

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

«БРАТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Кафедра информатики и прикладной математики

УТВЕРЖДАЮ:

Проректор по учебной работе

_____ Е.И. Луковникова

«_____» _____ 201__ г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
ИНТЕРНЕТ ТЕХНОЛОГИИ**

Б1.В.ДВ.05.02

НАПРАВЛЕНИЕ ПОДГОТОВКИ

44.03.01 Педагогическое образование

ПРОФИЛЬ ПОДГОТОВКИ

История

Программа академического бакалавриата

Квалификация (степень) выпускника: бакалавр

| | |
|------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| 1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ | 3 |
| 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ | 4 |
| 3. РАСПРЕДЕЛЕНИЕ ОБЪЕМА ДИСЦИПЛИНЫ | 4 |
| 3.1 Распределение объёма дисциплины по формам обучения..... | 4 |
| 3.2 Распределение объёма дисциплины по видам учебных занятий и трудоемкости | 4 |
| 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ | 5 |
| 4.1 Распределение разделов дисциплины по видам учебных занятий | 5 |
| 4.2 Содержание дисциплины, структурированное по разделам и темам | 5 |
| 4.3 Лабораторные работы..... | 5 |
| 4.4 Практические занятия..... | 5 |
| 4.5. Контрольные мероприятия: курсовая работа..... | 5 |
| 5. МАТРИЦА СООТНЕСЕНИЯ РАЗДЕЛОВ УЧЕБНОЙ ДИСЦИПЛИНЫ К ФОРМИРУЕМЫМ В НИХ КОМПЕТЕНЦИЯМ И ОЦЕНКЕ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ | 6 |
| 6. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ..... | 7 |
| 7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ..... | 7 |
| 8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО – ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ» НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ | 7 |
| 9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ..... | 8 |
| 9.1. Методические указания для обучающихся по выполнению практических работ | 8 |
| 10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ | 44 |
| 11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ | 44 |
| Приложение 1. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине..... | 45 |
| Приложение 2. Аннотация рабочей программы дисциплины | 48 |
| Приложение 3. Протокол о дополнениях и изменениях в рабочей программе | 49 |

1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Вид деятельности выпускника

Дисциплина охватывает круг вопросов, относящихся к педагогическому виду профессиональной деятельности выпускника в соответствии с компетенциями и видами деятельности, указанными в учебном плане.

Цель дисциплины

Целью преподавания дисциплины является изучение и освоение современных web-технологий.

Задачи дисциплины

Задачами дисциплины являются:

- изучение основ HTML и CSS;
- изучение технологий разработки web-сайтов;
- изучение основ разработки интернет приложений.

| Код компетенции | Содержание компетенций | Перечень планируемых результатов обучения по дисциплине |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | 2 | 3 |
| ОК-3 | способность использовать естественнонаучные и математические знания для ориентирования в современном информационном пространстве | знать: основные виды современных web-технологий и принципы их функционирования уметь: структурировать информацию для её публикации на web-ресурсах; применять современные web-технологии в профессиональной деятельности; владеть: навыками разработки макетов web-страниц, отвечающих современным требованиям. |
| ПК-7 | способность организовывать сотрудничество обучающихся, поддерживать активность и инициативность, самостоятельность обучающихся, развивать их творческие способности | знать: особенности построения коммуникации с участниками образовательного процесса; уметь: конструктивно взаимодействовать с участниками образовательного процесса; владеть: навыками организации сотрудничества обучающихся. |

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина Б1.В.ДВ.05.02 Интернет технологии относится к дисциплинам по выбору.

Дисциплина Интернет технологии базируется на знаниях, полученных при изучении дисциплины Б1.В.15 Информационные технологии.

Дисциплина «Интернет технологии» представляет основу для изучения дисциплин: Б1.В.ДВ.06.02 Прикладные программы обработки информации.

Такое системное междисциплинарное изучение направлено на достижение требуемого ФГОС уровня подготовки по квалификации бакалавр.

3. РАСПРЕДЕЛЕНИЕ ОБЪЕМА ДИСЦИПЛИНЫ

3.1. Распределение объема дисциплины по формам обучения

| Форма обучения | Курс | Семестр | Трудоемкость дисциплины в часах | | | | | | Курсовая работа (проект), контрольная работа, реферат, РГР | Вид промежуточной аттестации |
|--------------------------------------|------|---------|---------------------------------|------------------|--------|---------------------|----------------------|------------------------|------------------------------------------------------------|------------------------------|
| | | | Всего часов (с экз.) | Аудиторных часов | Лекции | Лабораторные работы | Практические занятия | Самостоятельная работа | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| Очная | 2 | 4 | 72 | 36 | – | – | 36 | 36 | – | зачет |
| Заочная | – | – | – | – | – | – | – | – | – | – |
| Заочная (ускоренное обучение) | – | – | – | – | – | – | – | – | – | – |
| Очно-заочная | – | – | – | – | – | – | – | – | – | – |

3.2. Распределение объема дисциплины по видам учебных занятий и трудоемкости

- учебный план 2015/2016 г.г

| Вид учебных занятий | Трудоемкость (час.) | в т.ч. в интерактивной, активной, инновационной формах, (час.) | Распределение по семестрам, (час.) |
|------------------------------------------------------------------|---------------------|----------------------------------------------------------------|------------------------------------|
| | | | 4 |
| 1 | 2 | 3 | 4 |
| I. Контактная работа обучающихся с преподавателем (всего) | 36 | 8 | 36 |
| Практические занятия (ПЗ) | 36 | 8 | 36 |
| Групповые (индивидуальные) консультации | + | – | + |
| II. Самостоятельная работа обучающихся (СР) | 36 | – | 36 |
| Подготовка к практическим занятиям | 16 | – | 16 |
| Подготовка к зачету | 20 | – | 20 |
| III. Промежуточная аттестация зачет | + | – | + |
| Общая трудоемкость дисциплины час. | 72 | – | 72 |
| | зач. ед. | 2 | 2 |

| <i>Вид учебных занятий</i> | <i>Трудоёмкость (час.)</i> | <i>в т.ч. в интерактивной, активной, инновационной формах, (час.)</i> | <i>Распределение по семестрам, (час.)</i> |
|------------------------------------------------------------------|----------------------------|-----------------------------------------------------------------------|-------------------------------------------|
| | | | 4 |
| 1 | 2 | 3 | 4 |
| I. Контактная работа обучающихся с преподавателем (всего) | 36 | 12 | 36 |
| Практические занятия (ПЗ) | 36 | 12 | 36 |
| Групповые (индивидуальные) консультации | + | – | + |
| II. Самостоятельная работа обучающихся (СР) | 36 | – | 36 |
| Подготовка к практическим занятиям | 16 | – | 16 |
| Подготовка к зачету | 20 | – | 20 |
| III. Промежуточная аттестация зачет | + | – | + |
| Общая трудоёмкость дисциплины час. зач. ед. | 72 | – | 72 |

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1. Распределение разделов дисциплины по видам учебных занятий

- для очной формы обучения:

| <i>№ раздела и темы</i> | <i>Наименование раздела и тема дисциплины</i> | <i>Трудоёмкость, (час.)</i> | <i>Виды учебных занятий, включая самостоятельную работу обучающихся и трудоёмкость; (час.)</i> | |
|-------------------------|-----------------------------------------------|-----------------------------|------------------------------------------------------------------------------------------------|-------------------------------------------|
| | | | <i>учебные занятия</i> | <i>самостоятельная работа обучающихся</i> |
| | | | <i>практические задания</i> | |
| 1 | 2 | 3 | 4 | 5 |
| 1. | Язык гипертекстовой разметки HTML | 46 | 26 | 20 |
| 2. | Каскадные таблицы стилей CSS | 26 | 10 | 16 |
| ИТОГО | | 72 | 36 | 36 |

4.2. Содержание дисциплины, структурированное по разделам и темам

Учебным планом не предусмотрено

4.3. Лабораторные работы

Учебным планом не предусмотрено

4.4. Практические задания

- учебный план 2015/2016 г.г

| <i>№ п/п</i> | <i>Номер раздела дисциплины</i> | <i>Наименование тем практических работ</i> | <i>Объем (час.)</i> | <i>Вид занятия в интерактивной, активной, инновационной формах, (час.)</i> |
|------------------|---------------------------------|--------------------------------------------|---------------------|----------------------------------------------------------------------------|
| 1 | 1. | Создание простейшего HTML-документа | 10 | Тренинг (8 час.) |
| 2 | 1. | Создание Web-страницы с фреймами | 12 | - |
| 3 | 2. | Таблицы стилей или CSS | 10 | - |
| 4 | 1. | HTML-формы | 4 | |
| ИТОГО | | | 36 | 8 |

- учебный план 2017/2018 г.г

| <i>№ п/п</i> | <i>Номер раздела дисциплины</i> | <i>Наименование тем практических работ</i> | <i>Объем (час.)</i> | <i>Вид занятия в интерактивной, активной, инновационной формах, (час.)</i> |
|------------------|---------------------------------|--------------------------------------------|---------------------|----------------------------------------------------------------------------|
| 1 | 1. | Создание простейшего HTML-документа | 10 | Тренинг (8 час.) |
| 2 | 1. | Создание Web-страницы с фреймами | 12 | - |
| 3 | 2. | Таблицы стилей или CSS | 10 | - |
| 4 | 1. | HTML-формы | 4 | Тренинг (8 час.) |
| ИТОГО | | | 36 | 12 |

4.5. Контрольные мероприятия: курсовой проект (курсовая работа), контрольная работа, РГР, реферат

Учебным планом не предусмотрено

5. МАТРИЦА СООТНЕСЕНИЯ РАЗДЕЛОВ УЧЕБНОЙ ДИСЦИПЛИНЫ К ФОРМИРУЕМЫМ В НИХ КОМПЕТЕНЦИЯМ И ОЦЕНКЕ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

| <i>№, наименование разделов дисциплины</i> | <i>Компетенции</i> | <i>Кол-во часов</i> | <i>Компетенции</i> | | <i>Σ комп.</i> | <i>t_{ср}, час</i> | <i>Вид учебных занятий</i> | <i>Оценка результатов</i> |
|---------------------------------------------|--------------------|---------------------|--------------------|-----------|----------------|----------------------------|----------------------------|---------------------------|
| | | | <i>ОК</i> | <i>ПК</i> | | | | |
| | | | <i>3</i> | <i>7</i> | | | | |
| 1 | | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1. Язык гипертекстовой разметки HTML | | 46 | + | + | 2 | 23 | ПЗ, СРС | зачет |
| 2. Каскадные таблицы стилей CSS | | 26 | + | + | 2 | 13 | ПЗ, СРС | зачет |
| <i>всего часов</i> | | 72 | 36 | 36 | 2 | 36 | | |

6. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

1. Ллойд, Й. Создай свой веб-сайт с помощью HTML и CSS : учебное пособие / Й. Ллойд; Пер. с англ. - Санкт-Петербург : Питер, 2013. - 416 с.

2. Крахоткина, Е.В. Технологии разработки Internet-приложений : учебное пособие / Е.В. Крахоткина ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет». - Ставрополь: СКФУ, 2016. - 124 с.: ил. - Библиогр. в кн.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=459070> (20.02.2018).

3. Гениатулина, Е.В. CMS – системы управления контентом : учебное пособие / Е.В. Гениатулина ; Министерство образования и науки Российской Федерации, Новосибирский государственный технический университет. - Новосибирск : НГТУ, 2015. - 63 с. : ил. - Библиогр. в кн. - ISBN 978-5-7782-2696-8; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=438332> (20.02.2018).

7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

| № | Наименование издания | Вид занятия (ПЗ) | Количество экземпляров в библиотеке, шт. | Обеспеченность, (экз./чел.) |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|------------------------------------------|-----------------------------|
| 1 | 2 | 3 | 4 | 5 |
| Основная литература | | | | |
| 1. | Ллойд, Й. Создай свой веб-сайт с помощью HTML и CSS : учебное пособие / Й. Ллойд; Пер. с англ. - Санкт-Петербург : Питер, 2013. - 416 с. | ПЗ | 10 | 0,5 |
| Дополнительная литература | | | | |
| 2. | Кобзов, А. Ю. Программные средства разработки Web- страниц и презентаций : методические указания к выполнению лабораторных работ / А. Ю. Кобзов, А. В. Кобзова. - Братск : БрГУ, 2014. - 43 с. | ПЗ | 45 | 1 |
| 3. | Федяев, П. А. Программные продукты проектирования WEB-страниц и презентаций : методические указания по выполнению практических занятий / П. А. Федяев, Т. А. Лебедева. - Братск : БрГУ, 2012. - 37 с. | ПЗ | 93 | 1 |

8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО - ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ» НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1. Электронный каталог библиотеки БрГУ http://irbis.brstu.ru/CGI/irbis64r_15/cgiirbis_64.exe?LNG=&C21COM=F&I21DBN=BOOK&P21DBN=BOOK&S21CNR=&Z21ID=.

2. Электронная библиотека БрГУ <http://ecat.brstu.ru/catalog> .

3. Электронно-библиотечная система «Университетская библиотека online» <http://biblioclub.ru> .

4. Электронно-библиотечная система «Издательство «Лань» <http://e.lanbook.com> .

5. Информационная система "Единое окно доступа к образовательным ресурсам"
<http://window.edu.ru> .
6. Научная электронная библиотека eLIBRARY.RU <http://elibrary.ru> .
7. Университетская информационная система РОССИЯ (УИС РОССИЯ)
<https://uisrussia.msu.ru/> .
8. Национальная электронная библиотека НЭБ <http://xn--90ax2c.xn--p1ai/how-to-search/>

9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

| Вид учебных занятий | Организация деятельности обучающихся |
|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Практические задания | Работа с конспектом лекций, обобщение, систематизация, углубление и конкретизация полученных теоретических знаний, выработка способности и готовности их использования на практике. Подготовка ответов к контрольным вопросам, работа с основной и дополнительной литературой, необходимой для освоения дисциплины. Выполнение заданий, решение задач, активное участие в интерактивной, активной, инновационной формах обучения, составление отчетов. |
| Самостоятельная работа обучающихся | <i>Подготовка к практическим заданиям.</i> Проработка основной и дополнительной литературы, терминов, сведений, формул требующихся для запоминания и являющихся основополагающими в разделе. Конспектирование прочитанных литературных источников. Проработка материалов по изучаемому вопросу, с использованием на рекомендуемых ресурсах информационно-телекоммуникационной сети «Интернет». Выполнение заданий преподавателя, необходимых для подготовки к участию в интерактивной, активной, инновационных формах обучения по изучаемой теме. <i>Подготовка к зачету.</i> При подготовке к зачету необходимо ориентироваться на конспекты лекций, рекомендуемую литературу, использовать рекомендуемые ресурсы информационно-телекоммуникационной сети «Интернет». |

9.1. Методические указания для обучающихся по выполнению практических работ

Практическое занятие № 1. Создание простейшего HTML-документа

Цель работы:

- научиться создавать простейший гипертекстовый документ;
- научиться использовать теги форматирования шрифта и абзаца;
- научиться выполнять вставку рисунков в HTML-документ;
- научиться создавать закладки и гиперссылки;
- научиться использовать таблицы для оформления WEB-страниц;
-

Основы языка разметки гипертекста - HTML.

Базовым элементом языка разметки гипертекста является - ТЕГ (дескриптор, маркер). Тег всегда заключен между скобками < > и имеет следующий вид:

<ТЕГ параметр1="ЗНАЧЕНИЕ" ... параметрN="`ЗНАЧЕНИЕ">

Теги бывают одиночными и контейнерными. *Контейнером* называется пара: открывающий <ТЕГ> и закрывающий </ТЕГ>.

<ТЕГ> Содержимое контейнера </ТЕГ>

Открывающий тег служит для указания программе-браузера начала какого-либо объекта или задания свойств объектов помещенных в контейнер. Закрывающий тег служит для указания программе-браузеру о конце объекта или окончания применения свойств, заданных в открывающем теге. Параметры (атрибуты) тега задают значения свойств данного объекта или объектов помещенных в контейнер. Значения свойств, содержащие пробелы, берутся в кавычки, в остальных случаях кавычки можно опустить.

HTML документ представляет собой обычный текстовый файл, содержащий маркированный тегами форматирования текст, а так же заданные специальными тегами ссылки на графические и прочие файлы мультимедиа, ссылки на другие документы HTML и ресурсы Internet.

Документ HTML начинается открывающим тегом <HTML> и заканчивается закрывающим тегом </HTML>. Между данной парой контейнерных тегов располагаются две другие основные части HTML документа:

заголовок заключенный в контейнер <HEAD>...</HEAD> и
тело документа в контейнере <BODY>...</BODY>. Таким образом структура простого HTML документа выглядит примерно так:

Структура HTML-документа

```
<!DOCTYPE HTML PUBLIC "-//IETF/DTD HTML 3.2 // EN" >
<HTML>
<HEAD> Заголовок документа </HEAD>
<BODY> Тело документа </BODY>
</HTML>
```

Объявление <!DOCTYPE>.

Элемент <!DOCTYPE> должен первым указываться в документе HTML (теоретически). Он сообщает серверу WEB способ обработки документа и то, какие дескрипторы могут находиться на странице, хотя чаще всего он игнорируется браузерами. Поэтому его применение строго не обязательно.

Синтаксис: <!DOCTYPE HTML "текст""URL">

Здесь текст определяет версию HTML, а URL позволяет браузерам пользователей загрузить DTD например:

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD HTML 4.0 // EN"
"http://WWW.W3.ORG/TR/REC-HTML40/strict.dtd">
```

Тэг <HTML>.

Тэг <HTML> определяет границы документа HTML, ему соответствует конечный тэг </HTML>. Между этими двумя тэгами располагается собственно весь документ. Как и <!DOCTYPE> тэги <HTML> и </HTML> - не являются строго обязательными. Но, все-таки, их использование является правилами хорошего тона т.к. браузеры у пользователей могут быть всякие и не известно - насколько корректно они визуализируют такой код.

В дополнение к обязательной структуре настоятельно рекомендуется вставлять различные структурные детали.

Каждый HTML документ должен содержать основную информацию о его происхождении.

Если Вы стремитесь к тому, чтобы люди отыскивали Ваш документ по соответствующим связям, важность предоставления информации о его происхождении становится очевидной. Когда пользователь найдет Ваш документ с помощью, например, поискового ресурса AltaVista, он, вероятнее всего, захочет узнать, к какому виду относится документ. Поэтому каждый файл HTML должен предоставлять самую основную информацию (или связи к информации) о его происхождении и природе. Например, в собрании книгоподобных документов, разделенных на малые файлы, каждый файл должен содержать, по крайней мере, связь к "первой странице" "книги" (home page).

О происхождении документа должна быть представлена, по крайней мере, следующая информация:

Автор документа, имеющий уникальное имя. При этом должна быть задана связь с домашней страницей автора. Если у документа несколько авторов, определите их всех, а также роль каждого из них; например, ведущего автора, редактора, действующего спонсора, а также лиц, формально отвечающих за документ. Дата создания документа или его последней модификации, или и та и другая. Представляемая дата должна быть понятна во всем мире; в частности, название месяца лучше писать словом, а не цифрой. Контекст документа и его статус, например: часть официальной документации компании об одном из ее продуктов, или часть личной информации о хобби автора, или что-то другое. Адрес (URL) документа. Такая информация зачастую чрезмерна, однако она может быть очень полезной, когда кому-то нужна копия именно того документа, который он нашел. Лучше не полагаться на браузер (и пользователя), добавляющих такую информацию, когда сделана копия документа.

Элемент <BODY>.

Элемент <BODY> предназначается для выделения той части документа, которая будет визуализирована для пользователя. Он имеет как начальный, так и конечный теги. Начальный тег <BODY> может иметь несколько атрибутов.

Вложенные атрибуты элемента <BODY>:

- **BACKGROUND** Атрибут задает графическое изображение, которое как черепица заполнит фон документа. Синтаксис: <BODY BACKGROUND="(URL)(путь) имя файла">
- **BGCOLOR** Этот атрибут задает цвет фона документа при помощи шестнадцатеричных значений интенсивности цветов RGB, или при помощи строчного литерала, соответствующего названию цвета. Синтаксис: <BODY BGCOLOR="#ff0000">или <BODY BGCOLOR="RED">
- **TEXT** Этот атрибут задает используемый по умолчанию цвет текста, который не является гиперссылкой. По умолчанию такой текст будет черным. Синтаксис: <BODY TEXT="цвет">

- **LINK** Этот атрибут задает цвет гиперссылки, в большинстве браузеров он задан по умолчанию темно синим. Синтаксис: <BODY LINK="цвет">
- **ALINK** Этот атрибут задает цвет активной гиперссылки, он меняет цвет гиперссылки в момент щелчка по ней мышью, не желательно задавать ему цвет фона по понятным причинам. Синтаксис: <BODY ALINK="цвет">
- **VLINK** Этот атрибут задает цвет посещенной гиперссылки, не желательно задавать ему цвет фона и цвет атрибута LINK по понятным причинам. Синтаксис: <BODY VLINK="цвет">

Заголовочные тэги.

Элемент <HEAD> </HEAD> определяет заголовок документа.

BASE - базовый, основной URL Цель Задание базового URL для относительных URL в документе (например, в атрибутах HREF элемента A). Этот элемент часто используется для отображения документов. Например:

```
<BASE href="http://foo.com/index.html">
```

Элемент BASE непосредственно не отображается в документе. Основной синтаксис <BASE HREF="URL">. Так как в документе допускается только один элемент BASE, Вы не можете иметь различные базовые URL в различных частях файла HTML. При отсутствии элемента BASE в документе URL самого документа становится базовым в пределах документа.

META - метаинформация.

Используется для задания метаинформации (информации о документе), т.е. пар имя/значение, описывающих свойства документа, например, авторство, истечение даты, список ключевых слов и т.д. Типичное отображение Элементы META не влияют на отображение самого документа. Они могут давать некоторый эффект при представлении информации о документе, например, в верхнем окне браузера или в ответе на запрос от поискового средства.

Синтаксис

```
<META NAME=имя элемента метаинформации CONTENT=содержимое информации> или  
<META HTTP-EQUIV=имя элемента метаинформации CONTENT=содержимое информации>
```

Примеры <META NAME=DESCRIPTION CONTENT= "An extensive guide to writing HTML 3.2 documents, with ../examples and practical advice.">

```
<META NAME=KEYWORDS CONTENT= "structural HTML, logical markup">
```

Тег META влияет на индексирование документа, когда он включается в базу данных поискового сервера.

TITLE - "внешний" заголовок (титул).

Используется для задания обязательного "внешнего" заголовка документа.

Титул может выводиться в окне заголовка программы просмотра; в списке результатов поиска, возвращаемых поисковым сервером; в горячем списке, определяемом пользователем; списке истории и т.д. Основной синтаксис <TITLE>последовательность символов</TITLE>

В TITLE можно использовать эскапе последовательности, например, < (для <) и ä (для д), но никакие теги HTML не разрешены, поэтому Вы не можете задавать в заголовке размеры шрифтов или выделения.

Примеры <TITLE>A study of population dynamics</TITLE> Примечания Написать хорошее заглавие - очень важно, так как списки результатов поиска, возвращаемые поисковым сервером, могут использовать его.

Комментарии

Файл HTML может содержать комментарии, дающие пояснения для человека, читающего HTML код. Комментарии не влияют каким-либо образом на представление документа, т.е. они игнорируются браузером. Вы можете начать комментарии с четырехсимвольной последовательности <!- (знак "меньше чем", восклицательный знак, два дефиса) и завершить его трехсимвольной последовательностью -> (два дефиса, знак "больше чем"). Например: <!- Написано Иваном Ивановым ->.

Оформление текста

**Элемент ** Элемент используется с целью выделения особым шрифтом слова или текста. Синтаксис: Текст

Элемент <CODE> Элемент <CODE> используется с целью дополнительного выделения фрагментов программного кода. По умолчанию он отображается телетайпным шрифтом. Данный элемент предпочтительнее, чем элемент <TT> (телетайпный шрифт). Поскольку расположение пробелов существенно для чтения программного кода, элемент <CODE> целесообразно употреблять в сочетании с элементом <PRE>. Синтаксис: <CODE> листинг кода </CODE>

Элемент <DFN> Элемент <DFN> используется с целью обозначения терминов и определений по типу словарей или глоссариев. Синтаксис: <DFN> Текст </DFN>

Элемент <CITE> Элемент <CITE> используется с целью обозначения источника информации ,из которого взята цитата. Синтаксис: <CITE> Текст </CITE>

**Элемент ** Элемент используется с целью выделения особым шрифтом слова или текста. Синтаксис: Текст

Элемент <I> Элемент <I> используется с целью выделения курсивным шрифтом слова или текста. Синтаксис: <I> Текст </I>

**Элемент ** Элемент используется с целью выделения полужирным шрифтом слова или текста. Синтаксис: Текст

Элемент <U> Элемент <U> используется с целью выделения подчеркиванием слова или текста. Синтаксис: <U> Текст </U>

Элемент <SUP> Элемент <SUP> используется с целью выделения надстрочных слова или текста. Синтаксис: ^{Текст}

Элемент <SUB> Элемент <SUB> используется с целью выделения подстрочных слова или текста. Синтаксис: _{Текст}

Элемент <BIG> Элемент <BIG> используется с целью выделения крупным шрифтом слова или текста относительно основного текста. Синтаксис: <BIG> Текст </BIG>

Элемент <SMALL> Элемент <SMALL> используется с целью выделения мелким шрифтом слова или текста относительно основного текста. Синтаксис: <SMALL> Текст </SMALL>

**Элемент ** Элемент используется с целью изменения выделения шрифтом слова или текста. С ним применяются два атрибута size и color. Некоторые браузеры поддерживают атрибут face, позволяющий задать любой из перечня шрифтов, если браузер не находит заданный шрифт - то используется шрифт, заданный по умолчанию. Синтаксис: Текст или Текст

Элемент <BASEFONT> Элемент <BASEFONT> используется как альтернатива атрибуту size элемента , он позволяет задать базовый размер шрифта во всем документе и не имеет конечного тега. По умолчанию значение его задается равным 3, значение size может выражаться так же и относительным размером, например, размер -1 означает размер на один меньший, чем по умолчанию. Синтаксис: <BASEFONT size=n>

Шесть уровней заголовков <Hn> Соответствующие каждому уровню гарнитура и размер шрифта зависят от браузера, стилю <H1> на-значается самый большой и самый жирный шрифт, а стилю <H6> назначается самый маленький и са-мый невзрачный шрифт. Элемент может иметь атрибут align, который указывает отступ left, centeR или Right. Синтаксис: <Hn align=отступ> Текст заголовка </Hn>

Разделительные линии <HR> Элемент <HR> используется для проведения горизонтальной черты в документе, он может иметь атрибуты : color, задающий цвет линии, size высота в пикселях Width ширина в пикселях или процентах от ширины экрана, align режим выравнивания, и не имеет конечного тега. Синтаксис: <HR align="centeR" size=n Width=n color="цвет">

Элемент <P> Этот элемент задает один из способов разбиения текста на абзацы. Он может иметь вложенный атрибут align, который указывает отступ left, center, right, justify. Каждый следующий абзац игнорирует, заданное для предыдущего абзаца значение align. Синтаксис: <P align=отступ> Текст абзаца </P>

**Элемент
** Этот элемент задает разрыв текста с переходом на новую строку. Он может иметь вложенный атрибут cleaR, который может принимать значения left, all или Right тем самым указывать обтекание текста во-круг плавающих изображений вставленных в текст нестандартным способом. Каждый следующий аб-зац игнорирует, заданное для предыдущего абзаца значение cleaR. Синтаксис: <BR cleaR=обтекание> Текст Может быть отменен тэгами <NOBR> и </NOBR>

Элемент <PRE> Весь текст, заключенный в тэги <PRE> и </PRE> будет визуализирован браузером точно так, как он визуализирован в исходном коде документа, кроме того текст выводится моноширинным шрифтом, что значительно упрощает задачу форматирования текста в колонки. Элемент поддерживается не всеми браузерами, он может иметь атрибут Width, который задает ширину отводимого пространства под текст в символах. Элемент сменил собой устаревшие элементы <XMP>, <LISTING> и <PLAINTEXT> Синтаксис: <PRE Width=число символов >...текст.. </PRE>

Элемент <DIV> Элемент <DIV> позволяет выделить в структуре документа несколько разделов. Он является блочным элементом, функционирующим во многом подобно элементу <P>. Если закрывающий тэг </P> опущен, то <DIV> эффективно заменяет его и начинает новый абзац. Он может иметь атрибут align, который указывает отступ left, centeR или Right. Каждый следующий раздел игнорирует, заданное для предыдущего раздела, значение align. Синтаксис: <DIV align=отступ> Текст раздела </DIV>

Элемент <ADDRESS> Элемент <ADDRESS> используется для оформления контактной информации текущего документа, будь то адрес электронной почты или полный почтовый адрес с номером телефона. Синтаксис: < ADDRESS>контактная информация </ADDRESS>

Элемент <BLOCKQUOTE> Элемент <BLOCKQUOTE> позволяет выделить объемный текст-цитату из общего текста. Синтаксис: <BLOCKQUOTE> Текст </BLOCKQUOTE>

Списки

**Элемент ** используется с целью задания нумерованных списков, имеет атрибуты type=1, или A, или a, или I, или i для задания вида нумерации и staRt для указания, с какого индекса начинается нумерация списка.

Элемент включает в себя дополнительный элемент , который задает элементы списка. Синтаксис: <OL type=1 start=1 > элемент списка элемент списка

Пример

1. элемент списка
2. элемент списка

**Элемент **, по сути, является аналогом без дополнительных элементов , он используется с целью задания нумерованных списков, имеет атрибут type=ciRcle,squaRe, или disc для зада-

ния вида маркера. Элемент включает в себя дополнительный элемент , который задает элементы списка. Синтаксис: <UL type=circle > элемент списка элемент списка

Пример:

o элемент списка

o элемент списка

Элемент <DL> используется с целью задания словарей, глоссариев и прочих перечней. Элемент <DL> включает в себя дополнительные элементы <DT> и <DD>, которые обозначают соответственно термин и определение. Синтаксис: <DL > <DT> термин 1 <DD>определение 1 <DT> термин 2 <DD>определение 2 </DL>

Пример:

термин 1 определение 1

термин 2 определение 2

Таблица основных тегов HTML-документа. Теги форматирования шрифта и абзаца

| Назначение | Вид тегов | Примечание |
|----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| Общая структура документа HTML | | |
| Тип документа | <HTML></HTML> | Начало и конец документа |
| Имя документа | <HEAD></HEAD> | Не отображается браузером |
| Заголовок | <TITLE></TITLE> | Содержимое строки заголовка окна браузера |
| Тело документа | <BODY></BODY> | Содержимое WEB-страницы |
| Структура содержания документа | | |
| Внутренние заголовки различного уровня | <H№> текст </H№> | Где № – номер уровня заголовка (от 1 до 6). Например, <H1>...</H1> - заголовок 1-го уровня. |
| Заголовок с выравниванием | <H№ ALIGN="LEFT CENTER RIGHT"> текст </H№> | LEFT - по левому краю, CENTER - по центру, RIGHT - по правому краю. |
| Форматирование абзацев | | |
| Создание абзаца (параграфа) | <P> текст </P> | Абзацы отделяются двойным межстрочным интервалом |
| Перевод строки внутри абзаца | | Одиночный тег |
| Выравнивание абзаца | <P ALIGN="LEFT">текст </P> <P ALIGN="CENTER">текст </P> <P ALIGN="RIGHT"> текст</P> <P ALIGN="JUSTIFY"> текст </P> | LEFT - по левому краю CENTER - по центру RIGHT - по правому краю JUSTIFY – по ширине |
| Разделительная горизонтальная линия между абзацами | <HR SIZE="?"> | Одиночный тег. «?» - толщина линии в пикселях. Толщину линии можно не указывать. |
| Форматирование шрифта | | |
| Жирный | текст | Жирный |
| Курсив | <I> текст </I> | <I> Курсив </I> |
| Подчеркнутый | <U> текст </U> | <U> Подчеркнутый </U> |
| Перечеркнутый | <S> текст </S> | <S> Перечеркнутый </S> |
| Увеличенный размер | <BIG> текст </BIG > | |
| Уменьшенный размер | <SMALL> текст </SMALL> | |
| Верхний индекс | ^{текст} | |

| | | |
|-------------------------|-------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| Нижний индекс | _{текст} | ^{Верхний индекс} _{Нижний индекс} |
| Размер шрифта | текст | ?- значения от 1 до 7 или относительное изменение (например, +2) |
| Базовый размер шрифта | <BASEFONT SIZE=«?»> | Одиночный тег ? – размер от 1 до 7; по умолчанию равен 3 и задается для всего документа в целом |
| Гарнитура шрифта | текст | Текст оформляется первым, установленным на компьютере шрифтом из списка названий |
| Цвет шрифта | текст | Цвет задается либо ключевым словом, либо шестнадцатеричным кодом с символом # RED –красный, #FF0000 – шестнадцатеричный код – красного цвета |
| Создание списков | | |
| Нумерованный | элементы списка | |
| Маркированный | элементы списка | Элемент списка 1 |
| Элемент списка | элементы списка | Элемент списка 2 Элемент списка 3 |

Таблица основных цветов

| Цвет | Color's name | Шестнадцатеричный код цвета | | |
|----------------|--------------|-----------------------------|-------|------|
| | | Red | Green | Blue |
| Черный | black | 00 | 00 | 00 |
| Темно-синий | navy | 00 | 00 | 80 |
| Голубой | blue | 00 | 00 | FF |
| Зеленый | green | 00 | 80 | 00 |
| Темно-зеленый | teal | 00 | 80 | 80 |
| Салатный | lime | 00 | FF | 00 |
| Бледно-голубой | aqua | 00 | FF | FF |
| Вишневый | maroon | 80 | 00 | 00 |
| Фиолетовый | purple | 80 | 00 | 80 |
| Оливковый | olive | 80 | 80 | 00 |
| Серый | gray | 80 | 80 | 80 |
| Светло-серый | silver | C0 | C0 | C0 |
| Красный | red | FF | 00 | 00 |
| Лиловый | fushsia | FF | 00 | FF |
| Желтый | yellow | FF | FF | 00 |
| Белый | white | FF | FF | FF |

Задание 1 Форматирование шрифта и абзаца

1. Создать файл с гипертекстовым документом:

- Запустить любой html редактор, ввести текст (произвольно)

например:

Чуть больше 50 лет назад в 5 тысячах км от Москвы рос молодой красивый город. В суровый край со всей земли собирались люди строить Братск. Для воплощения грандиозных замыслов требовался высококвалифицированный труд специалистов и уже в 1957 году под руководством Клавдии Георгиевны Поповой, супруги начальника строительства И.И. Наймушина, открылся первый учебно-консультационный пункт (УКП) Иркутского филиала Всесоюзного заочного инженерно-строительного института, который временно размещался в деревянном здании поселка Падун. А еще через два года пункт передается Иркутскому политехническому институту.

- Сохранить файл в рабочей папке с расширением **.htm**, например **name.htm** (где **name** – ваше имя или фамилия)
- Закрыть документ, найти его пиктограмму в окне рабочей папки, открыть файл и проанализировать, *с помощью какого приложения отображается файл* и как выглядит введенная фраза.

2. Ввести теги, определяющие структуру html-документа:

- Ввести приведенные ниже теги, в разделе заголовка документа (между тегами **<TITLE>** **</TITLE>**) указать свою фамилию.

```
<HTML>
<HEAD> <TITLE> ФИО </TITLE>
      </HEAD>
      <BODY>
```

введенная фраза (текст)

```
</HTML>
```

- **Сохранить** документ под тем же именем, обновить его отображение и проанализировать произошедшие изменения в отображении документа.

3. Отредактировать документ:

- Вызвать меню браузера **Вид/Просмотр HTML-кода** и добавить перед текстом **«История университета»**

Сохранить документ (но не закрывать) и обновить его просмотр в браузере.

- Используя одиночный тег **
, отредактировать документ так, чтобы слово **История располагалось с новой строки, а **университета** – в следующей строке. Просмотреть в браузере новый вариант.

4. Оформить фрагменты текста с помощью стилей **Заголовков**:

- Первую строку документа оформить стилем **Заголовок 1-го уровня** с помощью парного тега **<H1> ...</H1>**. Вторую строку оформить как **Заголовок 6-го уровня**, а третью как **Заголовок 4-го уровня**.
- Поменять стиль оформления первой строки на **Заголовок 2 уровня**, второй строки - на **Заголовок 5 уровня**, последней строки - на **Заголовок 3-го уровня**.

5. Выполнить форматирование шрифта:

- После строки **История университета** добавить еще одну строку текста

Нас утро встречает прохладой

- Оформить фразу по приведенному ниже образцу.

Нас *утро* встречает прохладой

В слове УТРО все буквы должны иметь **разные цвета**. В слове ПРОХЛАДОЙ оформить буквы ПРО – **красным** цветом, ОЙ – **синим**.

- Оформить строку История университета **курсивом**, размер шрифта задать относительным изменением. Использовать теги **** и **<I>**
- Просмотреть полученный документ в браузере.

6. Выполнить форматирование абзацев:

- Создать новый документ **2_name.htm**, сохранить его в той же рабочей папке.
- Ввести текст:

```
<HTML>
<HEAD> <TITLE> ФИО </TITLE>
      </HEAD>
      <BODY>
<BR> Монолог Гамлета
      </BODY>
</HTML>
```

- Выровнять текст **по центру**.
- Ввести текст:

Быть или не быть - вот в чем вопрос. Что благороднее: сносить удары неустовой судьбы - или против моря невзгод вооружиться, в бой вступить. И все покончить разом...

- Оформить выравнивание абзаца **по ширине**.
- Ограничить абзац горизонтальными разделительными линиями сверху и снизу, используя тег <HR>.
- Скопировать монолог и разбить его на абзацы. Выровнять **по центру**.

Быть или не быть - вот в чем вопрос.

*Что благороднее: сносить удары
Неустовой судьбы - или против моря
Невзгод вооружиться, в бой вступить
И все покончить разом...*

- Сохранить документ.
- Просмотреть документ в окне браузера, изменяя размер окна.

7. Выполнить оформление списков:

- Создать новый документ **3_name.htm**, сохранить его в той же рабочей папке
- Ввести текст:

```
<HTML>
<HEAD> <TITLE> ФИО</TITLE>
</HEAD>
<BODY>
</BODY>
</HTML>
```

- В файле создать все виды списков (содержание выбрать самостоятельно):

1. нумерованный
2. маркированный
3. многоуровневый
4. список определений

8. Предъявить результаты работы преподавателю в виде отчета.

Задание 2 Вставка в HTML-документ рисунков. Создание закладок и гиперссылок

Основные теги вставки рисунков, закладок и гиперссылок

| Вставка изображений | | |
|--------------------------------------------------------------------------|--------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| Вставка графического файла | | <i>Пример:</i> |
| Выравнивание картинки относительно текста | | |
| Вывод текста всплывающей подсказки при наведении курсора мыши на рисунок | | |
| Вставка ссылок | | |
| Ссылки на другую страницу | текст | Ссылка1 |
| Ссылка на закладку в другом документе | текст | На главную страницу |
| Ссылка на закладку в том же документе | текст | Ссылка2 |
| Определить закладку | текст | |
| Цвет фона, текста и ссылок | | |

| | | |
|------------------------|----------------------------------|---------------------------------|
| Фоновая картинка | <BODY BACKGROUND="файл рисунка"> | <BODY BACKGROUND="grafica.gif"> |
| Цвет фона | <BODY BGCOLOR="#\$\$\$\$\$"> | TEXT="black" (черный) |
| Цвет текста | <BODY TEXT="#\$\$\$\$\$"> | LINK="#FF0000" (красный) |
| Цвет ссылки | <BODY LINK="#\$\$\$\$\$"> | VLINK="#FFFF00" (желтый) |
| Цвет пройденной ссылки | <BODY VLINK="#\$\$\$\$\$"> | ALINK="#FFFFFF" (белый) |
| Цвет активной ссылки | <BODY ALINK="#\$\$\$\$\$"> | </BODY> |

- Скопировать из Интернета или какой-либо папки в личную папку 3 графических файла
- Вставка рисунков в документ.
 - Открыть в редакторе документ **2_name.htm**.
 - Вставить рисунок **1** в начало документа **2_name.htm**. Для вставки использовать тег **IMG** с параметрами **WIDTH** и **HEIGHT** для установки размеров рисунка 50 пикселей по горизонтали и по вертикали.
 - Сохранить документ под именем **4_name.htm**.
 - Просмотреть в браузере полученный результат.
 - Ввести в тег рисунка параметр **ALIGN** для выравнивания рисунка по правому краю. Просмотреть результат в браузере.
 - Вставить рисунок **2** в конец документа **4_name.htm**, подобрать тип выравнивания рисунка на свое усмотрение. Установить размер рисунка 100 пикселей по горизонтали и по вертикали. С помощью параметра **ALT** создать всплывающую подсказку «**Рисунок 2**», появляющуюся при наведении курсора мыши на рисунок.
 - Просмотреть в браузере полученный результат.
- Создание гиперссылок и закладок.
 - В документе **3_name.htm** закрепить гиперссылки за элементами списка:
 - 1-ая гиперссылка на документ **1_name.htm**.
 - 2-ая – на документ **1_name.htm**.
 - 3-ая – на документ **2_name.htm**.
 - Создать закладку в документе **1_name.htm** перед фразой «**Нас утро встречает прохладой**». Дать ей имя «**Morning**».
 - Изменить первую гиперссылку так, чтобы она указывала на закладку «**Morning**» в документе **1_name.htm**.
 - Создать закладку в начале текущего документа **3_name.htm**. Присвоить ей имя «**Hello**».
 - Изменить вторую гиперссылку, определив для неё переход в начало текущего документа на установленную закладку «**Hello**».
 - Проверить правильность переходов по всем гиперссылкам.
- Закрепить гиперссылки за графическими файлами:
 - Отредактировать тег вставки рисунка **1**, ввести в тег атрибут **ALT** для отображения текста подсказки «**Вернуться**». Просмотреть в браузере как реагирует рисунок на наведение курсора мыши.
 - Закрепить за рисунком **1** в документе **4_name.htm** гиперссылку на документ **3_name.htm**. Выполнить переход между документами.
- Предъявить результат преподавателю в виде отчета.

Задание 3 Создание и форматирование таблиц

Теги оформления таблиц

| | | |
|-----------------------------|-----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| Определить таблицу | <TABLE></TABLE> | Пример <TABLE border="1" align="center" width="50%"> <TR> <TH>Товар</TH> <TH>Цена</TH> </TR> <TR> <TD>Радиотелефон</TD> |
| Окантовка таблицы | <TABLE BORDER="?"></TABLE> | |
| Строка таблицы | <TR></TR> | |
| Выравнивание | <TR ALIGN=left right center middle bottom > | |
| Ячейка таблицы | <TD></TD> | |
| Выравнивание по горизонтали | <TD ALIGN=LEFT RIGHT CENTER> | |

| Выравнивание по вертикали | <TD VALIGN = TOP MIDDLE BOTTOM> | <TD>2000 </TD> </TR> </TABLE> | | | | |
|--------------------------------------------|----------------------------------|--------------------------------------------------------------------------------------------------------------------|-------|------|--------------|------|
| Установка ширины ячейки (в пикселях или %) | <TD WIDTH=«?»> | <table border="1"> <tr> <th>Товар</th> <th>Цена</th> </tr> <tr> <td>Радиотелефон</td> <td>2000</td> </tr> </table> | Товар | Цена | Радиотелефон | 2000 |
| Товар | Цена | | | | | |
| Радиотелефон | 2000 | | | | | |
| Заливка цветом ячейки | <TD BGCOLOR = «# цвет»> </TD> | <TD BGCOLOR = «#FF0000»> </TD> красный цвет | | | | |
| Заголовок столбца или строки | <TH>текст </TH> | Текст в ячейке выравнивается по центру, устанавливается жирный шрифт | | | | |

| | | |
|--|---|--|
| | | |
| | * | |
| | | |

1. Создать таблицу по приведенному образцу, сохранить документ под именем **tabl_name.htm**. Сверху над таблицей разместить заголовок **Таблица №1**

При отображении таблицы в браузере должны удовлетворяться следующие условия:

- таблица должна быть выровнена по центру и быть правильной (симметричной) формы;
- в центральной ячейке поместить символ * (звездочка), остальные ячейки должны быть пустыми.

Примечание. Для отображения пустых ячеек в них нужно поместить символьный примитив пробела ** **;

2. В этом же документе создать копию таблицы №1, ввести заголовок **Таблица №2** и модифицировать ее:

- В центральной ячейке разместить рисунок **3**.
- «Раскрасить» все остальные ячейки в различные цвета.

3. Создать еще одну копию таблицы – **Таблица №3** и отредактировать теги таблицы так, чтобы она соответствовала приведенному ниже образцу.

| | | |
|--|-------------------------------------------------------------------------------------|--|
| | | |
| |  | |
| | | |

Примечание. Для объединения ячеек в тегах <TD> необходимо использовать параметры **colspan=** и **rowspan=**

4. Создать новый HTML-документ - **rasp_name.htm** с расписанием занятий.

- Документ должен начинаться заголовком **Расписание занятий гр. История на весенний семестр 20 г.**

- Первая строка таблицы должна быть оформлена как заголовки полей (с использованием тегов <TH>).

- Таблица по ширине должна занимать полный размер окна. Ширину отдельных столбцов задать в относительных единицах (в %), с тем, чтобы при изменении ширины окна пропорции таблицы сохранялись.

| День недели | Время | Предмет | Преподаватель | Аудитория |
|-------------|-------------|---------|---------------|-----------|
| Понедельник | 8:15-09:50 | | | |
| | 10:00-11:35 | | | |
| | 11:45-13:20 | | | |
| | 13:30-15:05 | | | |
| | 15:15-16:50 | | | |
| Вторник | 8:15-09:50 | | | |
| | 10:00-11:35 | | | |
| | 11:45-13:20 | | | |
| ... | ... | ... | ... | ... |

- Просмотреть созданный документ в браузере при различных размерах окна и различных настройках размера шрифта.
- 5. Сохранить файл с расписанием под именем **rasp_menu_name.htm** и модифицировать его.
- 6. После заголовка создать таблицу, состоящую из одной строки меню с названиями дней недели.

Расписание

| | | | | | |
|---------------|----------------|--------------|----------------|--------------|--------------|
| <u>Понед.</u> | <u>Вторник</u> | <u>Среда</u> | <u>Четверг</u> | <u>Пятн.</u> | <u>Субб.</u> |
|---------------|----------------|--------------|----------------|--------------|--------------|

7. В таблице с расписанием установить закладки на названия дней недели.
8. В таблице меню создать гиперссылки на соответствующие дни недели.
9. Выполнить цветовое оформление каждой ячейки меню.
10. Проверить правильность выполнения переходов по гиперссылкам.
11. Создать группу web-страниц, объединенных меню:
 - На рабочем диске создать папку **My_raspisanie** для размещения файлов расписания.
 - Поместить расписание на каждый день недели и таблицу с меню в отдельные файлы. Имена файлов: **menu.htm** – для главной страницы, названия дней недели – для остальных. Все документы разместить в папке **My_raspisanie**.
 - Отредактировать гиперссылки меню так, чтобы по ним выполнялись переходы на соответствующий документ.
 - В конце каждого файла с расписанием на день организовать гиперссылку для возврата в главный документ с меню.
 - Оформить фон каждого дня недели собственным цветом, совпадающим с цветом ячейки таблицы меню.
12. Предъявить результат преподавателю в виде отчета.

Порядок выполнения:

Соответствует пункту 1-3 задания.

Форма отчетности:

Отчет по практическому заданию, скрепленный титульным листом сдаётся в печатном виде. В отчёте должны присутствовать:

1. Цель работы.
2. Задание.
3. Поэтапное выполнения всех заданий ВИЗ.
4. Заключение (вывод).

Задания для самостоятельной работы:

Предусмотрены ВИЗ обучающегося.

Рекомендации по выполнению заданий и подготовке к практическому занятию

При подготовке и выполнении практического задания рекомендуется использовать материал лекций соответствующих разделов и литературу, предложенную для изучения данной дисциплины.

Основная литература

1. Ллойд, Й. Создай свой веб-сайт с помощью HTML и CSS : учебное пособие / Й. Ллойд; Пер. с англ. - Санкт-Петербург : Питер, 2013. - 416 с.

Дополнительная литература

2. Кобзов, А. Ю. Программные средства разработки Web- страниц и презентаций : методические указания к выполнению лабораторных работ / А. Ю. Кобзов, А. В. Кобзова. - Братск : БрГУ, 2014. - 43 с.

Контрольные вопросы для самопроверки

1. Что представляет собой *web*-страница на языке *HTML*?
2. Что такое теги? Какие бывают теги?
3. Что такое исходный код?
4. Как обозначают начало и конец тега?
5. Что такое заголовок документа? Какими тегами он выделяется?
6. Что такое тело документа? Какими тегами оно выделяется?
7. Какими буквами следует писать теги?

8. Назовите основные атрибуты тега `<BODY> ... </BODY>`.
9. Как выделить текст полужирным шрифтом, курсивом?
10. Как выполнить мигание текста?
11. Какие теги изменяют размер шрифта?
12. Как сделать горизонтальную линию?
13. Как сделать отступ от линии вниз?
14. Как изменить цвет линии? Как указать параметры линии?
15. Какие параметры тега `<BODY>` влияют на изменение цвета?
16. Как установить цвет фона?
17. Что такое таблица цветов? Для чего она служит?
18. Какой тег позволяет перейти на новую строку?
19. Как оформить маркированные или нумерованные списки?
20. Как оформить вложенные списки?
21. Какими тегами определяется цвет гипертекстовой связи?
22. Какие объекты на *HTML*-странице могут иметь ссылку?
23. Какие теги используются для вставки гиперссылки?
24. Какой параметр определяет адрес ссылки?
25. Что нужно сделать, чтобы подготовить графический файл для последующей вставки изображения в *HTML*-страницу?
26. Назовите параметры графических изображений?
27. Назовите атрибуты тега ``?
28. Какими тегами вставляется таблица в *HTML*-документ? Какими параметрами она обладает?
29. Какие элементы могут располагаться между тегами вставки таблицы?
30. Какие теги используются для создания заголовков таблицы?
31. Какие теги позволяют объединять ячейки таблицы?
32. Какой тег позволяет редактировать обрамление таблицы?
33. Как задать ширину таблицы?

Практическое занятие № 2. Создание Web-страницы с фреймами

Цель работы:

научиться создавать web-страницы с фреймами

Фреймы – это прямоугольные области экрана, каждая из которых содержит свой собственный HTML-документ. Фреймы прекрасно подходят для оформления следующих документов:

- *Оглавление*. Если вы поместите на Web-страницу оглавление в виде вертикального столбца, то пользователь сможет обратиться к нему в любой момент, и ему не нужно будет постоянно щелкать на кнопке возврата к предыдущей странице.

Поскольку, находясь во фрейме, оглавление всегда будет под рукой, пользователю нужно будет просто выбрать другой его пункт и сразу же получить нужную информацию.

- *Неподвижные элементы интерфейса*. Можно зафиксировать на экране какое-то графическое изображение, например, логотип фирмы, в то время как остальная часть страницы будет прокручиваться в другом фрейме.

- *Формы и результаты*. Можно создать форму в одном фрейме, а в другом отобразить результаты запроса.

Тег `<FRAMESET>`

Тег `<FRAMESET>` - заменяет тег `<BODY>` и используется для разделения экрана.

Имеет закрывающий тег.

Атрибуты

- **COLS**. Делит экран по вертикали. Принимает значения в пикселях, в процентах или просто *. Значение * говорит о том, что конкретный столбец будет занимать всю остальную часть экрана

- **ROWS**. Делит экран по горизонтали. Принимает значения в пикселях, в процентах или просто *. Значение * говорит о том, что конкретная строка будет занимать всю остальную часть экрана

- **FRAMEBORDER**. Определяет наличие рамок, т.е. границ фреймов. Принимает значения "yes" или "no".

- **BORDER**. Определяет ширину рамки в пикселях.

- **BORDERCOLOR**. Определяет цвет рамок. По умолчанию (если используется стандартная цветовая схема системы Windows) границы фреймов имеют тусклый, серый оттенок. Но при желании можно выбрать любой другой цвет. При определении цвета можно выбрать как его название, так и числовой эквивалент в системе RGB. Например, BLUE или #0000FF.

В отдельном теге `<FRAMESET>` имеет смысл использовать только один из атрибутов –COLS или ROWS. Это означает, что фреймовая структура будет состоять или только из столбцов, или только из строк. Чтобы

создать строки внутри столбцов или столбцы внутри строк, вам потребуется вложенные контейнеры <FRAMESET>...</FRAMESET>.

Пример

```
<FRAMESET COLS="25%,75%" FRAMEBORDER="YES" BORDER="2">  
<FRAMESET ROWS="50%,50%" FRAMEBORDER="YES" BORDER="2">  
</FRAMESET>  
<FRAMESET ROWS="10%,90%" FRAMEBORDER="YES" BORDER="2">  
</FRAMESET>  
</FRAMESET>
```

Результат:

| | |
|---|---|
| 1 | 3 |
| 2 | 4 |

Используя сразу оба атрибута – и COLS, и ROWS в одном теге <FRAMESET>, вы не сможете получить такое разбиение экрана.

Пример

```
<FRAMESET COLS="25%,75%" ROWS="50%,50%">  
</FRAMESET>
```

Результат:

| | |
|---|---|
| 1 | 2 |
| 3 | 4 |

Обратите внимание также на различие в порядке нумерации фреймов.

Тег <FRAME>

Тег <FRAME>, помещенный в контейнер <FRAMESET>...</FRAMESET>, определяет, что именно должно отображаться в конкретном фрейме. Не имеет закрывающего тега.

Атрибуты

- **SRC**. Определяет URL, связанный с конкретным фреймом.
- **MARGINWIDTH**. Определяет расстояние между содержимым фрейма и его границами справа и слева.
- **MARGINHEIGHT**. Определяет расстояние между содержимым фрейма и его границами сверху и снизу.
- **SCROLLING**. Определяет присутствие в окне фрейма полосы прокрутки. Может принимать значения "yes", "no" и "auto"(по умолчанию).

NORESIZE. Определяет границы фрейма как "жестко закрепленные" и не позволяет пользователю изменить размеры окна фрейма. Причем фиксируются размеры всех фреймов, имеющих с закрепленным общую границу.

· **FRAMEBORDER, BORDER, BORDERCOLOR**. Эти атрибуты связаны с рамками и аналогичны таким же в теге <FRAMESET>.

· **NAME**. Определяет имя окна фрейма. Единственный атрибут, не влияющий на внешний вид фрейма. Позволяет контролировать процесс загрузки фреймов. Если окно фрейма имеет уникальное имя, то к нему можно непосредственно обратиться из других фреймов. Присваивая имена окнам фреймов, надо помнить об одном ограничении: имя не должно начинаться с символа подчеркивания "_", иначе оно будет игнорироваться. Так как с этого символа начинаются некоторые служебные имена.

Пример

```
<FRAMESET COLS="25%,75%" FRAMEBORDER="yes" BORDER="5"  
BORDERCOLOR="#008800">  
<FRAME src="left.htm" MARGINWIDTH="20" MARGINHEIGHT="20">  
<FRAME src="right.htm" NAME="main_window" MARGINWIDTH="0" SCROLLING="no"  
NORESIZE>  
</FRAMESET>
```

Использование тега **A** для загрузки во фрейм

С помощью тега гиперссылки **<A>** можно загрузить документ в определенный фрейм.

Атрибуты

· **HREF**. Определяет URL или имя нового документа, который вы хотите загрузить в определенное окно фрейма.

· **TARGET**. Определяет имя фрейма, в который будет загружен новый документ.

Это имя должно быть присвоено фрейму атрибутом **NAME** в теге **<FRAME>**.

В следующем примере экран разбивается на две вертикальные области. В меньшей левой части находится оглавление, а в правую будут загружаться все остальные HTML-документы. В файле оглавления находятся простые текстовые гиперссылки на соответствующие разделы. Чтобы все работало, необходимо создать HTML-документы в файлах `cosm.htm`, `eat.htm`, `perf.htm`, `massage.htm` и `manic.htm`.

Пример

```
<!-- Файл фреймовой структуры frame.htm -->
<HTML>
<FRAMESET COLS="25%,75%" FRAMEBORDER="yes"
BORDER="5" >
<FRAME src="left.htm"
NAME="toc_window">
<FRAME src="right.htm"
NAME="main_window">
</FRAMESET>
</HTML>
<!-- Файл оглавления left.htm -->
<HTML>
<H3>Оглавление</H3>
<UL>
<LI>Товары
<UL>
<LI><A href="cosm.htm"
TARGET="main_window">Косметика</A>
<LI><A href="eat.htm"
TARGET="main_window">Пищевые добавки</A>
<LI><A href="perf.htm"
TARGET="main_window">Парфюмерия</A>
</UL>
<LI>Услуги
<UL>
<LI><A href="massage.htm"
TARGET="main_window">Массаж</A>
<LI><A href="manic.htm"
TARGET="main_window">Маникюр</A>
</UL>
</UL>
</BODY>
</HTML>
<!-- Файл заставки основного окна right.htm -->
<HTML>
<BODY>
<H1>Здравствуйте!</H1>
</BODY>
</HTML>
```

Тег **<BASE>**

Тег **<BASE>** избавит вас от необходимости повторять атрибут **TARGET** в каждой отдельной гиперссылки, если все они указывают на одно и то же окно фрейма. Для этого надо поместить тег **<BASE>** с атрибутом **TARGET** в контейнер **<HEAD>...</HEAD>**.

Использование тега **<BASE>** даст возможность уменьшить размер файла оглавления `left.htm` предыдущего примера.

Пример

```
<!-- Файл оглавления left.htm -->
<HTML>
<HEAD>
<BASE TARGET="main_window">
</HEAD>
<H3>Оглавление</H3>
<UL>
```

```

<LI>Товары
<UL>
<LI><A href="cosm.htm">Косметика</A>
<LI><A href="eat.htm">Пищевые добавки</A>
<LI><A href="perf.htm">Парфюмерия</A>
</UL>
<LI>Услуги
<UL>
<LI><A href="massage.htm">Массаж</A>
<LI><A href="manic.htm">Маникюр</A>
</UL>
</UL>
</BODY>
</HTML>

```

```

<!-- Файл оглавления left.htm -->
<HTML>
<HEAD>
<BASE TARGET="wares_window">
</HEAD>
<H3>Оглавление</H3>
<UL>
<LI>Товары
<UL>
<LI><A href="cosm.htm">Косметика</A>
<LI><A href="eat.htm">Пищевые добавки</A>
<LI><A href="perf.htm">Парфюмерия</A>
</UL>
<LI>Услуги
<UL>
<LI><A href="massage.htm"
TARGET="service_window">Массаж</A>
<LI><A href="manic.htm"
TARGET="service_window">Маникюр</A>
</UL>
</UL>
</BODY>
</HTML>

```

```

<!-- Файл заставки окна товаров wares.htm -->
<HTML>
<BODY>
<H1>Здесь товары!</H1>
</BODY>
</HTML>

```

```

<!-- Файл заставки окна услуг service.htm -->
<HTML>
<BODY>
<H1>Здесь услуги!</H1>
</BODY>
</HTML>

```

Специальные эффекты, получаемые с помощью атрибута TARGET

Как уже упоминалось, имена фреймов не могут начинаться с символа подчеркивания, так как с него начинаются служебные имена, передающие браузеру особую информацию.

Ниже перечислены специальные имена, каждое из которых используется для достижения определенного эффекта.

- **TARGET="_blank"**. Документ загружается в новое окно браузера.
- **TARGET="_self"**. Документ загружается в текущее окно.
- **TARGET="_parent"**. Документ загружается в родительский фрейм. Обычно это фрейм, стоящий в контейнере <FRAMESET>...</FRAMESET> перед текущим. Если родительского фрейма нет, то значение "_parent" становится тождественным "_self".
- **TARGET="_top"**. Документ загружается в самый верхний фрейм.

В основном эти служебные имена предназначены для того, чтобы тем или иным способом "вырваться" из текущей фреймовой структуры. Поэкспериментируйте с ними, чтобы понять, как можно загружать документы в различные окна.

Тег <IFRAME>

В HTML 4.0 появились плавающие (или встроенные) фреймы.

С помощью тега <IFRAME> можно поместить один фрейм в обычный HTML-документ. Причем контейнер <FRAMESET>...</FRAMESET> здесь не нужен. Закрывающий тег </IFRAME> обязателен!

Атрибуты

ALIGN. Определяет вид выравнивания. Принимает значения "left", "center", "right".

HEIGHT. Определяет высоту фрейма в пикселях

WIDTH. Определяет ширину фрейма в пикселях.

SRC, MARGINWIDTH, MARGINHEIGHT, SCROLLING, FRAMEBORDER, BORDER, BORDERCOLOR, NAME. Аналогичны рассмотренным ранее.

vspace. Устанавливает поля сверху и снизу с наружи от iframe

hspace. Устанавливает поля сбоков с наружи от iframe

marginwidth. Определяет величину отступов по левому и правому краям внутри iframe-a; должно быть равно или больше 1.

marginheight. Определяет величину отступов по верхнему и нижнему краям внутри iframe-a; должно быть равно или больше 1.

scrolling. Указывает будет-ли выводиться линейка прокрутки в iframe; значение value может быть "yes," "no," или "auto". Значение по умолчанию для обычных документов - auto.

title. Текст всплывающей подсказки.

```
<IFRAME src="banner.html" width="468" height="60"
```

```
hspace="10" vspace="10" align="center" src="aa.htm">
```

Ваш браузер не поддерживает плавающие фреймы!

```
</IFRAME>
```

Тег <NOFRAMES>

Определяет, что будет показано в окне браузера если он не поддерживает фреймы.

```
<NOFRAMES>
```

Ваш браузер не поддерживает плавающие фреймы!

```
</NOFRAMES>
```

Задание:

1. Создайте четыре файла: a.html, b.html, c.html, d.html, (тематику оформить самостоятельно) которые будут в дальнейшем заполнять поля фреймов.

Файл a.html

```
<HTML>
```

```
<HEAD><TITLE>Страничка A</TITLE></HEAD>
```

```
<BODY bgcolor=red text=white>
```

```
<p>Страничка A</p>
```

```
</BODY>
```

```
</HTML>
```

Файл b.html

```
<HTML>
```

```
<HEAD><TITLE>Страничка B</TITLE></HEAD>
```

```
<BODY bgcolor=yellow text=black>
```

```
<p>Страничка B</p>
```

```
</BODY>
```

```
</HTML>
```

Файл c.html

```
<HTML>
```

```
<HEAD><TITLE>Страничка C</TITLE></HEAD>
```

```
<BODY bgcolor=green text=white>
```

```
<p>Страничка C</p>
```

```
</BODY>
```

```
</HTML>
```

Файл d.html

```
<HTML>
```

```
<HEAD><TITLE>Страничка D</TITLE></HEAD>
```

```
<BODY bgcolor=blue text=white>
```

```
<p>Страничка D</p>
```



```
</BODY>
</HTML>
```

2. Создайте файлы Web-страниц с различными вариантами расположения фреймов.

Файл index-2-frames.html

```
<HTML>
<HEAD>
<TITLE>ФРЕЙМЫ-2</TITLE></HEAD>
<!-- Пример создания двух вертикальных фреймов одинаковой ширины -->
<FRAMESET cols="50%,50%">
<FRAME src="a.html">
<FRAME src="b.html">
</FRAMESET>
</HTML>
```

Файл index-4-frames-cols.html

```
<HTML>
<HEAD>
<TITLE>ФРЕЙМЫ-4</TITLE></HEAD>
<!-- Пример создания четырех вертикальных фреймов. Из них 3 фрейма одинаковой ширины, а один – занимает всю оставшуюся площадь экрана -->
<FRAMESET cols="20%,20%,20%,*">
<FRAME src="a.html">
<FRAME src="b.html">
<FRAME src="c.html">
<FRAME src="d.html"> </FRAMESET></HTML>
```

Файл index-4-frames-rows.html

```
<HTML>
<HEAD>
<TITLE>ФРЕЙМЫ-4</TITLE></HEAD>
<FRAMESET rows="20%,20%,20%,*">
<!-- Пример создания четырех горизонтальных фреймов. Из них 3 фрейма одинаковой ширины, а один – занимает всю оставшуюся площадь экрана -->
<FRAME src="a.html">
<FRAME src="b.html">
<FRAME src="c.html">
<FRAME src="d.html">
</FRAMESET>
</HTML>
```

Файл index-3-frames-cols-coot.html

```
<HTML>
<HEAD>
<TITLE>ФРЕЙМЫ-4</TITLE></HEAD>
<!--Пример создания фреймов различной ширины путем задания соотношения между ними-->
<FRAMESET cols="100,*,2*">
```

Форма отчетности:

Отчет по практическому заданию, скрепленный титульным листом сдаётся в печатном виде. В отчёте должны присутствовать:

1. Цель работы.
2. Задание.
3. Поэтапное выполнения всех заданий ВИЗ.
4. Заключение (вывод).

Задания для самостоятельной работы:

Предусмотрены ВИЗ обучающегося.

Рекомендации по выполнению заданий и подготовке к практическому занятию

При подготовке и выполнении практического задания рекомендуется использовать материал лекций соответствующих разделов и литературу, предложенную для изучения данной дисциплины.

Основная литература

1. Ллойд, Й. Создай свой веб-сайт с помощью HTML и CSS : учебное пособие / Й. Ллойд; Пер. с англ. - Санкт-Петербург : Питер, 2013. - 416 с.

Дополнительная литература

2. Кобзов, А. Ю. Программные средства разработки Web- страниц и презентаций : методические указания к выполнению лабораторных работ / А. Ю. Кобзов, А. В. Кобзова. - Братск : БрГУ, 2014. - 43 с.

Контрольные вопросы для самопроверки

1. Какие теги используются для создания фреймов?
2. Как делится экран между фреймами?
3. Какой параметр используется для деления на фреймы?
4. Как разместить изображение в одном из фреймов?
5. Как сделать полосу прокрутки?
6. Какие атрибуты задают отступы внутри фрейма?
7. Как сделать гиперссылку из фрейма?
8. Как открыть страничку без фреймов?

Практическое занятие № 3. Таблицы стилей или CSS

Цель работы: получение практических навыков по созданию и применению таблиц стилей CSS

Таблицы стилей (style sheets) представляют собой абстракцию, в которой стиль документа определяется отдельно от его содержания. Термин "таблицы стилей" прочно вошел в русскую речь, хотя в некоторых книгах вы можете встретить и более точный перевод - "листы стилей".

По методам добавления стилей в документ различают три вида стилей.

Внутренние стили

Определяются атрибутом непосредственно в элементе. Мало отличается от традиционного HTML. Внешний вид документа трудно изменить.

Пример

```
<!-- каждому параграфу – свой цвет -->
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
</head>
<P style="color:blue"> Таблицы стилей (style sheets) представляют собой абстракцию, в которой стиль документа определяется отдельно от его содержания.</P>
<P style="color:red"> Термин "таблицы стилей" прочно вошел в русскую речь, хотя в некоторых книгах вы можете встретить и более точный перевод -"листы стилей".
</P>
</html>
```

Глобальные стили

Определяются тегом **STYLE** с атрибутом **TYPE="text/css"** в контейнере **<HEAD>...</HEAD>**, являются универсальным средством, позволяющим не только оперативно изменять внешний вид страницы, но и бороться с перегруженностью документа оформительскими тегами.

Пример

```
<!-- все параграфы лилового цвета -->
<html>
<head>
<style type="text/css">
<!--
P {color:#FF00FF}
-->
</style>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
</head>
<P> Таблицы стилей (style sheets) представляют собой абстракцию, в которой стиль документа определяется отдельно от его содержания.</P>
<P> Термин "таблицы стилей" прочно вошел в русскую речь, хотя в некоторых книгах вы можете встретить и более точный перевод - "листы стилей". </P>
</html>
```

Обратите внимание на то, что описание стиля заключено в тэги комментариев. Это сделано для того, чтобы браузеры, не понимающие, о чем идет речь, не вываливали описание стилей на страницу, как обычный текст

Связанные стили

Определяются во внешнем файле. Позволяют всем страницам, принадлежащим одному серверу выглядеть единообразно. Для связи с файлом, на котором описана таблица стилей, используется тэг <LINK> с атрибутом REL="stylesheet".

Другие атрибуты

TITLE. Задаёт заголовок страницы, на которой с помощью тэга <STYLE>, определена таблица стилей.

HREF. Ссылка на страницу с таблицей стилей или имя файла.

TYPE. Тип файла, с которым нужно установить связь.

Пример

```
<LINK TITLE="mystyle" REL="stylesheet" HREF="style.htm" TYPE="text/css">
```

Каскадные таблицы стилей (CSS)

Каскадные таблицы стилей CSS (Cascading Style Sheets) - первый стандарт стилей, объявленный консорциумом W3C. Термин *каскадные* указывает на возможность слияния различных видов стилей и на наследование стилей внутренними тэгами от внешних.

CSS - это язык, содержащий набор свойств для определения внешнего вида документа. Спецификация CSS определяет свойства и описательный язык для установления связи с HTML-элементами. Здесь можно найти самые разнообразные свойства стилей. Однако, пока еще браузеры не поддерживают все возможности CSS.

Можно назначать любые стили, потому что неподдерживаемые стили будут просто игнорироваться.

Наиболее популярные свойства стилей BACKGROUND. Фон элемента. Задаётся одним или двумя цветами (т.е. их смесью) или именем графического файла. Возможные значения: red, black/white, fill2.gif ().

BORDER. Рамка вокруг элемента. Возможные значения: solid 0.5mm (есть рамка шириной 0.5 миллиметра), none (нет рамки).

BORDER-COLOR. Цвет рамки. Возможные значения: blue, #993300.

COLOR. Цвет элемента.

CURSOR. Тип курсора. Возможные значения (их можно видеть, перемещая курсор по первому столбцу):

| | |
|-------------|-----------------------------------------------------------------------------------|
| auto | Браузер определяет какой курсор показывать в зависимости от контекста |
| default | Курсор зависит от платформы. Обычно стрелка |
| crosshair | Обычно простой крестик |
| hand | Рука IE |
| pointer | Рука NS6/ IE6 |
| move | Четырёхсторонняя стрелка. Обычно используется при указании на перемещаемый объект |
| text | Изменяемый текст. Обычно I-вида |
| help | Стрелка с вопросом. Используется при указании на подсказку |
| wait | Курсор показывает, что программа busy и следует подождать. Обычно песочные часы |
| progress | Курсор показывает, что идет процесс обработки. Только IE6 |
| not-allowed | Курсор показывает, что объект или действие недоступно. Только IE6 |
| n-resize | Стрелка, указывающая в заданном направлении (north, north-west, west, south- |
| nw-resize | west, ...) |
| w-resize | |
| sw-resize | |
| s-resize | |
| se-resize | |
| e-resize | |
| ne-resize | |

Точный вид курсора зависит от того, какие курсоры установлены на машине пользователя.

Примеры использования Style Cursor

Пример 1:

```
<style type="text/css">
body{
cursor: url(mycursor.cur)
}
</style>
```

В этом примере для всей WEB страницы задан курсор из файла с именем

mycursor.cur.

Пример 2:

```
<div style="cursor: move; width: 200px; height: 200px"></div>  
<a href="help.htm" style="cursor: help;">Help</a>
```

В этом примере внутри DIV форма курсора "move", а на ссылке форма курсора "help".

Пример 3:

```
<TD onmouseover="this.style.cursor='wait'">se-resize</TD>
```

В этом примере при наведении на ячейку таблицы изменяется стиль курсора на "wait".

Пример 4:

```
<body onSelectstart="this.style.cursor='not-allowed'; return false;"  
onmouseup="this.style.cursor='default'">
```

Пример запрета выделения на странице. Если вы начнете выделять, то Курсор изменится на запрещающий и выделения не будет производиться

Изменение стиля шрифта с помощью Style Font

FONT-FAMILY. Тип шрифта. Возможные значения: Arial (без засечек), **CourierNew** (моноширинный), Times New Roman (с засечками).

FONT-SIZE. Размер шрифта. Возможные значения: 14pt, 120%. Единицами измерения для этого и других размерных свойств могут быть: pt (пункты), px (пиксели), mm (миллиметры), cm (сантиметры), in (дюймы), % (проценты).

FONT-WEIGHT. Толщина шрифта. Возможные значения: light (легкий), medium (средний), bold (жирный).

FONT-STYLE. Стиль шрифта. Возможные значения: italic (наклонный), normal (обычный).

FONT. Полная характеристика шрифта. Сочетание четырех предыдущих свойств.

LIST-STYLE-IMAGE. Символ маркера в маркированном списке. Задается как gif-файл. Может определяться как для всего списка, так и для его отдельных элементов.

Пример

```
<html>  
<head>  
<style type="text/css">  
<!--  
ul {list-style-image:URL(fill1.gif)}  
-->  
</style>  
<meta http-equiv="Content-Type" content="text/html;charset=windows-1251">  
</head>  
<ul>Напитки  
<li>Молоко</li>  
<li>Чай</li>  
<li>Сок</li>  
<li>Кофе</li>  
</ul>  
</html>
```

LETTER-SPACING. Межбуквенный интервал. Возможные значения: **2px** (интервал в 2 пиксела), **-4pt** (интервал, уменьшенный на 0.4 пункта).

LINE-HEIGHT. Межстрочный интервал. Возможные значения: **14pt, 50%**.

MARGIN-BOTTOM, MARGIN-LEFT, MARGIN-RIGHT, MARGIN-TOP.

Отступы соответственно снизу, слева, справа, сверху элемента. На многие элементы не действуют. Действуют на **DIV**. Возможные значения: **150pt, 0.5in**.

MARGIN. Отступы вокруг элемента.

PADDING-BOTTOM, PADDING-LEFT, PADDING-RIGHT, PADDING-TOP.

Отступы от рамки соответственно снизу, слева, справа, сверху.

PADDING. Все отступы от рамки до элемента.

TEXT-ALIGN. Выравнивание текста по горизонтали. Возможные значения: **left** (по левому краю), **center** (по центру), **right** (по правому краю), **justify** (по ширине).

TEXT-DECORATION. Оформление текста. Возможные значения: **underline** (подчеркивание), **line through** (перечеркивание), **none** (без оформления).

TEXT-INDENT. Отступ первой строки текста. Чаще всего используется для создания параграфов с табулированной первой строкой. Возможные значения:

50pt, 1cm.

WIDTH. Ширина поля, предоставленного элементу. Возможные значения: **100pt, 5cm.**

Использование атрибутов CLASS и ID

Стиль в CSS может быть задан для определенного элемента, класса или идентификатора.

Пример

```
<html>
<head>
<style type="text/css">
<!--
H1 {color:red;text-align:center}
.myclass {font-style:italic}
#myid {background:yellow}
-->
</style>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
</head>
<H1>Seasons</H1>
<H2 class="myclass">Lesson Seventy-Nine</H2>
<EM id="myid">LET US LEARN</EM>
<P class="myclass">Learn to pronounce and use the new words.</P>
</html>
```

Кроме того можно использовать и **контекстуальные селекторы**, определяющие иерархию контейнеров, с которыми должен быть связан стиль.

Примеры

```
<!-- все элементы EM внутри элемента P будут на желтом фоне -->
P EM {background:yellow}
<!-- все элементы класса pilot внутри элемента LI класса man,
которые находятся внутри элемента UL будут голубого цвета -->
UL LI.man .pilot {color:blue}
```

Если несколько элементов будут иметь общий стиль, то соответствующий им селекторы можно перечислить в таблице стилей через запятую.

Пример

```
<html>
<head>
<style type="text/css">
<!--
H1,H2,H3,H4 {color:red;text-align:center}
-->
</style>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
</head>
<H1>Seasons</H1>
<H2>Lesson Seventy-Nine</H2>
<H3>LET US LEARN</H3>
<H4>1. Learn to pronounce and use the new words.</H4>
</html>
```

Оформление ссылок (Псевдоклассы)

Псевдоклассами называются элементы одного типа, удовлетворяющие определенному критерию. Например, посещенные элементы **A** (ссылки) составляют псевдокласс **visited**. В настоящее время поддерживаются только псевдоклассы ссылок. Но и это уже кое-что. С помощью таблицы стилей, где псевдокласс следует отделять двоеточием (:), можно изменить оформление активной (**active**), посещенной (**visited**), непосещенной (**link**) ссылки, а также ссылки, на которую наведена мышь (**hover**).

Пример

```
A:link {text-decoration:none}
A:visited {text-decoration:none}
A:active {text-decoration:none}
A:hover {text-decoration:underline}
```

Использование тэгов и <DIV>

Стиль можно установить почти для всех элементов HTML, но два тэга и <DIV> особенно удобны тем, что они не накладывают на стиль никакого собственного типа представления. При использовании атрибутов **CLASS** и **ID** эти тэги вместе с CSS практически неограниченно расширяют HTML.

... и <DIV>...</DIV>

являются простыми контейнерами для HTML-элементов. Они позволяют применять атрибуты форматирования к конкретным частям документа. Друг от друга они отличаются лишь тем, что браузеры обычно помещают символ перевода строки до и после HTML-элементов, заключенных в контейнер <DIV>...</DIV>.

Пример

```

<html>
<head>
<style type="text/css">
<!--
.ital { font-size:120%;font-style:italic;color:blue }
.bold { font-size:120%;font-weight:bold;color:blue }
.end { font-weight:bold;color:blue }
-->
</style>
<meta http-equiv="Content-Type"content="text/html; charset=windows-1251">
</head>
<pre>
<span class=ital>Правильные</span> глаголы образуют прошедшее время путем добавления окончания
<span class=bold>-ed</span> или <span class=bold>-d</span>, которое читается как:
[t](help<span class=end>ed</span>), [d](open<span class=end>ed</span>),
[id](want<span class=end>ed</span>).
</html>

```

В контейнер **<DIV>...</DIV>** можно включать другие тэги, исключая **<HEAD>**, **<BODY>** и тэги, связанные с фреймами (кроме **<IFRAME>**). Используя тэг **<DIV>**, Web-страницу можно разделить на части и каждой из них присвоить отдельный стиль.

Пример

```

<html>
<head>
<style type="text/css">
<!--
.title { text-align:center;font-size:150%;font-weight:bold;color:red }
.bold { font-size:120%;font-weight:bold;color:blue }
.end { font-weight:bold;color:blue }
-->
</style>
<meta http-equiv="Content-Type" content="text/html;charset=windows-
1251">
</head>
<div class=title>
Наречие<br>(The Abverb)
</div>
<p> Наречия часто образуются от прилагательных добавлением окончания
<span class=bold>-ly</span>:
slow - slow<span class=end>ly</span>,
quick - quick<span class=end>ly</span>.
</html>

```

Задание:

1. Создадим таблицу и применим к ней CSS -стили. В таблицу внесем данные о популярности различных браузеров. Для заголовка таблицы используем тег **<th>...</th>**.

HTML-код:

```

<html>
<head>
<title>Популярность браузеров в Мире</title>
</head>
<body>
<table>
<tr>
<th>Год\Браузер</th> <th>IE</th> <th>Firefox</th> <th>Safari</th>
<th>Opera</th> </th>
<tr>
<td>2010</td> <td>61.43%</td> <td>24.40%</td> <td>4.55%</td>
<td>2.37%</td> </tr> <tr>
<td>2009</td> <td>69.13%</td> <td>22.67%</td> <td>3.58%</td>
<td>2.18%</td>
</tr>
<tr>

```

```

<td>2008</td>
<td>77.83%</td>
<td>16.86%</td>
<td>2.65%</td>
<td>1.84%</td> </tr> <tr>

```

Без CSS-оформления таблица будет выглядеть так:

Год\Браузер IE Firefox Safari Opera

```

2010 61.43% 24.40% 4.55% 2.37%
2009 69.13% 22.67% 3.58% 2.18%
2008 77.83% 16.86% 2.65% 1.84%
2007 79.38% 14.35% 4.70% 0.50%

```

По умолчанию содержимое заголовочных ячеек отображается жирным шрифтом с выравниванием по центру.

Добавим в тег <head>...</head> тег <style>...</style>, а к тегу <table>...</table> атрибут id="browser_stats".

Запишем CSS-правила для таблицы. Для заголовочных ячеек установим серый фон и отступ содержимого от границ (padding) в половину высоты строки, для ячеек с данными - выравнивание по правому краю и padding три десятых от высоты строки. Вокруг таблицы зададим двойную рамку, а для ячеек – обычную одинарную.

```

<style>
/* стиль таблицы */
TABLE#browser_stats {
border : 3px double black; }
/* стиль заголовочных ячеек */
TABLE#browser_stats
TH{
border: 1px solid black;
background-color: gray;
padding: 0.5em; }
/* стиль ячеек с данными */
TABLE#browser_stats
TD{
border: 1px solid black;
padding: 0.3em;
text-align: right; }
</style>

```

В браузере:

| Год\Браузер | IE | Firefox | Safari | Opera |
|-------------|--------|---------|--------|-------|
| 2010 | 61.43% | 24.40% | 4.55% | 2.37% |
| 2009 | 69.13% | 22.67% | 3.58% | 2.18% |
| 2008 | 77.83% | 16.86% | 2.65% | 1.84% |
| 2007 | 79.38% | 14.35% | 4.70% | 0.50% |

рис. 1

Виден существенный недостаток: у каждой ячейки появилась собственная рамка.

Чтобы этого не происходило, необходимо указать в правилах для таблицы свойство **border-collapse** со значением **collapse**.

| Год\Браузер | IE | Firefox | Safari | Opera |
|-------------|--------|---------|--------|-------|
| 2010 | 61.43% | 24.40% | 4.55% | 2.37% |
| 2009 | 69.13% | 22.67% | 3.58% | 2.18% |
| 2008 | 77.83% | 16.86% | 2.65% | 1.84% |
| 2007 | 79.38% | 14.35% | 4.70% | 0.50% |

рис. 2

Теперь применим к той же таблице другое форматирование. Разделим таблицу двумя линиями на 3 части: названия браузеров, годы и процентные данные. Названия браузеров и процентные доли

выровняем по центру, годы - по правому краю. Зададим одинаковую ширину для столбцов с информацией по браузерам.

| Год\Браузер | IE | Firefox | Safari | Opera |
|-------------|--------|---------|--------|-------|
| 2010 | 61.43% | 24.40% | 4.55% | 2.37% |
| 2009 | 69.13% | 22.67% | 3.58% | 2.18% |
| 2008 | 77.83% | 16.86% | 2.65% | 1.84% |
| 2007 | 79.38% | 14.35% | 4.70% | 0.50% |

Чтобы применить правила CSS к левой колонке (годы), нам придется задать новый класс `lc` и прописать атрибут `class="lc"` в каждой ячейке левой колонки.

Горизонтальная линия создается путем указания свойства `border-bottom` для ячеек `th`, вертикальная - `border-left` для ячеек класса `lc`

Код-страницы:

```
<html>
<head>
<title>Популярность браузеров в Мире</title>
<style>
TABLE#browser_stats {
border-collapse: collapse;
}
TABLE#browser_stats TH{ border-bottom: 1px solid black; }
TABLE#browser_stats TD{ padding: 0.3em; text-align: center; width: 7
0px; }
TABLE#browser_stats .lc{ text-align: right; border-right: 1px solid
black; width: 100px; }
</style> </head> <body>
<table id="browser_stats"> <tr>
<th class="lc">Год\Браузер</th>
<th>IE</th>
<th>Firefox</th>
<th>Safari</th>
<th>Opera</th> </tr> <tr>
<td class="lc">2010</td>
<td>61.43%</td>
<td>24.40%</td>
<td>4.55%</td>
<td>2.37%</td> </tr> <tr>
<td class="lc">2009</td>
<td>69.13%</td>
<td>22.67%</td>
<td>3.58%</td>
<td>2.18%</td> </tr> <tr>
<td class="lc">2008</td>
<td>77.83%</td>
<td>16.86%</td>
<td>2.65%</td>
<td>1.84%</td> </tr> <tr>
<td class="lc">2007</td>
<td>79.38%</td>
<td>14.35%</td>
<td>4.70%</td>
<td>0.50%</td>
</tr>
</table>
</body>
</html>
```

Псевдоклассы

Способы привязки правил оформления CSS к элементам документа HTML могут быть: по названию тега, по имени класса, по ID и т.п. В CSS также существует несколько **псевдоклассов**.

С помощью псевдоклассов можно задать стиль в зависимости от состояния элемента или его положения в документе.

Для ссылок определено 4 псевдокласса:

link - ссылки, которые не посещались пользователем;

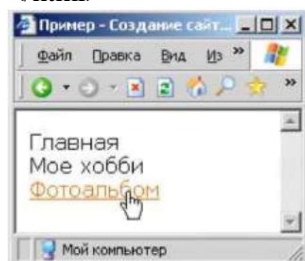
visited - посещенные ссылки;

active - активная (нажатая) ссылка;

hover - ссылка, на которую наведен курсор.

Пример:

```
<html>
<head>
<title>Пример</title>
<style>
A:link, A:visited {
color: black;
font-family: Verdana, sans-serif;
text-decoration: none;
}
A:hover {
color: #de7300;
text-decoration: underline;
}
</style>
</head>
<body>
<a href="index.html">Главная</a><br>
<a href="hobby.html">Мое хобби</a><br>
<a href="photo.html">Фотоальбом</a><br>
</body>
</html>
```



Позиционирование

С помощью CSS можно точно задать положение элемента на странице. Режимом позиционирования управляет свойство position:

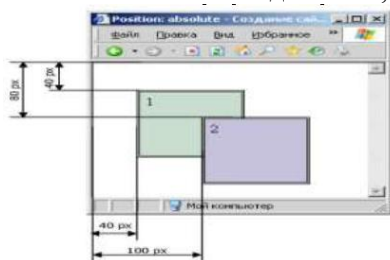
```
<html>
<head>
<title>Position: absolute</title>
<style>
DIV {
width: 100px;
height: 100px;
border: 3px double black;
padding: 5px;
position: absolute;
}
DIV#first {
background-color: #c0dcc0; top: 40px; left: 40px; }
DIV#second {
background-color: #c0c0dc; top: 80px; left: 100px; }
</style>
</head>
<body>
```

```

<div id="first">1</div>
<div id="second">2</div>
</body>
</html>

```

Для блоков задается отступ от верхнего и левого края свойствами top и left. Так как второй блок объявлен в HTML-коде позже, он перекрывает первый блок на странице.



Для управления порядком наложения элементов друг на друга необходимо использовать свойство z-index.

Плавающие элементы

По умолчанию блочные элементы идут строго друг под другом. Изменить этот порядок можно сделав элементы «плавающими». Для этого служит CSS атрибут float. Он задает, по какой стороне будет выравниваться элемент: левой (left) или правой (right).

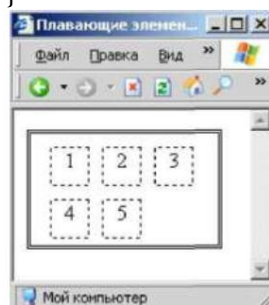
Плавающий элемент будет стремиться к левой или правой стороне родительского элемента, а с других сторон он может обтекаться текстом или другими элементами.

Создадим пример с несколькими плавающими блоками. Зададим основной контейнер с фиксированной шириной, а в него поместим пять плавающих блоков с выравниванием по левому краю.

```

<html>
<head>
<title>Плавающие элементы</title>
<style>
DIV#main {
border: double black 3px;
width: 150px;
padding: 5px;
}

```



Первый блок выравнивается по левому краю родительского контейнера. Второй блок тоже стремится к левому краю, но так как место уже занято первым блоком, второй блок становится (обтекает) справа от первого. Аналогично поступает третий блок.

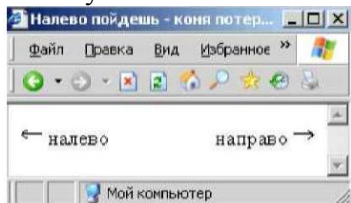
Четвертый блок уже не может встать справа от третьего, поэтому он помещается ниже остальных и выравнивается по левому краю. И наконец, пятый блок обтекает четвертый справа.

Можно одновременно использовать блоки с выравниванием по левому и правому краю.

```

<div style="float: left">← налево</div> <div style="float: right">направо →</div>

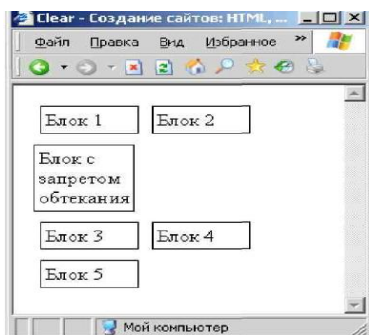
```



Еще одним свойством, связанным с плавающими элементами, является clear. Clear запрещает обтекание элемента с левой (left), правой (right) или с обеих сторон (both). По умолчанию значе-

ние - none - обтекание разрешено. Рассмотрим пример:

```
<html> <head>
<style>
DIV {
border: solid black 1px;
width: 75px;
}
DIV.floating { float: left; }
</style>
</head>
<body>
<div class="floating">Блок 1</div>
<div class="floating">Блок 2</div>
<div style="clear: both">Блок с запретом обтекания</div>
<div class="floating"><Блок 3</div>
<div class="floating">Блок 4</div>
<div class="floating">Блок 5</div>
</body>
</html>
```



При создании сайтов плавающие элементы, свойства float и clear часто используются для создания «каркаса» страниц сайта

2) Реализуйте примеры таблиц (рис. 1 и рис. 2).

3) Измените форматирование таблицы на следующий вариант:

| Год | Браузер | IE | Firefox | Safari | Opera |
|------|---------|--------|---------|--------|-------|
| 2010 | 61.43% | 24.40% | 4.55% | 2.37% | |
| 2009 | 69.13% | 22.67% | 3.58% | 2.18% | |
| 2008 | 77.83% | 16.86% | 2.65% | 1.84% | |
| 2007 | 79.38% | 14.35% | 4.70% | 0.50% | |

4) Измените форматирование: задайте одинаковую высоту строк и установите чередование фона («зебра»).

Указание: используйте разные классы стилей для четных и нечетных строк таблицы.

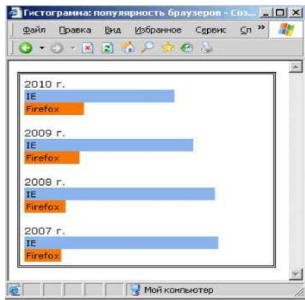
| Год | Браузер | IE | Firefox | Safari | Opera |
|------|---------|--------|---------|--------|-------|
| 2010 | 61.43% | 24.40% | 4.55% | 2.37% | |
| 2009 | 69.13% | 22.67% | 3.58% | 2.18% | |
| 2008 | 77.83% | 16.86% | 2.65% | 1.84% | |
| 2007 | 79.38% | 14.35% | 4.70% | 0.50% | |

5) Модифицируйте таблицу из задания «4». Чередование цвета фона у строк замените чередованием цвета фона у колонок.

6) Отключите фон у ячеек таблицы. Добавьте фоновое изображение для таблицы.

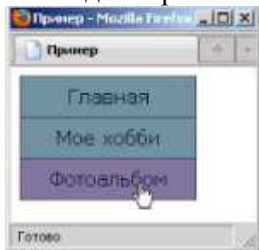
7) На основе данных о популярности браузеров за 2010 год создайте столбиковую диаграмму (гистограмму).

Пример

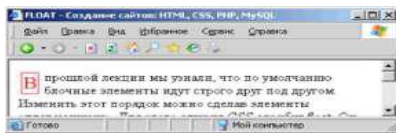


Указание: используйте элементы **DIV** заданной ширины

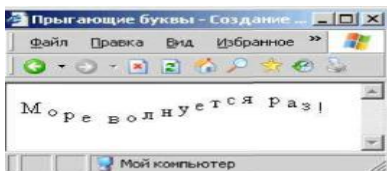
- 8) С помощью псевдокласса **first-letter** создайте свой вариант буквы. Подберите шрифт, размер, цвет и оформление
- 9) Добавьте эффект выделения ссылок при наведении курсора на своем сайте,
- 10) С помощью таблицы реализуйте меню, как показано на рисунке. Используя псевдокласс **hover** для строки таблицы, добавьте выделение пункта меню цветом при наведении курсора.



- 11) Используя плавающий блок, создайте буквицу, смещенную на одну строку вниз, как показано на рисунке



- 12) С помощью относительного позиционирования каждой буквы создайте «эффект морской волны».



- 13) Посредством абсолютного позиционирования изобразите точечный график математической функции. Пример для $y=x^2$ показан на рис. 15.

Указание: Создайте изображение с нарисованными на нем осями X и Y. Поверх него поместите точки графика абсолютным позиционированием.



Порядок выполнения:

Соответствует пунктам 1-13 задания.

Форма отчетности:

Отчет по практическому заданию, скрепленный титульным листом сдаётся в печатном виде. В отчете следует указать:

1. Цель работы.
2. Задание.
3. Поэтапное выполнения всех заданий ВИЗ.
4. Заключение (вывод).

Задания для самостоятельной работы:

Предусмотрены ВИЗ обучающегося.

Рекомендации по выполнению заданий и подготовке к практическому занятию

При подготовке и выполнении практического задания рекомендуется использовать материал лекций соответствующих разделов и литературу, предложенную для изучения данной дисциплины.

Основная литература

1. Ллойд, Й. Создай свой веб-сайт с помощью HTML и CSS : учебное пособие / Й. Ллойд; Пер. с англ. - Санкт-Петербург : Питер, 2013. - 416 с.

Дополнительная литература

2. Кобзов, А. Ю. Программные средства разработки Web- страниц и презентаций : методические указания к выполнению лабораторных работ / А. Ю. Кобзов, А. В. Кобзова. - Братск : БрГУ, 2014. - 43 с.

Контрольные вопросы для самопроверки

1. Задание логики сайта с помощью тегов DIV.
2. Подключение CSS. Значение CSS – каскадных таблиц стилей. Структура CSS..
3. Селекторы и свойства body и #header.
4. Использование отступов *. padding и margin.
5. Стилизация вложенных тегов. Импортирование содержимого CSS-файла в текущую стилевую таблицу @import.

Практическое занятие № 4. HTML-формы

Цель работы: получение практических навыков по созданию HTML-форм

Форма (формуляр) в HTML-документе определяется в виде блока <FORM> </FORM>, внутри которого располагаются теги, задающие те или иные формы, а также играющие роль подсказок текстовые строки и обращения к графическим файлам.

Внутри формы могут использоваться обычные HTML-теги, например, табличные.

В составе тега <FORM> записываются атрибуты, определяющие способ и характеристики обработки заключающейся в форме информации, в частности, способ пересылки ее на сервер и кодирования данных в процессе этой пересылки. Основные теги и атрибуты приведены в таблице:

| Тег | Атрибут | Функция и значение |
|--------|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <Form> | Name | <i>Создание формы:</i> Определяет имя, под которым содержимое поля будет передано обработчику (обязательный атрибут) |
| | <Form> Action | Указывает URL, по которому следует передать введенную информацию (обязательный атрибут) |
| | <Form> Enctype | Определяет способ кодирования содержимого формы при отправке. По умолчанию используется "APPLICATION/X- WWW-FORM-URLENCODED" |
| | <Form> Method | Определяет способ отправки содержимого формы. METHOD=POST - данные из формы передаются серверу отдельно от URL сценария обработчика формы. METHOD=GET - информация из формы для передачи на сервер записывается в конец URL обработчика |

| Тег | Атрибут | Функция и значение |
|------------|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <Form> Target | Определение имени окна для результата TARGET="_SELF" - указание на текущую форму TARGET="_PARENT" TARGET="_TOP" – указание на окно самого высшего уровня TARGET="_BLANK" - окно не будет иметь имени TARGET="_ИМЯ" - явно указанное имя окна |
| <Textarea> | Name | Создание поля для ввода нескольких строк текста: Определяет название, которое будет использоваться при идентификации заполненного поля сервером (обязательный параметр) |
| | <Textarea> Rows | Определяет количество строк текста, видимых на экране |
| | <Textarea> Cols | Определяет ширину текстового поля - в печатных символах |
| | <Textarea> Wrap | Определяет способ переноса слов в данной заполняемой форме. WRAP="OFF" – без переноса слов WRAP="VIRTUAL" – деление слов браузером WRAP="PHYSICAL" – перенос слов во всех точках переноса |
| <Option> | Value | Описание отдельных пунктов: Устанавливает значение, которое будет передано обработчику. Обязательный элемент |
| | <Option> Selected | Показывает, что данный элемент выбран |
| <Select> | Multiple | Создание меню выбора: Дает возможность выбора нескольких пунктов меню (по умолчанию - один пункт меню) |
| | <Select> Name | Определяет имя меню, которое будет использоваться при передаче данных на сервер |
| | <Select> Size | Определяет количество видимых пунктов в меню. SIZE=n Если n>1, то результат будет – раскрывающийся список пунктов |
| <Input> | Name | Создает поле формы (кнопку, поле ввода, чекбокс и т.п.): Указывает имя, под которым содержимое поля будет передано обработчику. Является обязательным |
| | <Input> Size | Устанавливает размер поля ввода текста или пароля в символах |
| | <Input> Maxlength | Устанавливает максимальное количество символов, вводимых в поле ввода текста или пароля |
| | <Input> | Устанавливает текст по умолчанию для поля |
| | <Input> | Включает флажок или переключатель |
| | <Input> | Устанавливает нужный тип поля ввода |

Пример

```
<!DOCTYPE HTML>
<html>
```

```

<head>
<meta charset="cp1251">
<title>Ter FORM</title>
</head>
<body>
<form action="handler.php">
<p><b>Как по вашему мнению расшифровывается аббревиатура "ОС"?</b></p>
<p><input type="radio" name="answer" value="a1">Офицерский состав<Br>
<input type="radio" name="answer" value="a2">Операционная система<Br>
<input type="radio" name="answer" value="a3">Однозначно согласен</p>
<p><input type="submit"></p>
</form>
</body>
</html>

```

Пример

```

<FORM>
<TEXTAREA NAME="MyText" ROWS=5 COLS=30>
Тут можно что-нибудь написать
</TEXTAREA>
</FORM>

```

Список - <SELECT>

Предназначен для создания меню, из которого пользователь может выбрать один или несколько из предложенных вариантов.

Атрибуты

NAME. Задаёт имя поля. Обязательный атрибут.

SIZE. Задаёт вертикальный размер окна для вариантов выбора. Если атрибут опущен или его значение равно 1, выводится всплывающий список вариантов. Если указано число больше 1, то варианты выводятся в окне с полосой прокрутки. Если значение атрибута больше, чем фактическое количество элементов списка, добавляются пустые строки. При их выборе пользователем возвращаются пустые поля.

MULTIPLE. Этот атрибут позволяет выбрать несколько вариантов одновременно.

Следует обратить внимание, что при множественном выборе в потоке данных от одной формы может присутствовать несколько переменных с одним и тем же именем. Ваша программа обработки должна предусматривать подобные ситуации и корректно их обрабатывать.

Список вариантов включается в контейнер <SELECT>...</SELECT> при помощи тэгов <OPTION>.

Тэг имеет два необязательных атрибута:

VALUE. Задаёт значение, передаваемое сценарию в случае выбора варианта пользователем. Если этот атрибут отсутствует, то в сценарий будет отослан текст, расположенный сразу после тэга <OPTION>.

DISABLED. Блокирует вариант для его выбора.

SELECTED. Указывает вариант, выделенный по умолчанию.

Примеры

```

<!--Выпадающий список-->
<FORM>
<SELECT NAME="Item">
<OPTION VALUE="tea">Чай
<OPTION VALUE="coffee">Кофе
<OPTION VALUE="milk">Молоко
<OPTION VALUE="ham">Ветчина
<OPTION VALUE="cheese">Сыр
</SELECT>
</FORM>

```

```

<!--Список с множественным выбором-->
<FORM>
<SELECT NAME="Item" SIZE=5 MULTIPLE>
<OPTION VALUE="tea">Чай
<OPTION VALUE="coffee" SELECTED>Кофе
<OPTION VALUE="milk">Молоко
<OPTION VALUE="ham">Ветчина
<OPTION VALUE="cheese">Сыр
</SELECT>
</FORM>

```

OPTGROUP

OPTGROUP - позволяет группировать пункты в элементе SELECT. В этом случае добавляются отступы и названия каждой группы, так что пользователю при выборе необходимого пункта ориентироваться гораздо

проще. Название группы задается в атрибуте LABEL элемента OPTGROUP. Вот пример использования группировки:

Пример

```
<form> <br>
<p>Справочники:</p> <br>
<select name="ask">
<optgroup label="HTML"> </optgroup>
<option label="3.2">HTML 3.2</option>
<option label="4.0">HTML 4.0</option>
</optgroup>
<optgroup label="CSS"> </optgroup>
<option label="1">CSS-1</option>
<option label="1">CSS-2</option>
<option label="3">CSS-3</option>
<option value="js">JavaScript</option>
<option value="dhtml">DHTML</option>
</select>
</form>
<INPUT>
```

Чаще всего в формах используется тэг **<INPUT>**. Он не имеет закрывающего тэга.

Вся информация, необходимая браузеру для обработки, содержится непосредственно в тэге **<INPUT>** и задается с помощью различных атрибутов.

Атрибуты

TYPE. Задает тип поля ввода. Может принимать значения: кнопка (BUTTON, SUBMIT, RESET), поле ввода (TEXT), поле ввода пароля (PASSWORD), скрытое поле (HIDDEN), флажок (CHECKBOX), переключатель (RADIO), файл (FILE).

С HTML5 добавились новые типы: color, date, datetime, datetime-local, email, month, number, range, search, tel, time, url, week. Если браузер не поддерживает какой-то из новых типов он будет считать, что это TEXT.

NAME. Задает имя поля. Каждое создаваемое поле ввода должно иметь собственное уникальное имя, иначе сценарий не определит, к каким полям относятся полученные значения. Конечно, имя поля ввода должно соответствовать имени, которое описано для него в программе обработки.

VALUE. Задает значение поля по умолчанию или надпись на кнопке.

ONCLICK. Задает обработчик щелчка на кнопке.

SIZE. Задает размер поля типа TEXT.

MAXLENGTH. Задает ограничение в поле типа TEXT вводимого количества символов.

AUTOCOMPLETE="OFF". Отключает автозаполнение для данного поля.

AUTOFOCUS="autofocus" or "". после загрузки страницы делает поле ввода активным.

readonly="readonly" or "". Запрещает изменение элемента.

disabled="disabled" or "". Делает элемент недоступным. Недоступные элементы не передаются на сервер.

required. Только HTML 5 Web Forms 2.0. Поле является обязательным. Автоматически проверяет заполнено ли поле.

required email. Только HTML 5 Web Forms 2.0. Автоматически проверяет корректно ли указан адрес e-mail.

min, max, step. Только HTML 5 Web Forms 2.0. Задает диапазон возможных значений, для даты формат 2012-12-19.

placeholder. Текст-подсказка в поле формы, который пропадает автоматически при получении поле фокуса.

Только HTML 5. Создает список вариантов, которые можно выбирать при наборе в текстовом поле. Изначально этот список скрыт и становится доступным при получении полем фокуса или наборе текста.

```
<input list="<идентификатор>">
<datalist id="<идентификатор>">
<option value="Текст1">
<option value="Текст2">
</datalist>
```

Кнопка - BUTTON

В форме изображается кнопка с надписью, заданной атрибутом VALUE, при нажатии на которую вызывается JavaScript-обработчик, заданный атрибутом ONCLICK.

Атрибут NAME служит для JavaScript-именования кнопки, а не для передачи на сервер.

Атрибут TYPE, который может принимать значения submit, reset и button. Первые два значения и так ясно для чего, а вот третье предназначено для тех случаев, когда надо исполнять какой-нибудь скрипт. То есть на кнопку вешается событие OnClick и вызывается нужная функция.

Пример

```
<FORM>
<INPUT TYPE="BUTTON" VALUE ="Жми!" ONCLICK="alert('Нажал!');">
</FORM>
```

Кнопку можно задать также с помощью парного тэга **<button>...</button>**. Внутри элемента BUTTON

можно размещать другие HTML-элементы, так что можно сделать кнопку с текстом и рисунком. Вот так:

Пример

```
<button type="button"> <br>  
Кнопочка  
 <br>  
</button>
```

Можно вместо рисунка вставить что угодно, хоть таблицу, если вам это необходимо.

Пример

```
<button type="button"> <br> <br> <br> <br>  
<table border="1" cellspacing="0" width="200"> <tbody>  
<tr> <td>нажми на кнопку</td>  
<td>получишь результат</td> </tr> </tbody>  
</table> <br>  
</button>
```

Атрибуты

TYPE. Тип кнопки. Может принимать значения:

BUTTON (просто кнопка; по умолчанию),

SUBMIT (кнопка завершения работы и передачи данных),

RESET (кнопка сброса).

ONCLICK. Обработчик щелчка на кнопке.

Данный элемент не поддерживается браузером Opera 5+.

Кнопка - SUBMIT

Эта кнопка предназначена для передачи формы. В большинстве браузеров внешне почти не отличима от кнопки BUTTON. Опять же сама она не передается, а служит только для управления. Атрибут ONCLICK практически не используется, так как лучше использовать обработчик событий ONSUBMIT, заданный в тэге <FORM>. Ведь чтобы передать введенные в форму данные, в общем случае совсем не обязательно нажимать на кнопку SUBMIT. Можно просто нажать на клавиатуре клавишу ENTER, находясь в текстовом поле ввода. При этом произойдет передача данных.

Пример

```
<FORM>  
<INPUT TYPE="SUBMIT" VALUE ="Принять заказ">  
</FORM>
```

Атрибут **VALUE** дает определенные преимущества при использовании более одной кнопки передачи данных. В этом случае на основании значения полученной переменной сценарий сможет определить, как обрабатывать полученную информацию далее.

Пример

```
<FORM>  
<INPUT TYPE="SUBMIT" NAME="BN" VALUE ="Вступить в клуб">  
<INPUT TYPE="SUBMIT" NAME="BN" VALUE ="Отказаться">  
<INPUT TYPE="SUBMIT" NAME="BN" VALUE ="Принять как наблюдателя">  
</FORM>
```

Атрибут **FormNoValidate** может быть применен чтобы предотвратить проверку значений формы. Например:

```
<form>  
<input type="text" required="" />  
<input type="submit" value="Проверка" />  
<input type="submit" formnovalidate="" value="Не проверять" />  
</form>
```

Кнопка - RESET

Это кнопка очистки формы. При ее нажатии всем измененным элементам возвращается значение по умолчанию.

Пример

```
<FORM>  
<INPUT TYPE="RESET" VALUE ="Сброс">  
</FORM>
```

Поле ввода - TEXT

Текстовое поле ввода используется в формах наиболее часто. Более того, его можно по праву считать основным и главнейшим элементом форм. Этот тип используется тэгом <INPUT> по умолчанию, его не нужно каждый раз указывать, чтобы вывести текстовое поле. Таким образом, на каждом текстовом поле

вы экономите 12 нажатий на клавиши, а главное, размер HTML-кода будет меньше. Имя поля, задаваемое атрибутом NAME, обязательно, так как базируясь именно на этом параметре, браузер передает сценарию пару имя=значение.

Пример

```
<FORM>
<INPUT NAME="Фамилия" VALUE="Иванов" SIZE=20 MAXLENGTH=20>
</FORM>
```

Текст Иванов помещается в созданное поле в качестве начального значения. Если пользователь не внесет изменений или нажмет кнопку RESET, то значение Иванов будет отправлено сценарию в качестве фамилии пользователя.

Поле ввода пароля - PASSWORD

Поле ввода пароля очень похоже на простое текстовое поле. Отличается оно тем, что вместо вводимых символов в нем отображаются звездочки. Такая возможность очень важна, когда нужно выспросить у пользователя секретную информацию типа пароля, которую не должны видеть другие.

Пример

```
<FORM>
<INPUT TYPE="PASSWORD" NAME="Пароль" SIZE=20>
</FORM>
```

Скрытое поле - HIDDEN

Это специальный (скрытый) тип текстового поля. Если один сценарий обрабатывает несколько разных форм, то в скрытом поле каждой формы можно указать идентификатор, который позволит определить, с какой формой вы имеете дело.

Пример

```
<FORM METHOD="POST" ACTION="/cgi_bin/test.cgi">
<INPUT TYPE="HIDDEN" NAME="NumForm" VALUE="1">
...Другие элементы формы...
</FORM>
<FORM
METHOD="POST" ACTION="/cgi_bin/test.cgi">
<INPUT
TYPE="HIDDEN" NAME="NumForm" VALUE="2">
...Другие элементы формы...
</FORM>
```

Браузер не отображает скрытое поле, хотя его можно обнаружить, если перевести браузер в режим просмотра HTML-файла и проанализировать текст HTML-страницы.

Скрытые поля полезны, если необходимо указать требуемую для сценария информацию, но при этом нежелательно, чтобы пользователь имел возможность вносить в нее изменения. Однако учтите, что сообразительный пользователь может сохранить вашу форму в файле, отредактировать его, а затем передать эту форму серверу в измененном виде. Поэтому не стоит полагаться на скрытые поля с целью создания какой-либо защиты.

Пример

```
<INPUT TYPE="HIDDEN" NAME="FormVersion" VALUE="1.2">
```

Сценарий получит переменную с именем FormVersion, которой будет присвоено значения 1.2. Эта информация может использоваться для определения способа обработки остальной информации, полученной от формы. Если пользователь изменит это значение, то программа сценария может повести себя неожиданным образом.

Флажок - CHECKBOX

Браузер отображает поле этого типа в виде небольшого квадрата. По смыслу флажок служит для того, чтобы быть установленным (квадрат перечеркнут), либо нет (квадрат пуст). Когда он установлен, его значение, заданное атрибутом VALUE, передается программе сценария. Если он не установлен, то его значение не передается совсем. Флажок может быть сразу установлен по умолчанию, если указан атрибут CHECKED. По умолчанию атрибут VALUE имеет значение ON (установлен). Так как атрибут VALUE здесь задает не надпись на флажке, а его внутреннее значение, передаваемое программе сценария, то если надо что-то подписать, пишите рядом с флажком.

Пример

```
<FORM>
Нужна накладная
<INPUT TYPE="CHECKBOX" NAME="Receipt" VALUE="Yes">
</FORM>
```

Для задания окружающего кликабельного текста используется тег **label**

```
<label><input type="checkbox" /> флаг</label>
```

```
<input type="checkbox" id="flag" /><label for="flag"> флаг</label>
```

Атрибут **for** тега **label** — задает уникальный идентификатор, определяемый с помощью атрибута **id** элемента `<input>`, с которым следует установить связь.

Переключатель - RADIO

Переключатель напоминает флажок, поскольку он тоже может находиться во включенном или выключенном состоянии. По смыслу всегда предполагается, что в форме имеется несколько переключателей с одинаковым атрибутом **NAME**. У каждого из них свое значение атрибута **VALUE**. Группа переключателей с одним и тем же именем в форме ведет себя таким образом, что только один из них может быть включенным. При передаче данных передается значение только выбранного переключателя. Один переключатель из группы может быть изначально выбран по умолчанию с помощью атрибута **CHECKED**.

Пример

```
<FORM>
```

```
<INPUT TYPE="RADIO" NAME="Sex" VALUE="Man" CHECKED>Мужской
```

```
<INPUT TYPE="RADIO" NAME="Sex" VALUE="Woman">Женский
```

```
</FORM>
```

Файл - FILE

Позволяет передать сценарию любой файл. Максимальный размер файла задается скрытым полем **MAX_FILE_SIZE**.

Пример

```
<form name=forma enctype="multipart/form-data"
```

```
method=POST action=http://htmlweb.ru/cgi-bin/file.cgi >
```

```
<INPUT TYPE="hidden" NAME="MAX_FILE_SIZE"
```

```
VALUE="10000" >
```

```
Фото:<INPUT TYPE="file" NAME="mURL" size=45>
```

```
</form>
```

Некоторые браузеры (например, Internet Explorer) отображают элементы форм, даже если они не находятся внутри контейнера `<FORM>...</FORM>`. Но злоупотреблять этим не стоит. Во-первых, в одном документе может находиться несколько форм, и именно `<FORM>...</FORM>` определяет, какие элементы к какой форме относятся; во-вторых, в тэге `<FORM>` задается URL обработчика формы и метод передачи данных. В этом плане корректнее всех поступает браузер Netscape Navigator, он вообще не отображает никаких элементов управления, описанных вне контейнера

```
<FORM>...</FORM>
```

```
</p>
```

```
<a name="fieldset"></a><H3>FIELDSET</H3>
```

```
<p>
```

Этот элемент позволяет группировать элементы форм.

Данная группировка помогает пользователям проще ориентироваться в формах, то есть ему сразу становится понятно, что вот все эти элементы относятся именно к этой форме.

Кроме того, можно эту группу еще и обозвать. Для этого используется элемент

```
<a href="#legend">LEGEND</a></p>
```

Ниже приведена форма, которая заключена в элемент **FIELDSET** и называется

«Ваш комментарий»

```
<form> <br>
```

```
<fieldset style="border: 2px solid rgb(0, 0, 0); padding: 10px; width:
```

```
200px;"> <br>
```

```
<legend>Ваш комментарий</legend>
```

```
<p>Вы можете разместить на сайте ваш комментарий, касающийся данной статьи</p>
```

```
<br>
```

```
<table> <tbody>
```

```
<tr><td>Имя:</td><td><input size="12"></td></tr>
```

```
<tr><td>E-mail:</td><td><input size="12"></td></tr>
```

```
<tr><td>Страна:</td>
```

```
<td><select>
```

```
<option selected="selected">Россия </option>
```

```
<option selected="selected">Китай </option>
```

```
<option selected="selected">Испания </option>
```

```
</select></td></tr>
```

```
<tr><td colspan="2">Комментарий:<br>
```

```
<textarea rows="4" cols="6"></textarea></td></tr>
```

```
<tr><td colspan="2">
```

```
<input value="отправить" type="submit"></td></tr>
```

```
</tbody></table> <br>
```

</fieldset>
</form>

Задание:

1. К созданной в первой лабораторной работе HTML-странице добавить форму, в соответствии с вариантом задания (или выбрать самостоятельно).
2. Оформить форму с использованием CSS (для оформления формы создать отдельный CSS-файл).

Варианты форм:

- 1 Форма для редактирования личной информации
- 2 Форма отправки сообщений работодателю
- 3 Форма редактирования текста визитки
- 4 Форма редактирования текста страницы
- 5 Форма добавления комментариев на странице
- 6 Форма отправки сообщения другу
- 7 Форма обратной связи с факультетом
- 8 Форма редактирования блюд
- 9 Форма добавления комментариев на страницу
- 10 Форма обратной связи с администрацией города
- 11 Форма добавления новости на страницу
- 12 Форма редактирования городов России
- 13 Форма редактирования страницы
- 14 Форма добавления новости на страницу

Порядок выполнения:

Соответствует пунктам 1-2 задания.

Форма отчетности:

Отчет по практическому заданию, скрепленный титульным листом сдаётся в печатном виде. В отчете следует указать:

1. Цель работы.
2. Задание.
3. Поэтапное выполнения всех заданий ВИЗ.
4. Заключение (вывод).

Задания для самостоятельной работы:

Предусмотрены ВИЗ обучающегося.

Рекомендации по выполнению заданий и подготовке к практическому занятию

При подготовке и выполнении практического задания рекомендуется использовать материал лекций соответствующих разделов и литературу, предложенную для изучения данной дисциплины.

Основная литература

1. Ллойд, Й. Создай свой веб-сайт с помощью HTML и CSS : учебное пособие / Й. Ллойд; Пер. с англ. - Санкт-Петербург : Питер, 2013. - 416 с.

Дополнительная литература

2. Кобзов, А. Ю. Программные средства разработки Web- страниц и презентаций : методические указания к выполнению лабораторных работ / А. Ю. Кобзов, А. В. Кобзова. - Братск : БрГУ, 2014. - 43 с.

Контрольные вопросы для самопроверки

1. Какие теги используются для создания форм?
2. Какие атрибуты тега <Form> вы знаете?
3. Назовите атрибуты тега <Select> и <Option>?
4. Перечислите атрибуты тега <Text Area> и <Input>?

10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

- Авторские комплекты слайдов, используемых при проведении лекционных занятий.
- ОС Windows 7 Professional.
- Microsoft Office 2007 Russian Academic OPEN No Level.
- Антивирусное программное обеспечение Kaspersky Security.
- Chrome.

11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

| <i>Вид занятия</i> | <i>Наименование аудитории</i> | <i>Перечень основного оборудования</i> | <i>№ ПЗ</i> |
|--------------------|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 1 | 2 | 3 | 4 |
| ПЗ | Лекционный кабинет/ дисплейный класс | Интерактивная доска SMART Board 680I со встроенным XGA проектором Unifi 35 (77"/195,6 см); 16-ПК: CPU 5000/RAM 2Gb/HDD; Монитор TFT 19 LG1953S-SF; Принтер: HP LaserJet P3015; Сканер: EPSON GT1500 | ПЗ №1-4 |
| СР | Читальный зал №1 | Оборудование 10 ПК i5-2500/H67/4Gb (монитор TFT19 Samsung); принтер HP LaserJet P2055D | - |

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ
ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО
ДИСЦИПЛИНЕ**

1. Описание фонда оценочных средств (паспорт)

| № компетенции | Элемент компетенции | Раздел | Тема | ФОС |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|-------------------------------------------------------|------------------------------|
| ОК-3 | способность использовать естественно-научные и математические знания для ориентирования в современном информационном пространстве | 1. Язык гипертекстовой разметки HTML | 1.1 Основные понятия HTML | вопросы к зачету 1.1. – 1.7. |
| | | 2. Каскадные таблицы стилей CSS | 2.1 Основы языка описания внешнего вида документа CSS | вопросы к зачету 2.1. – 2.5. |
| ПК-7 | способность организовывать сотрудничество обучающихся, поддерживать активность и инициативность, самостоятельность обучающихся, развивать их творческие способности | 1. Язык гипертекстовой разметки HTML | 1.1 Основные понятия HTML | вопросы к зачету 1.1. – 1.7. |
| | | 2. Каскадные таблицы стилей CSS | 2.1 Основы языка описания внешнего вида документа CSS | вопросы к зачету 2.1. – 2.5. |

2. Вопросы к зачету

| № п/п | Компетенции | | ВОПРОСЫ К ЗАЧЕТУ | № и наименование раздела |
|-------|-------------|----------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|
| | Код | Определение | | |
| 1 | 2 | 3 | 4 | 5 |
| 1. | ОК-3 | способность использовать естественнонаучные и математические знания для ориентирования в современном информационном пространстве | 1.1. Структура HTML–документа 1.2. Форматирование web-страниц 1.3. Создание списков и гиперссылок 1.4. Графика в HTML 1.5. Создание и форматирование таблиц в HTML 1.6. Создание фреймовой структуры в HTML 1.7. Создание форм в HTML | 1. Язык гипертекстовой разметки HTML |
| | | | 2.1. Что такое CSS? 2.2. Какие способы применения стилей в документе HTML существуют? | 2. Каскадные таблицы стилей CSS |

| | | | | |
|----|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------|
| | | | 2.3. Что такое селекторы? 2.4. Что такое псевдоклассы? 2.5. Что такое маски? | |
| 2. | ПК-7 | способность организовать сотрудничество обучающихся, поддерживать активность и инициативность, самостоятельность обучающихся, развивать их творческие способности | 1.1. Структура <i>HTML</i> -документа 1.2. Форматирование <i>web</i> -страниц 1.3. Создание списков и гиперссылок 1.4. Графика в <i>HTML</i> 1.5. Создание и форматирование таблиц в <i>HTML</i> 1.6. Создание фреймовой структуры в <i>HTML</i> 1.7. Создание форм в <i>HTML</i> | 1. Язык гипертекстовой разметки <i>HTML</i> |
| | | | 2.1. Что такое <i>CSS</i> ? 2.2. Какие способы применения стилей в документе <i>HTML</i> существуют? 2.3. Что такое селекторы? 2.4. Что такое псевдоклассы? 2.5. Что такое маски? | 2. Каскадные таблицы стилей <i>CSS</i> |

3. Описание показателей и критериев оценивания компетенций

| Показатели | Оценка | Критерии |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Знать: ОК-3: – основные виды современных <i>web</i> -технологий и принципы их функционирования; ПК-7: – особенности построения коммуникации с участниками образовательного процесса; Уметь: ОК-3: – структурировать информацию для её публикации на <i>web</i> -ресурсах; ПК-7: – конструктивно взаимодействовать с участниками образовательного процесса; Владеть: ОК-3: – навыками разработки макетов <i>web</i> -страниц, отвечающих современным требованиям; ПК-7: – навыками организации сотрудничества обучающихся. | зачтено | Оценка «зачтено» выставляется студенту, если вопросы раскрыты, изложены логично, без существенных ошибок, показано умение иллюстрировать теоретические положения конкретными примерами, продемонстрировано усвоение ранее изученных вопросов, сформированность компетенций, устойчивость используемых умений и навыков. Допускаются незначительные ошибки. |
| | не зачтено | Оценка «не зачтено» выставляется, если не раскрыто основное содержание учебного материала; обнаружено незнание или непонимание большей или наиболее важной части учебного материала; допущены ошибки в определении понятий, которые не исправлены после нескольких наводящих вопросов; не сформированы компетенции, умения и навыки. |

4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и опыта деятельности

Цель и задачи дисциплины «Интернет технологии» представлены в разделе 1 настоящей рабочей программы. Место дисциплины в структуре образовательной программы представлено в разделе 2 настоящей рабочей программы. Распределение объема

дисциплины по формам обучения с указанием видов учебных занятий представлено в разделе 3 настоящей рабочей программы. Содержание дисциплины указано в разделе 4 настоящей рабочей программы.

Учебно-методические материалы для самостоятельной работы обучающихся по дисциплине находятся в свободном доступе в соответствии с разделом 6 настоящей рабочей программы.

При изучении дисциплины необходимо использовать литературу, указанную в разделе 7 настоящей рабочей программы, а также перечень ресурсов информационно-телекоммуникационной сети «Интернет», представленных в разделе 8 настоящей рабочей программы.

Консультации для обучающихся по дисциплине проводятся в соответствии с графиком проведения консультаций, представленном на стенде кафедры, за которой закреплена указанная дисциплина.

К зачету допускаются обучающиеся очной формы обучения, которые выполнили и оформили все практические задания, предусмотренные в конкретном семестре. Методические указания по выполнению и оформлению представлены в разделе 9.1. настоящей рабочей программы.

Информационные технологии, используемые при освоении дисциплины, перечислены в разделе 10 настоящей рабочей программы.

Система оценивания уровня освоения дисциплины предусматривает текущий и итоговый (промежуточная аттестация) виды контроля.

Текущий контроль основан на проверке выполнения практических работ. При этом оценивается: правильность выполнения заданий, соблюдение требований к содержанию и оформлению отчетов, соблюдение сроков выполнения работ, уровень ответов при защите работ.

Оценка знаний, умений, навыков осуществляется в процессе промежуточной аттестации обучающихся по дисциплине, которая осуществляется в виде зачета (устного собеседования). Для оценивания знаний, умений, навыков используются ФОС по дисциплине, содержащий вопросы для зачета.

Описание показателей и критериев оценивания компетенций, сформированных по итогам изучения дисциплины, представлено в разделе 3 Приложения 1 настоящей рабочей программы.

АННОТАЦИЯ рабочей программы дисциплины Интернет технологии

1. Цель и задачи дисциплины

Целью преподавания дисциплины является изучение и освоение современных web-технологий.

Задачами изучения дисциплины являются:

- изучение основ HTML и CSS;
- изучение технологий разработки web-сайтов;
- изучение основ разработки интернет приложений.

2. Структура дисциплины

2.1 Распределение трудоемкости по отдельным видам учебных занятий, включая самостоятельную работу: практические занятия – 36 часов, самостоятельная работа 36 часов.

Общая трудоемкость дисциплины составляет 72 часа, 2 зачетные единицы.

2.2 Основные разделы дисциплины:

- 1 – Язык гипертекстовой разметки HTML
- 2 –Каскадные таблицы стилей CSS

3. Планируемые результаты обучения (перечень компетенций)

Процесс изучения дисциплины направлен на формирование следующих компетенций:

ОК-3 – способность использовать естественнонаучные и математические знания для ориентирования в современном информационном пространстве;

ПК-7 – способность организовывать сотрудничество обучающихся, поддерживать активность и инициативность, самостоятельность обучающихся, развивать их творческие способности.

4. Вид промежуточной аттестации: зачет.

*Протокол о дополнениях и изменениях в рабочей программе
на 20__-20__ учебный год*

1. В рабочую программу по дисциплине вносятся следующие дополнения:

2. В рабочую программу по дисциплине вносятся следующие изменения:

Протокол заседания кафедры № _____ от «___» _____ 20__ г.,
(разработчик)

Заведующий кафедрой _____
(подпись)

(Ф.И.О.)

Программа составлена в соответствии с федеральным государственным образовательным стандартом высшего образования по направлению подготовки 44.03.01 Педагогическое образование от «04» декабря 2015 г. № 1426

для набора 2015 года: и учебным планом ФГБОУ ВО «БрГУ» для очной формы обучения от «03»июля 2018 г. №413

для набора 2016 года: и учебным планом ФГБОУ ВО «БрГУ» для очной формы обучения от «06»июня 2016 г. № 429

для набора 2017 года: и учебным планом ФГБОУ ВО «БрГУ» для очной формы обучения от «06»марта 2017 г. №125

для набора 2018 года: и учебным планом ФГБОУ ВО «БрГУ» для очной формы обучения от «12»марта 2018 г. №130

Программу составили:

Ефремова Аида Николаевна,
ст. преподаватель кафедры ИиПМ _____

Рабочая программа рассмотрена и утверждена на заседании кафедры ИиПМ

от «19» декабря 2018 г., протокол № 5

И.о. заведующего кафедрой ИиПМ _____ А.С. Толстиков

СОГЛАСОВАНО:

Заведующий базовой кафедрой ИПиП _____ В.В. Кудряшов

Директор библиотеки _____ Т.Ф. Сотник

Рабочая программа одобрена методической комиссией ЕН факультета

от «20» декабря 2018 г., протокол № 4

Председатель методической комиссии факультета _____ М.А. Варданян

СОГЛАСОВАНО:

Начальник
учебно-методического управления _____ Г.П. Нежевец

Регистрационный № _____