. Главное меню страницы

Сбор и подготовка информации и материалов для сайта.

Подготовка материалов и дальнейшее уточнение структуры сайта взаимосвязаны. В соответствии с информационным наполнением идет процесс корректировки структуры. Проанализируйте существующие материалы и с точки зрения содержания, и с точки зрения занимаемых объемов. Например, после дополнительной обработки можно использовать имеющиеся буклеты, печатные работы, фотографии и рисунки, в том числе, эмблемы. Очень важный момент при работе с материалами – соблюдение авторских прав. Относитесь с уважением к чужому труду, соблюдайте требования действующего законодательства в области интеллектуальной собственности, указывайте авторов текстов, фотографий и т. д.

В целом при компоновке страниц необходим учет ограничений, связанных с объемом размещенных на ней материалов. Ради обеспечения достойного качества конечного продукта некоторыми материалами придется пожертвовать, а некоторые следует разместить с соответствующим предупреждением о времени загрузки и возможностью их проигнорировать. Например, хорошим тоном является возможность такого выбора перед запуском флэш-элементов. А фотоальбомы, несмотря на неминуемую громоздкость даже при хорошей структуризации по разделам, могут быть очень интересны людям.

Задание

Для выполнения практической работы № 1 необходимо изучить общие принципы разработки сайта. Студент должен выбрать из табл. 1.1 один из вариантов названия будущего разрабатываемого сайта. Для определения основной темы разрабатываемого сайта необходимо сформулировать, что должно присутствовать на сайте, представляющем компанию (или проект). Разработать структуру, содержание, дизайн, макет сайта. Написать техническое задание (сценарий), собрать, подготовить информацию и материалы для разрабатываемого сайта. Определить аппаратно-программные требования сайта.

Таблица 1.1

№ варианта	*Название проектов и технологий* (расположение проектов: www.SSGA.RU/ Проекты и технологии)
1	Сеть активных базовых станций ГЛОНАСС/GPS
2	Трехмерное лазерное сканирование
3	Карты бумажные и настенные
4	Цифровые карты и ГИС
5	Инженерно-геодезические работы
6	Геодинамика
7	Метрология геодезических приборов
8	Физические явления в оптике
9	Системы тепловидения
10	Справочно-информационные ГИС на CD
11	3D-модели природных объектов
12	Голограмма
13	Цифровая фотограмметрия
14	Планетарий
15	Повышение квалификации
16	Центр тестирования и профориентации

Контрольные вопросы

1. Состав и содержание действий, предшествующих непосредственным техническим работам по созданию сайта.
2. Технологическая последовательность процесса создания сайта.
3. Недопустимые или нежелательные технические решения при создании сайта.
4. На что следует обратить особое внимание при разработке сайта?

Практическая работа № 2
Создание шаблона сайта

Цель работы: изучение и освоение принципов работы с HTML тегами, их атрибутами и умение создавать базовые элементы *web*-страницы.

Оборудование: IBM PC

Программное обеспечение: Microsoft Window

Указания к выполнению работы

1. Создание заголовочной части web-страницы.

Создать шаблон, содержащий все элементы типовой страницы. Вид HTML документа определяют инструкции или теги, обрамляющие те или иные элементы страницы.

HTML документ начинается с заголовочной части, содержащей служебную информацию и общие инструкции браузеру по отображению контента.

Листинг 02.01

```
01 <!DOCTYPE html>
2  <html>
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=win-1251" />
5     <title>Шаблон страницы HTML5</title>
6   </head>
```

Тег <!DOCTYPE html> сообщает браузеру, что страница создана по стандарту HTML5.

Парный тег <html> обрамляет все остальные секции любого HTML документа (данный тег должен быть в единственном экземпляре и должен присутствовать на каждой странице).

Парный тег <head> отделяет заголовочную часть страницы со служебной информацией.

Одиночный тег <meta> Тег <meta> предоставляет метаданные о документе HTML браузеру. Метаданные не отображаются, а только используются для служебных целей либо движком браузера, либо поисковыми системами. Метаэлементы, как правило, используется для описания страницы (description), указания ключевых слов (keywords), указания автора документа (author), указания типа контента (content) и его кодировки (charset).

Парный тег <title> – единственный тег из этой секции, который выводит видимую информацию, он определяет заголовок *web*-страницы, отображающийся в строке заголовка окна браузера.

2. Создание основной части web-страницы.

Создание тега HTML документа.

В HTML 5 для каждой части страницы имеется свой элемент.

Вот их список и краткое назначение:

1) section; назначение – определение секций. Его используют для описания определенного блока текста, например, хорошим применением этого элемента будет разбиение большой части текста на более малые, как происходит разбиение одной статьи на несколько абзацев;

2) header; назначение – определение верхней секции на странице;

3) footer; назначение – определение нижней секции на странице;

4) nav; назначение – определяет набор ссылок на другие страницы (часто используют для навигации по сайту);

5) article; назначение – выделить определенную часть текста.

Листинг 02.02

```
 01  <body>
 02    <section id="page">
 03      <header> <!-- Шапка -->
 04        <h1>Логотип</h1>
 05        <h3>Слоган</h3>
 06        <nav class="clear">
 07          <ul>
 08            <li><a href="#article1">Главная</a></li>
 09            <li><a href="#article2">Статья 1</a></li>
 10            <li><a href="#article3">Статья 2</a></li>
</ul>
11        </nav>
12      </header>
```

Парный тег <body> содержат все теги, определяющие структуру и содержание web-страницы.

Парный тег <section> используется для разделения страницы на семантические части.

Парный тег <header> определяет шапку страницы. В создаваемом шаблоне она содержит заголовки первого и третьего уровней (<h1>, <h3>) для логотипа и слогана соответственно, и элементы навигации (<nav>) в виде маркированного списка (<ul> определяет список, <li> – элементы списка).

Атрибут href тега <a> содержит ссылку на документ, который связан с данным пунктом навигационного меню, часть после # отвечает за ID статьи, к которой мы хотим перейти.

3. Добавление элементов на страницу.

Разделим линией заголовочную и основную части и расположим текст статьи.

Листинг 02.03

```
1  <!-- Статья 1 начало -->
2        <div class="line"></div>
3        <article id="article1">
4          <h2>Статья 1</h2>

5          <div class="line"></div>
```

```
6          <div class="articleBody clear">

 07          <figure>
 08          </figure>
<p>Текст первой статьи</p>
9          </div>
10       </article>
11 <!-- Статья 1 конец -->
12     <footer> <!-- Подвал -->

13       <div class="line"></div>

14       <p>Copyright 2011</p>

15     </footer>

16   </section>

17 </body>

18 </html>
```

Парный тег – тег-контейнер, выделяет логический блок.

Парный тег <figure> служит для отображения рисунка для статьи.

Парный тег <p> обозначает отдельный абзац текста.

Опишем оставшуюся часть страницы – «подвал», а также закроем секцию page и тело документа.

В результате объединения вышеописанных частей получаем текст HTML страницы, просмотрев которую в браузере получим следующий результат (рис. 2.1).

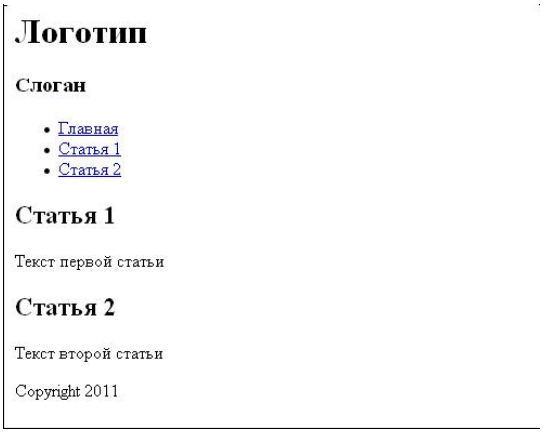


Рис. 2.1. Примерный результат выполнения практической работы № 2

Задание

Для выполнения практической работы № 2 необходимо изучить процесс создания шаблона сайта. Научиться работать с HTML тегами, их атрибутами. Научиться задавать базовую структуру HTML документа. Научиться создавать базовые элементы *web*-страницы.

Контрольные вопросы

1. Что такое базовые элементы *web*-страницы?
2. Основные компоненты IDE.
3. Назначение тегов <html>, <body>, <header>, <section>, <p>, <figure>, .
4. Что входит в состав метаданных о документе HTML?
5. С помощью какого тега задается заголовок web-страницы?

Практическая работа № 3
Оформление сайта

Цель работы: изучить принципы создания и использования каскадных стилей (CSS) для оформления элементов *web*-страницы.

Оборудование: IBM PC

Программное обеспечение: Microsoft Window

Указания к выполнению работы

1. Установка визуального оформления базовых элементов.

Следуя этим инструкциям, последовательно установим визуальное отображение элементов страницы.

Один и тот же набор атрибутов можно одновременно устанавливать для нескольких элементов страницы, последовательно указав их селекторы через запятую.

Листинг 03.02

```
1  header,footer,
2  article,section,
3  hgroup,nav,
4  figure{
 05    display:          block;
 06  }
```

Свойство display:block означает, что элемент будет отображаться как элемент блокового уровня. Блоковый элемент имеет отступы до и после себя. Кроме этого в одной строке вместе с этим элементом не могут находиться другие элементы (если только дополнительно для них не применять свойство float).

Листинг 03.03

```
 01 body{
 02   font-size:         0.825em;
 03   color:             #fcfcfc;
 04   background-color: #355664;
 05   font-family:       Arial, Helvetica, sans-serif;
 06 }
```

Здесь устанавливаются значения по умолчанию шрифта (font-family, несколько шрифтов перечисляются на тот случай, если предпочтительный не будет найден у пользователя), его размера (font-size), цвета (color), цвета фона страницы (background-color).

В следующем блоке устанавливаем внешний вид ссылок (text-decoration – оформление текста, outline – наличие рамки, border – наличие окантовки):

Листинг 03.04

```
a, a:visited {
  color:           #0196e3;
  text-decoration: none;
  outline:         none;
}
a:hover{
  text-decoration: underline;
}
a img{
  border:          none;
}
```

2. Установка параметров шрифта.

Далее оформим текст логотипа, слогана, заголовков статей:

Листинг 03.05

```
h1,h2,h3{
font-family: "Myriad Pro","Helvetica Neue",Helvetica,Arial,Sans-Serif;
  text-shadow:0 1px 1px black;
}
h1{
  /* текст логотипа */
  font-size:      3.5em;
  padding:        0.5em 0 0;
  text-transform: uppercase;
}
h3{
  /* текст слогана */
```

```
13    font-family: forte,"Myriad Pro","Helvetica
Neue",Helvetica,Arial,Sans-Serif;

14    font-size:2em;

15    font-weight:normal;
16    margin:    0 0 1em;
}

17 h2{
18    font-size:2.2em;

19    font-weight:normal;

20    letter-spacing: 0.01em;

21    text-transform: uppercase;

22 }
```

Атрибут стиля text-shadow задает параметры тени: text-shadow: none | <цвет> <горизонтальное смещение> <вертикальное смещение> [<радиус размытия>]

Атрибут стиля padding позволяет сразу указать величины внутренних отступов со всех сторон элемента web-страницы:

padding: <отступ 1> [<отступ 2> [<отступ 3> [<отступ 4>]]]

Атрибут стиля text-transform позволяет изменить регистр символов текста:

text-transform: capitalize|uppercase|lowercase|none|inherit

Можно преобразовать текст к верхнему (значение uppercase этого атрибута) или нижнему (lowercase) регистру, преобразовать к верхнему регистру первую букву каждого слова (capitalize) или оставить в изначальном виде (none).

Атрибут стиля font-weight устанавливает «жирность» шрифта: font-weight: normal|bold|bolder|lighter|100|200|300|400|500|600| 700|800|900|inherit

Атрибут margin задает величины отступа одновременно со всех сторон элемента web-страницы:

margin: <отступ 1> [<отступ 2> [<отступ 3> [<отступ 4>]]]

Атрибут стиля letter-spacing позволяет задать дополнительное расстояние между символами текста:

letter-spacing: normal|<расстояние>

Оформление абзацев текста:

Листинг 03.06

```
01 p{
02   line-height:  1.5em;
03   padding-bottom:     1em;
\endash   }
```

Добавление элементов оформления секций и статей.

Определим вид линии, разделяющей секции страницы:

Листинг 04.07

```
01 .line{
02   height:       1px;
03   background-color:   #24404c;
04   border-bottom: 1px solid #416371;
05   margin:       1em 0;
06   overflow:     hidden;
07 }
```

Атрибут стиля overflow задает поведение контейнера при переполнении:

overflow: visible|hidden|scroll|auto|inherit

Также установим представление линии, отделяющей заголовок статьи от текста

Листинг 03.08

```
01 article .line{
02   background-color: #15242a;
03   border-bottom-color:     #204656;
04   margin:           1.3em 0;
05 }
```

и линию, отделяющую нижнюю часть страницы:

Листинг 03.09

```
01 footer .line{
02   margin:       2em 0;
1. }
```

Установка визуального стиля для навигационной панели.

Теперь перейдем к установке стилей отображения навигационной панели и ее элементов:

Листинг 03.10

```
01 nav{
02   background:         #f8f8f8;
03   padding:            0 5px;
04   position:           absolute;
05   right:              0;
06   top:                4em;
07   border:             1px solid #FCFCFC;
-moz-box-shadow:0 1px 1px #333333;
\endash   -webkit-box-shadow: 0 1px 1px #333333;
0    box-shadow:    0 1px 1px #333333;
}
и  nav ul li{
13   display:            inline;
}
и  nav ul li a, nav ul li a:visited{
и    color:#565656;
 17  display:            block;
 18  float:              left;
 19  font-size:          1.25em;
 20  font-weight:        bold;
```

```
 21  margin:              5px 2px;
 22  padding:             7px 10px 4px;
 23  text-shadow:         0 1px 1px white;
 24  text-transform:      uppercase;
}
в  nav ul li a:hover{
в    text-decoration:none;
в    background-color:   #f0f0f0;
в  }
в  nav, article, nav ul li a,figure{
в    /* Применение закругленных углов у элементов */
в    -moz-border-radius: 10px;
в    -webkit-border-radius:   10px;
в    border-radius:10px;
в  }
```

Конструкция nav ul li a является комбинированным стилем и означа-ет, что ссылка будет выглядеть соответствующе, только если окажется последовательно вложенной в элементы nav ul li.

конструкции nav ul li a:hover элемент hover является новым компонентом CSS3. Он является псевдоклассом и позволяет создавать альтернативные стили для элементов. Эти стили будут присваиваться элементам при наведении на них курсором мыши. Указывать псевдоклассы следует после имени элемента или класса после знака ":". Данный псевдокласс применим не только к ссылкам, по большому счету, его можно применять для всех тегов.

Атрибут стиля box-shadow создает тень вокруг изображения или другого элемента. Box-shadow – это стандартное написание, варианты -moz-box-shadow и –webkit-box-shadow используются для браузеров FireFox 2.Chrome, Safari соответственно.

Display:inline означает, что элемент отображается как встроенный, содержимое блочных элементов начинается с того места, где окончился предыдущий элемент.

3. Установка визуального стиля для статей.

Установим оформление основных элементов содержания страницы – статей. Зададим окантовку, фон, размещение рисунков и текста:

Листинг 03.11

```
01 #page{
02   width:           960px;
03   margin:          0 auto;
04   position:        relative;
05 }
06 article{
07   background-color:   #213E4A;
08   margin:          3em 0;
09   padding:         20px;
```

```
к    text-shadow:0 2px 0 black;
к  }
к  figure{
 13  border:          3px solid #142830;
 14  float:           right;
 15  height:          300px;
 16  margin-left:     15px;
 17  overflow:        hidden;
 18  width:           500px;
}
   figure:hover{
     -moz-box-shadow: 0 0 2px #4D7788;
     -webkit-box-shadow: 0 0 2px #4D7788;
     box-shadow:0 0 2px #4D7788;
   }
   figure img{
     margin-left:-60px;
   }
```

Атрибут стиля overflow задает поведение контейнера при переполнении.

Доступны четыре значения:

visible – высота контейнера увеличится, чтобы полностью вместить все содержимое (обычное поведение);

hidden – не помещающееся в контейнер содержимое будет обрезано. Контейнер сохранит свои размеры;

scroll – в контейнере появятся полосы прокрутки, с помощью которых можно просмотреть не помещающуюся часть содержимого. Эти полосы прокрутки будут присутствовать в контейнере всегда, даже если

1. них нет нужды;

auto – полосы прокрутки появятся в контейнере, только если в них возникнет необходимость.

2. ***Установка визуального стиля для нижней части страницы.***

И теперь осталось задать внешний вид нижней части страницы:

Листинг 03.12

```
01 footer{
a    margin-bottom: 30px;
a    text-align:center;
a    font-size:0.825em;
a  }
a  footer p{
a    margin-bottom: -2.5em;
a    position:relative;
a  }
a  footer a,footer a:visited{
 11  color:         #cccccc;
 12  background-color:   #213e4a;
 13  display:       block;
 14  padding:       2px 4px;
 15  z-index:       100;
 16  position:      relative;
```

```
}
3. footer a:hover{
4.   text-decoration:none;
5.   background-color:   #142830;
6. }
7. footer a.by{
8.   float:left;
24
}
4. footer a.up{
5.   float:right;
6. }
```

После того как в файл внесены все необходимые настройки, его нужно сохранить с расширением css и привязать к созданному ранее HTML файлу, дописав после тега <title> строку:

Листинг 03.13

```
01 <link rel="stylesheet" type="text/css" href="имя_файла.css" />
```

Задание

Для выполнения практической работы № 3 необходимо изучить процесс использования каскадных стилей, научиться создавать стили и атрибуты и привязывать их к элементам HTML страницы, научиться задавать оформление элементов HTML страницы с помощью каскадных стилей, научиться создавать интерактивные элементы с помощью каскадных стилей.

Контрольные вопросы

1. Роль селектора в технологии CSS.
2. Какие атрибуты используются для задания размера, цвета, «жирности», вида шрифта и расстояния между символами в тексте?
3. Перечислить применяемые значения атрибута стиля overflow.
4. Порядок действий при создании выпадающего меню.

Практическая работа № 4
Применение языка сценариев JAVASCRIPT

Цель работы: изучить основы использования языка сценариев JavaScript для создания интерактивных элементов на *web*-страницах.

Оборудование: IBM PC

Программное обеспечение: Microsoft Window

Указания к выполнению работы

Для примера создадим на странице объекты обоих типов.

Автономным элементом будет прямоугольник, на котором рисуется волна, а интерактивным – фотогалерея.

1. Объект canvas.

Для создания визуального эффекта волны воспользуемся новым тегом HTML – canvas. Он предназначен для создания растрового изображения при помощи JavaScript. На сегодняшний день canvas чаще используется для построения графиков, простой анимации и игр в веб-браузерах.

Создадим холст в объекте, предназначенном для вставки изображения для первой статьи.

Листинг 04.01

```
01 <figure>
02 <canvas id="canvas"></canvas>
03 </figure>
```

Код на языке JavaScript может находиться как внутри текста HTML страницы, так и во внешнем файле. Второй способ удобен более «прозрачной» файловой структурой проекта, уменьшением размера и упрощением содержимого HTML документов, возможностью использовать одни \endash те же скрипты на разных страницах, при этом изменения в коде файла сразу отразятся на всех связанных с ним страницах.

1)Добавление сценариев JavaScript на web-страницу.

Код создаваемого скрипта поместим в файл waves.js и присоединим его к странице, добавив следующую строчку в конце секции body:

Листинг 04.02

```
01 </section> <!-- Закрываем секцию #page -->
02 <script language="JavaScript" src="src/waves.js"></script>
03 </body>
```

2. Установка оформления.

7 файле CSS определим фоновый цвет, размеры и положение объекта canvas.

Листинг 04.03

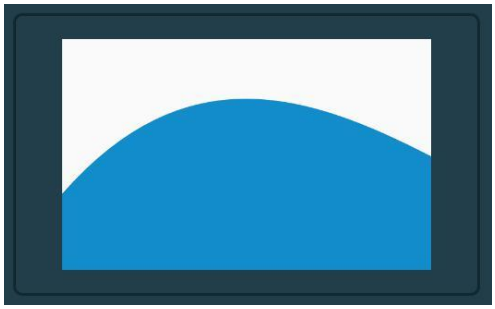
```
01 #canvas {
02   background-color:   #FAFAFA;
03   position: relative;
04   margin-top:    25px;
05   margin-left:   50px;
width: 400px;  height: 250px;
3. }
```

Далее представлен код, содержащийся в файле waves.js:

Листинг 04.04

```
 01 var canvas     = document.getElementById('canvas');
 02 var ctx        = canvas.getContext('2d');
 03 canvas.width   = window.innerWidth;
 04 canvas.height = window.innerHeight;
 05 var waves = ["rgba(157, 187, 210, 0.3)"]
 06 var i = 0;
 07
 08 function draw() {
 09 canvas.width = canvas.width;
 10
 11 var offset     = i + 1 * Math.PI * 12;
 12 ctx.fillStyle = "#138CCB";
 13
 14 var randomLeft            = (Math.sin(offset/100)+ 1) / 2 * 150;
 15 var randomRight          = (Math.sin((offset/100) + 10) + 1) / 2 * 150;
 16 var randomLeftConstraint  = (Math.sin((offset/60)+ 2)+ 1) / 2 * 150;
 17 var randomRightConstraint = (Math.sin((offset/60)+ 1)+ 1) / 2 * 150;
 18
ctx.beginPath();
B  ctx.moveTo(0, randomLeft + 150);
B
B  // ctx.lineTo(canvas.width, randomRight + 100);
B  ctx.bezierCurveTo(canvas.width / 3, randomLeftConstraint, canvas.width /
6)    * 2, randomRightConstraint, canvas.width, randomRight + 150);
13 ctx.lineTo(canvas.width , canvas.height);
14 ctx.lineTo(0, canvas.height);
15 ctx.lineTo(0, randomLeft + 150);
в  ctx.closePath();
```

Результат представлен на рис. 4.1.



Задание

Для выполнения практической работы № 4 необходимо изучить принципы использования сценариев JavaScript, научиться создавать интерактивные элементы с использованием JavaScript, научиться пользоваться сторонними фреймворками.

Контрольные вопросы

1. Определение и назначение *web-сценариев.*
2. Перечислить принципы использования сценариев JavaScript
3. Привести примеры динамического содержимого на *web*-страницах.
4. Порядок создания автономного элемента типа падающих снежинок.
5. Сформулировать назначение и перечислить настройки объекта slideshow.

Практическая работа № 5
Подгружаемые элементы страниц сайта

Цель работы: изучить принципы составления единой *web*-страницы из отдельных подгружаемых элементов.

Оборудование: IBM PC

Программное обеспечение: Microsoft Window

Указания к выполнению работы

и ***Создание отдельных подгружаемых элементов web-страницы.***

Заполним текстовые блоки, содержащие статьи текстом из внешнего файла. Для этого создадим дополнительную папку **html,** в которую положим два html файла с простой структурой, содержащих тексты первой и второй статей.

и основном html файле обрамим теги параграфов с заготовками текста тегами **div**, в которые будут загружаться созданные страницы:

Листинг 05.01

```
01 <div id="txtart1">

1. <p>Текст первой статьи</p>

2. </div>
```

Листинг 05.02

```
01 <div id="txtart2">

в  <p>Текст второй статьи</p>
в  </div>
```

с ***Подключение загружаемых элементов.***

Теперь, чтобы поместить текст в элементы страницы, создадим в теле страницы внедренный скрипт со следующим текстом:

Листинг 05.03

```
01 <script language="JavaScript">

к  $('#txtart1').load('html/page1.html');
к  $('#txtart2').load('html/page2.html');
к  </script>
```

Данный вариант будет «работать», только если подключен фреймворк jQuery, как в предыдущей работе.

В итоге получим следующий результат (рис. 5.1).

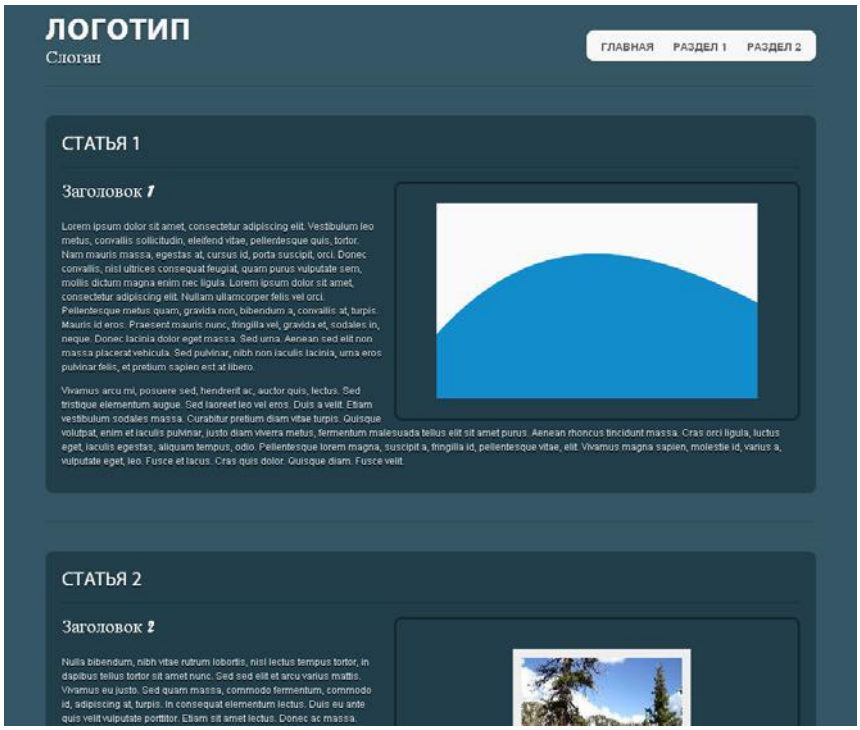


Рис. 5.1. Примерный результат выполнения практических работ по созданию *web*-страницы

Задание

Для выполнения практической работы № 5 необходимо изучить принципы модульного построения *web*-страниц, научиться разделять HTML страницу на отдельные элементы, научиться формировать HTML страницу из отдельных модулей.

Контрольные вопросы

1) В чем заключается сущность и преимущества модульного принципа построения *web*-страницы?

2) Порядок действий при реализации модульного подхода к созданию *web*-страницы.

Информационные источники

Основные источники:

1. Крис Джамса, Конрад Кинг, Энди Андерсон. Эффективнй курс по креативному Web-дизайну. HTML, XHTML, CSS, JavaScript, PHP, ASP, ActiveX. Текст, графика, звук и анимация. Москва, Санкт-Петербург. Киев. 2015. 665 с.
2. Мотросов А., Сергеев А., Чаунин М. HTML. Санкт-Петербург «БХВ - Петербург», 2015. 674с.
3. Майкл Гурвиц,Лора Мак Кейб Использование Macromedia Flash Москва, Санкт-Петербург. Киев. 2014. 706 с.

Дополнительные источники:

1. Петюшкин А., HTML - Экспресс курс. Санкт-Петербург «БХВ - Петербург», 2016. 258с.
2. 2. Полонская Е.Л. Самоучитель. Язык HTML – Москва, Санкт-Петербург. Киев. 2016 322 с.
3. 3. Гончаров А. Самоучитель HTML. СПб.: Питер, 2016. — 240 с.
4. 4. Тим К. Чанг • Шон Кларк • Эрик Е. Долецки • Джон Игнацио Джелос Михаэль Грюндвиг • Джоб Макар • Макс Ошман • Вильям Б. Сандерс • Скотт Смит Популярные WEB приложения на FLASH MX. Пер. С англ. – М.: КУДИЦ – ОБРАЗ 2015. 272 с.

Интернет-ресурсы:

1.	http://ru.vectorboom.com	« Портал дизайна»	*04.09.17*
2.	http://skillsup.ru	«**Skillsup** — крупнейший обучающий портал для дизайнеров и творческих людей»	*04.09.17*
3.	http://design-mania.ru	«Блог о Веб-дизайне»	*04.09.17*
4.	http://egraphic.ru	«Веб-дизайн»	*04.09.17*
5.	http://compteacher.ru/	«Видеоуроки»	*04.09.17*

6.	http://ru.wikibooks.org/	«Викиучебник»	*04.09.17*
7.	http://balbesof.net/	«Все о графике и дизайне»	*04.09.17*
8.	http://www.dejurka.ru	«Журнал по дизайну»	*04.09.17*
9.	http://www.itstan.ru/	«Информация»	*04.09.17*
10.	http://www.novtex.ru/IT/	«Научно-технический и научно производственный журнал Информационные Технологии»	*04.09.17*
11.	http://www.intuit.ru/	«Национальный Открытый Университет «ИНТУИТ»	*04.09.17*
12.	http://www.ict.edu.ru/	«Портал Информационно-коммуникационные технологии в образовании»	*04.09.17*
13.	http://inftech.webservis.ru	«Сайт Информационных технологий»	*04.09.17*
14.	http://www.citmgu.ru/	«Центр информационных технологий»	*04.09.17*