

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«Брестский государственный университет имени А.С. Пушкина»
Кафедра алгебры, геометрии и математического моделирования

Д.В. Грицук

ИНТЕРНЕТ-ТЕХНОЛОГИИ

*Учебно-методический комплекс для студентов специальности
1-31 03 06-01 04 «Экономическая кибернетика (математические
методы и компьютерное моделирование в экономике)» со
специализацией "Информационные технологии управления в
экономике"
физико-математического факультета*

Брест, 2017



*Кафедра
АГММ*

Начало

Содержание



Страница 1 из 187

Назад

На весь экран

Закреть

Автор:

Грицук Дмитрий Владимирович — доцент кафедры алгебры, геометрии и математического моделирования Учреждения образования «Брестский государственный университет имени А.С. Пушкина», кандидат физико-математических наук

Рецензенты:

Матысик Олег Викторович — заведующий кафедрой прикладной математики и информатики Учреждения образования «Брестский государственный университет имени А.С. Пушкина», кандидат физико-математических наук, доцент

Кафедра информатики Учреждения образования «Гомельский государственный технический университет имени П.О. Сухого»



*Кафедра
АГММ*

Начало

Содержание



Страница 2 из 187

Назад

На весь экран

Закреть

Учебно-методический комплекс написан в соответствии с действующей учебной программой по дисциплине «Интернет-технологии» и ставит своей целью облегчить самостоятельную работу студентов с теоретическим материалом при подготовке к лекциям, практическим занятиям и к зачету.

Предназначен для студентов специальности 1-31 03 06-01 04 «Экономическая кибернетика (математические методы и компьютерное моделирование в экономике)» со специализацией "Информационные технологии управления в экономике" физико-математического факультета.



*Кафедра
АГyMM*

Начало

Содержание



Страница 3 из 187

Назад

На весь экран

Закрyть

СОДЕРЖАНИЕ

Предисловие	9
Содержание учебного материала	10
Примерный тематический план	12
Раздел 1 ВВЕДЕНИЕ	15
1.1 Основные понятия интернет-технологий	15
Раздел 2 ЯЗЫК ГИПЕРТЕКСТОВОЙ РАЗМЕТКИ HTML	18
2.1 История возникновения HTML	18
2.2 Структура документа	21
2.3 Форматирование шрифта	24
2.3.1 Физические стили	25
2.3.2 Логические стили	27
2.4 Форматирование абзацев	28
2.5 Заголовки	29
2.6 &-последовательности	30
2.7 Комментарии	31
2.8 Графика	31
2.9 Гиперссылка	36
2.10 Списки	39
2.11 Таблицы	42
2.12 Фреймы	51



Кафедра
АГyMM

Начало

Содержание



Страница 4 из 187

Назад

На весь экран

Заккрыть

2.12.1	Описание фреймов	51
2.12.2	Сложные фреймовые структуры	54
2.13	Формы	58
2.13.1	Задание формы	58
2.13.2	Элементы управления	59
Раздел 3	КАСКАДНЫЕ ТАБЛИЦЫ СТИЛЕЙ	69
3.1	CSS	69
3.2	Использование CSS в HTML-страницах	74
3.3	Основные свойства CSS	76
Раздел 4	ЯЗЫК JAVASCRIPT	83
4.1	Типы данных	85
4.2	Операторы	87
4.2.1	Арифметические операторы	87
4.2.2	Оператор присваивания	88
4.2.3	Операторы сравнения	88
4.2.4	Логические операторы	89
4.2.5	Оператор получения типа typeof	90
4.2.6	Условный оператор "if"	90
4.2.7	Условный оператор "?"	91
4.2.8	Оператор выбора "switch"	92
4.2.9	Цикл со счетчиком	93



*Кафедра
АГуММ*

Начало

Содержание

◀ ▶

◀◀ ▶▶

Страница 5 из 187

Назад

На весь экран

Закреть

4.2.10	Цикл с предусловием	94
4.2.11	Цикл с постусловием	95
4.2.12	Операторы прерывания и перезапуска цикла	95
4.3	Функции	96
4.3.1	Объявление и вызов функций	96
4.3.2	Встроенные функции JavaScript	98
4.3.3	Рекурсия	100
4.4	Объекты	101
4.4.1	Объект строки String	102
4.4.2	Объект числа Number	105
4.4.3	Объект логического значения Boolean	107
4.4.4	Объект значения даты Date	108
4.4.5	Объект массив Array	110
4.4.6	Объект Math	112
4.5	Общие принципы написания Web-сценариев	114
4.5.1	Внутреннее представление страницы (Document Object Model)	114
4.5.2	Работа с окнами Web-обозревателя	117
4.5.3	События элементов страницы и их обработка	119
4.5.4	Работа с web-формами и элементами управления	120
4.5.5	Управление графикой и мультимедийными элементами	127



*Кафедра
АГyMM*

Начало

Содержание



Страница 6 из 187

Назад

На весь экран

Закреть

Раздел 5 СОВРЕМЕННЫЕ СРЕДСТВА СОЗДАНИЯ WEB-СТРАНИЦ **129**

5.1 Скриптовые языки программирования	129
5.2 Средства создания Web-страниц	131
5.3 Создание Web-страницы с помощью FrontPage	134
5.3.1 Вставка таблиц в редакторе FrontPage	136
5.3.2 Добавление ссылок и изображений	142

Раздел 6 ОСНОВНЫЕ КОМПОНЕНТЫ WEB-СТРАНИЦЫ И СПОСОБЫ ИХ ВИЗУАЛЬНОГО ПРЕДСТАВЛЕНИЯ НА СТРАНИЦАХ САЙТА **145**

6.1 Эргономика WEB-сайта	145
6.2 Требования к иллюстрациям в Internet	150
6.3 Обзор типов файлов иллюстраций	151
6.4 Форматы JPEG, GIF, PNG	156
6.5 WEB-оптимизация изображений	158

ПРАКТИЧЕСКИЙ РАЗДЕЛ **160**

Лабораторная работа №1	160
Лабораторная работа №2	160
Лабораторная работа №3	164
Лабораторная работа №4	168



*Кафедра
АГУММ*

Начало

Содержание



Страница 7 из 187

Назад

На весь экран

Заккрыть

Лабораторная работа №5	169
Лабораторная работа №6	170
Лабораторная работа №7	171
Лабораторная работа №8	173
Лабораторная работа №9	175
Лабораторная работа №10	176
ТЕСТЫ	177
Вопросы к зачету	184
Список использованной и рекомендованной литературы	186



*Кафедра
АГyMM*

Начало

Содержание



Страница 8 из 187

Назад

На весь экран

Закрyть

Предисловие

Настоящий учебно-методический комплекс предназначен для студентов специальности 1-31 03 06-01 04 «Экономическая кибернетика (математические методы и компьютерное моделирование в экономике)» со специализацией "Информационные технологии управления в экономике" физико-математического факультета. Он написан в соответствии с действующей учебной программой по курсу «Интернет-технологии».

В теоретическом разделе учебно-методического комплекса излагаются основы создания Web-страниц: язык гипертекстовой разметки HTML, каскадные таблицы стилей, язык JavaScript и др. Изложение сопровождается большим количеством примеров. В практической части приводятся задания лабораторных работ, выполнение которых будет способствовать формированию умений и навыков, которые могут быть применены в профессиональной деятельности, в частности, при проектировании Web-сайтов. Вспомогательный раздел содержит список использованной и рекомендованной литературы. В электронном учебно-методическом комплексе приводятся тесты по курсу дисциплины "Интернет-технологии" и список вопросов для подготовки к зачету.

Электронный учебно-методический комплекс ставит своей целью облегчить самостоятельную работу студентов с теоретическим материалом при подготовке к лекциям, лабораторным занятиям и зачету.

Автор



Кафедра
АГуММ

Начало

Содержание



Страница 9 из 187

Назад

На весь экран

Закреть

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1. ВВЕДЕНИЕ

Введение. Основные понятия интернет-технологий: WWW (всемирная паутина), WEB-страница, WEB-узел, сайт, гипертекст, гиперссылка, браузер.

Раздел 2. Язык гипетекстовой разметки HTML

История возникновения HTML. Этапы разработки WEB-сервера. Принципы гипертекстовой разметки. Структура документа. Элементы разметки заголовка документа. Элементы тела документа, виды тэгов. Физическое и логическое форматирование текста WEB-страницы. Добавление на страницу изображений, гиперссылок. Навигация между HTML-документами и в пределах документа.

Теги и атрибуты создания списков.

Таблицы – основное средство для управления расположением материалов. Структура и элементы таблицы. Теги и атрибуты создания таблиц.

Фреймы: набор фреймов, целевой фрейм отдельных гиперссылок, целевой фрейм по умолчанию.

Теги и атрибуты создания форм, элементов управления.

Раздел 3. Каскадные таблицы стилей

Возможности, внешние и встроенные таблицы стилей, записи таблицы стилей, порядок применения.

Раздел 4. Современные средства создания WEB-страниц

Создание и редактирование страниц с помощью редактора WEB-страниц FrontPage. Шаблоны WEB-узлов, установка параметров страницы, установка параметров форматирования, добавление ссылок и рисунков. Вставка таблиц, использование тем в редакторе, создание рамок на страницах, компоненты редактора.



Кафедра
АГyMM

Начало

Содержание



Страница 10 из 187

Назад

На весь экран

Закреть

Раздел 5. Основные компоненты WEB-страницы и способы их визуального представления на страницах сайта

Распределение информации по разделам сайта с учетом информационной, логической и визуальной взаимосвязи между разделами. Эргономика WEB-сайта. Факторы, которые затрудняют и факторы, которые облегчают восприятие пользователем информации на сайте. Требования к иллюстрациям в Internet. Обзор типов файлов иллюстраций. Форматы JPEG, GIF, PNG и др. WEB-оптимизация изображений: уменьшение времени загрузки изображений, уменьшение размеров изображений, средства сглаживания и оптимизации изображений, обеспечение качества изображений.

Раздел 6. Язык JavaScript

Назначение и применение JavaScript. Общие сведения. Размещение кода JavaScript на HTML-странице. Основные типы данных и операторы.

Функции. Объявление и вызов функций. Рекурсия. Встроенные функции JavaScript. Объекты JavaScript: встроенные и пользовательские.

Работа с WEB-формами.

Общие принципы написания WEB-сценариев. Обработка событий.

Управление графикой и мультимедийными элементами.

Анимация на WEB-страницах.



*Кафедра
АГуММ*

Начало

Содержание



Страница 11 из 187

Назад

На весь экран

Закреть

ПРИМЕРНЫЙ ТЕМАТИЧЕСКИЙ ПЛАН

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Название раздела, темы	Количество часов
<i>Раздел 1. Введение</i>	2
Введение. Основные понятия интернет-технологий: WWW (всемирная паутина), WEB-страница, WEB-узел, сайт, гипертекст, гиперссылка, браузер	2
<i>Раздел 2. Язык гипертекстовой разметки HTML</i>	8
История возникновения HTML. Этапы разработки WEB-сервера. Принципы гипертекстовой разметки. Структура документа. Элементы разметки заголовка документа. Элементы тела документа, виды тэгов. Физическое и логическое форматирование текста WEB-страницы. Добавление на страницу изображений, гиперссылок. Навигация между HTML-документами и в пределах документа	2
Основные теги и атрибуты списков. Таблицы - основное средство для управления расположением материалов. Структура и элементы таблицы. Основные теги и атрибуты.	2
Фреймы: набор фреймов, целевой фрейм отдельных гиперссылок. Основные теги и атрибуты.	2
Формы, создание, элементы управления. Основные теги и атрибуты	2



Кафедра
АГММ

Начало

Содержание



Страница 12 из 187

Назад

На весь экран

Закрыть

Название раздела, темы	Количество часов
<i>Раздел 3. Каскадные таблицы стилей</i>	2
Каскадные таблицы стилей: возможности, встроенные таблицы стилей, внешние таблицы стилей записи таблицы стилей, порядок применения	2
<i>Раздел 4. Современные средства создания WEB-страниц</i>	2
Создание и редактирование страниц с помощью редактора WEB-страниц FrontPage. Шаблоны WEB-узлов, установка параметров страницы, установка параметров форматирования, добавление ссылок и рисунков. Вставка таблиц, использование тем в редакторе, создание рамок на страницах, компоненты редактора	2
<i>Раздел 5. Основные компоненты WEB-страницы и способы их визуального представления на страницах сайта</i>	2
Распределение информации по разделам сайта с учетом информационной, логической и визуальной взаимосвязи между разделами. Эргономика WEB-сайта. Факторы, которые затрудняют и факторы, которые облегчают восприятие пользователем информации на сайте	2
<i>Раздел 6. Язык JavaScript</i>	10
Назначение и применение JavaScript. Общие сведения. Размещение кода JavaScript на HTML-странице. Основные типы данных и операторы	2



Кафедра
АГуММ

Начало

Содержание



Страница 13 из 187

Назад

На весь экран

Закреть

Название раздела, темы	Количество часов
Функции. Объявление и вызов функций. Рекурсия. Встроенные функции JavaScript. Объекты JavaScript: встроенные и пользовательские	2
Работа с WEB-формами	2
Общие принципы написания WEB-сценариев. Обработка событий	2
Управление графикой и мультимедийными элементами	2
Анимация на WEB-страницах	2
ВСЕГО:	26

ПРАКТИЧЕСКАЯ ЧАСТЬ

Лабораторная работа	Количество часов
Лабораторная работа №1	2
Лабораторная работа №2	2
Лабораторная работа №3	2
Лабораторная работа №4	2
Лабораторная работа №5	2
Лабораторная работа №6	4
Лабораторная работа №7	4
Лабораторная работа №8	4
Лабораторная работа №9	4
Лабораторная работа №10	8
ВСЕГО:	34



Кафедра
АГуММ

Начало

Содержание



Страница 14 из 187

Назад

На весь экран

Заккрыть

РАЗДЕЛ 1

ВВЕДЕНИЕ

1.1 Основные понятия интернет-технологий

Интернет давно уже прочно обосновался в жизни современных людей, а для некоторых из них это единственный способ самовыражения и поиска друзей. Тем не менее, как это ни смешно, но мало кто сможет сразу ответить на вопрос «что же такое интернет?» Как правило, очевидным ответом станет «всемирная компьютерная сеть».

Если же быть точным, то *Интернет* – это не единственная сеть, а совокупность более мелких сетей, связанных друг с другом общими каналами и стандартами. Таких сетей превеликое множество: огромные территориальные сети, раскинувшиеся на целые области и государства, ведомственные сети, объединяющие родственные организации, локальные компьютерные сети отдельных организаций. Благодаря проложенным между ними каналам связи они и составляют Интернет.

Интернет имеет достаточно долгую и бурную историю. Он появился в конце 60-х годов XX века, когда Министерство обороны США финансировало проект создания компьютерной сети, устойчивой к сбоям. Создавалась эта сеть для нужд обороны, и имела другое название – ARPANET. Позднее, в начале 80-х, эта сеть отошла к ученым, а военные приступили к созданию другой сети, которой пользуются до сих



Кафедра
АГyMM

Начало

Содержание



Страница 15 из 187

Назад

На весь экран

Закрyть

пор. И в то же время ARPANET был переименован в Internet.

Первоначально, еще во времена ARPANET эта сеть использовалась для пересылки электронной почты и обмена файлами. Web-странички, ради которых мы, в основном, и путешествуем по Сети, появились только в 80-х. Именно тогда Интернет и «пошел в народ», перестав быть сетью ученых и превратившись в сеть для всех.

В СССР Интернет официально пришел в 1991 году, но популярность среди широких масс «компьютерщиков» приобрел только в середине 90-х. В настоящее время, наверное, и не найти человека не слышавшего об Интернете.

На основе интернета работает Всемирная паутина (World Wide Web, WWW) и множество других систем передачи данных.

WWW (World Wide Web, Всемирная паутина) – сервис Интернета, который состоит из Web-страниц, связанных между собой посредством гиперссылок и расположенных на различных компьютерах, которые подключены к Интернету.

Web-страница – это интернет-документ, предназначенный для распространения через интернет посредством сервиса WWW. С технической точки зрения Web-страница – это обычный текстовый файл, который можно создавать в любом текстовом редакторе, например, Блокноте, стандартно поставляемом в составе Windows. Этот файл содержит собственно текст Web-страницы и различные команды форматирования этого текста. Команды форматирования называют тегами, а описывает



Кафедра
АГУММ

Начало

Содержание



Страница 16 из 187

Назад

На весь экран

Закреть

их особый язык HTML (HyperText Markup Language, язык гипертекстовой разметки). Файл Web-страницы обязательно должен иметь расширение html.

Особые связи, ведущие от одной Web-страницы к другой называются *гиперссылками*. Набор Web-страниц, подчиненных общей тематике и объединенных в единое целое, посредством гиперссылок, называется *Web-сайтом*. Web-сайт сохраняется на жестких дисках серверного компьютера, на котором работает Web-сервер (серверная программа, обеспечивающая работу сервиса WWW), в виде набора различных файлов. Прежде всего, это файлы Web-страниц, составляющих сайт. Многие сайты включают файлы графических изображений, помещенных на страницах. Также сайт может содержать файлы архивов и дистрибутивов программ и некоторые другие файлы.

Программы, предназначенные для просмотра Web-страниц и Web-сайтов называются *Web-обозревателями* или *браузерами*. Основное назначение браузера – принять введенный пользователем адрес (URL) искомой страницы, скопировать ее с сервера в Интернете на диск локального компьютера (в специально отведенную "буферную" область), а затем правильно отобразить все элементы страницы (текст, графику, звук и т.п.) в рабочем окне. Браузеры используются не только как средство навигации в Интернете, но и для просмотра любых HTML-документов, а также рисунков форматов GIF и JPEG в режиме "off-line", то есть без подключения к Интернету.



Кафедра
АГУММ

Начало

Содержание



Страница 17 из 187

Назад

На весь экран

Закрыть

РАЗДЕЛ 2

ЯЗЫК ГИПЕРТЕКСТОВОЙ РАЗМЕТКИ HTML

2.1 История возникновения HTML

Название языка HTML является сокращением от HYPERTEXT MARKUP LANGUAGE, то есть "язык гипертекстовой разметки".

Гипертекст (hypertext) - текст, представленный в виде ассоциативно связанных блоков, переход между которыми осуществляется с помощью гиперссылок.

Гиперссылка - фрагмент текста (как правило отличающегося элементами форматирования: цвет, подчеркивание, курсив) или графика, выбор которого позволяет выполнить переход к другому фрагменту текста (файла, Web-странице).

Разметка - вставка в текст дополнительных служебных символов (например, пометки корректора тоже можно считать своего рода разметкой). Каждый служебный символ в HTML представляет собой команду, которая указывает программе-браузеру, как следует отображать текст.

Язык HTML был разработан в начале 80-х годов XX века Тимом Бернесом Ли. Кроме самого HTML, он также создал первые программы Web-обозревателя и Web-сервера – и первые Web-страницы. Можно сказать, что он создал WWW и Интернет в современном его виде.



Кафедра
АГММ

Начало

Содержание



Страница 18 из 187

Назад

На весь экран

Закреть

HTML был задуман как средство структурирования и форматирования документов без их привязки к средствам воспроизведения (отображения). Текст с разметкой HTML должен был без стилистических и структурных искажений воспроизводиться на оборудовании с различной технической оснащённостью (цветной экран современного компьютера, монохромный экран органайзера, ограниченный по размерам экран мобильного телефона или устройства и программы голосового воспроизведения текстов).

С момента своего появления стандарт HTML претерпел множество изменений. Самая последняя версия, вышедшая в 2014 году, HTML5.

HTML является компьютерным языком, но не является языком программирования. Он предназначен не для написания программ, а совсем для других целей. HTML был задуман как универсальный язык для функциональной классификации различных частей документа. Другими словами, HTML определяет, для чего нужна та или иная часть вашей страницы. Вы указываете, где находится заголовок, какую часть документа следует выделить, где должны располагаться графические изображения и т.д.

HTML предназначен для того, чтобы классифицировать части документа и обеспечивать его правильное отображение в браузере. Язык HTML обладает собственным синтаксисом, терминологией и правилами.

Ключевым понятием языка HTML является т.н. «тег» – конструкция, которая влияет на отображение контента (т.е. содержимого) на web-



Кафедра
АГуММ

Начало

Содержание



Страница 19 из 187

Назад

На весь экран

Закрыть

странице. Все в документе HTML содержится в тегах, и, если не обрамляется ими, то содержится в значениях атрибутов тегов. Почти все теги имеют открывающую часть, которая указывает на начальную границу охватываемого контента и содержит информацию о поведении тега. Закрывающая часть тега, если она есть, указывает на конечную границу охватываемого контента. Если тег имеет и открывающую, и закрывающую части, он называется *парным*. Теги, не охватывающие явно какой-либо контент, не имеют закрывающей части и называются *непарными*; поведение таких тегов полностью описывается их атрибутами (например, теги `<meta>`, `<input>` и др.).

Синтаксис внутри тегов HTML подчиняется следующим правилам:

- имена тегов не чувствительны к регистру;
- между символами `<`, `>`, `/` и именами тегов, а также внутри имен тегов не допускаются пробелы и переносы строк;
- в обычном тексте, не являющемся тегом, не должны присутствовать символы `<` и `>`;
- пары "*атрибут тега = значение*" перечисляются через пробелы и только внутри открывающей части тега, при этом:
 - значение атрибута может не указываться, если оно совпадает с названием атрибута (например, атрибут `loop` в теге `<video>`);



Кафедра
АГУММ

Начало

Содержание



Страница 20 из 187

Назад

На весь экран

Заккрыть



Кафедра
АГУММ

Начало

Содержание



Страница 21 из 187

Назад

На весь экран

Заккрыть

- внутри имен атрибутов не должны присутствовать пробелы;
- значение атрибута тега пишется после его имени и заключается в двойные кавычки;
- между именем атрибута тега и его значением ставится знак равенства;
- между именем атрибута тега, знаком равенства и открывающими кавычками могут присутствовать пробелы или разрывы строк;
- символы двойных кавычек недопустимы и не должны присутствовать в обычном тексте;
- закрывающая часть тега имеет вид $\langle /название\ тега \rangle$.

2.2 Структура документа

Независимо от того, как выглядит ваша страница и какую информацию вы хотите отобразить, существует три тега, которые в соответствии со стандартами HTML должны присутствовать на каждой странице:

1. $\langle HTML \rangle \dots \langle /HTML \rangle$.

Эти тэги сообщают браузеру, что текст между ними следует интерпритировать как HTML-текст.

2. <HEAD>...</HEAD>.

Отмечает вводную и заголовочную части HTML-документа. Они должны быть между тэгами <HTML>...</HTML>. Тэги <HEAD> и </HEAD> отмечают ту информацию в нашем документе, которая будет служить названием. В этих тэгах должна содержаться следующая информация:

- <TITLE> *Ваша страничка* </TITLE> - Между этими двумя тэгами необходимо ввести заголовок Вашей странички.

3. <BODY>...</BODY>.

Отмечает основной текст и информацию. С их помощью можно изменить стиль страницы, перед тем как набирать текст.

Таким образом, HTML-документ имеет вполне определенную структуру, которая представляет собой следующее:

```
<HTML>
  <HEAD>
    <TITLE> Безымянный </TITLE>
  </HEAD>
  <BODY>
    Это наш первый документ
  </BODY>
```



Кафедра
АГуММ

Начало

Содержание



Страница 22 из 187

Назад

На весь экран

Закрыть

</HTML>

На теге <BODY> и </BODY> следует заострить внимание, так как он содержит важные атрибуты, без знания которых невозможно обойтись.

1. Атрибут *bgcolor="цвет"* задает цвет фона (при отсутствии этого атрибута цвет фона – белый). Вместо атрибута *bgcolor="цвет"* может использоваться атрибут *background="имя графического файла"* – фоновый рисунок.
2. Атрибут *text="цвет"* задает цвет текста (при его отсутствии цвет текста – черный).
3. Атрибут *link="цвет"* задает цвет гиперссылок.
4. Атрибут *alink="цвет"* задает цвет активной гиперссылки.
5. Атрибут *vlink="цвет"* задает цвет посещенных гиперссылок.

Задание цвета производится несколькими способами:

- названием (например, "red");
- шестнадцатеричным числом (например, "#FFFFFF");
- в десятичном виде (например, "255,0,0");
- в процентах (например, "30%, 40%, 30%").



Кафедра
АГММ

Начало

Содержание



Страница 23 из 187

Назад

На весь экран

Заккрыть

Допустимые названия цветов:

aqua – аквамарин;

black – черный;

blue – синий;

fuchsia – фуксиновый;

green – зеленый;

gray – серый;

lime – светло-зеленый;

maroon – каштановый;

navy – ультрамарин;

olive – оливковый;

purple – пурпурный;

red – красный;

silver – серебристый;

teal – сизый;

white – белый;

yellow – желтый.

2.3 Форматирование шрифта

Для задания фрагменту текста некоторых параметров шрифта, отличных от первоначальных, используется тег `...` с сопутствующими атрибутами.



Кафедра
АГУММ

Начало

Содержание



Страница 24 из 187

Назад

На весь экран

Закрыть



1. Атрибут *color* устанавливает цвет шрифта.
2. Атрибут *size* устанавливает размер шрифта.

Размер шрифта в HTML задается одним из следующих способов:

- в процентах (например, `size="150%"`);
- в пикселах (например, `size="30px"`);
- в типографских пунктах (например, `size="24pt"`).

3. Атрибут *face* устанавливает тип шрифта: Courier, Arial и т.д.

В HTML допускается два способа задания параметров шрифта, таких как начертание и некоторые дополнительные эффекты выделения фрагментов текста. Во-первых, можно прямо указать, что шрифт на некотором участке текста должен быть **жирным** или *наклонным*, то есть изменить *физический стиль* текста. Во-вторых, можно пометить некоторый фрагмент текста, как имеющий некоторый отличный от начального *логический стиль*.

2.3.1 Физические стили

1. `...`

От английского bold – полужирный. Шрифт текста, к которому применен данный тег, будет иметь **полужирное** начертание.



2. `<i>...</i>` От английского *italic* – курсивный. Шрифт текста, к которому применен данный тег, будет иметь *курсивное* начертание.

3. `<u>...</u>`

От английского *underlined* – подчеркнутый. Текст, к которому применен данный тег, будет подчеркнут.

4. `<s>...</s>`

От английского *striked out* – зачеркнутый. Текст, к которому применен данный тег, будет ~~зачеркнут~~.

5. `<BIG>...</BIG>`

Данный тег выводит текст шрифтом на *ОДИН размер* больше изначального.

6. `<SMALL>...</SMALL>`

Этот тег выводит текст шрифтом на один размер меньше изначального.

7. `<TT>...</TT>`

Текст, размещенный между этими метками, будет написан шрифтом, имитирующим пишущую машинку, то есть имеющим фиксированную ширину символа.

2.3.2 Логические стили

При использовании логических стилей автор документа не может знать заранее, что он увидит в окне браузера. Разные браузеры толкуют одни и те же метки логических стилей по-разному. Некоторые браузеры игнорируют некоторые метки вообще и показывают нормальный текст вместо выделенного логическим стилем. Рассмотрим самые распространенные логические стили.

1. `...`

От английского *emphasis* – акцент. Как правило соответствует курсивному начертанию.

2. `...`

От английского *strong emphasis* – сильный акцент. Как правило соответствует полужирному начертанию.

3. `<CODE>...</CODE>`

Рекомендуется использовать для фрагментов исходных текстов.

4. `<SAMP>...</SAMP>`

От английского *sample* – образец. Рекомендуется использовать для демонстрации образцов сообщений, выводимых на экран программами.



Кафедра
АГyMM

Начало

Содержание



Страница 27 из 187

Назад

На весь экран

Закреть

5. `<KBD>...</KBD>`

От английского keyboard – клавиатура. Рекомендуется использовать для указания того, что нужно ввести с клавиатуры.

6. `<VAR>...</VAR>`

От английского variable – переменная. Рекомендуется использовать для написания имен переменных.

2.4 Форматирование абзацев

Для задания фрагменту текста некоторых параметров абзаца используется тег `<p>...</p>`. Этот тег оформляет все заключенные в него элементы в абзац.

У данного тега есть атрибут *align*, который управляет выравниванием текста. Данный атрибут может принимать следующие значения: *right*, *left*, *center*, *justify* – соответственно выравнивание по правому, левому краям страницы, по центру и по ширине.

Смысл абзаца в HTML состоит в том, что между двумя соседними абзацами автоматически устанавливается некоторый интервал, больший чем междустрочный интервал внутри данного абзаца. Если же у нас отсутствует необходимость визуально различать абзацы в окне браузера, однако, необходимо, чтобы каждое из группы слов начиналось с новой строки, то вместо тега `<p>...</p>` используется тег `
` – тег перехо-



Кафедра
АГуММ

Начало

Содержание



Страница 28 из 187

Назад

На весь экран

Закрыть

да на следующую строку (закрытия не требует), при этом сохраняется выравнивание, указанное в теге `<p>`.

Как было отмечено ранее, браузеры игнорируют в HTML-файлах символы абзаца и множественные пробелы. Однако, текст заключенный в теге `<PRE>...</PRE>` (от английского *preformatted*), выводится браузером на экран как есть – со всеми пробелами, символами табуляции и абзаца.

Для оформления больших цитат, вынесенных в отдельный абзац используется тег `<BLOCKQUOTE>...</BLOCKQUOTE>`. Текст, заключенный внутри этого тега, отображается (большинством браузеров) с отступами как с левой так и с правой стороны, таким образом выделяя его содержимое.

2.5 Заголовки

По аналогии с текстовым процессором Microsoft Word в HTML поддерживаются встроенные стили заголовков. Они определяются тегом `<h#>`, где # – число от 1 до 6.

Так же, как и для тега абзаца, для заголовков применяется атрибут *align*, управляющий выравниванием текста заголовка.



Кафедра
АГММ

Начало

Содержание



Страница 29 из 187

Назад

На весь экран

Закрыть

2.6 &-последовательности

Из правил применения HTML-разметки вытекает логичный вопрос: как показать на экране в окне браузера специальные символы (например, $<$ ("меньше") или $>$ ("больше"))?

В HTML это делается с помощью *Э-последовательностей* (их еще называют *символьными объектами* или *эскейп-последовательностями*). Например,

- символ $<$ (меньше) обозначается литералом *Эlt*; (по первым буквам английских слов less than – меньше, чем);
- символ $>$ (больше) обозначается литералом *Эgt*; (по первым буквам английских слов greater than – больше, чем);
- символ $&$ (амперсанд) обозначается литералом *Эamp*;;
- символ $\$$ обозначается литералом *Э#167*;;
- двойные кавычки $"$ обозначаются литералом *Эquot*;;

Замечание! Точка с запятой является обязательным элементом &-последовательности. Кроме того, все буквы, составляющие последовательность, должны быть строчными (маленькими).



Кафедра
АГММ

Начало

Содержание



Страница 30 из 187

Назад

На весь экран

Закреть

2.7 Комментарии

Комментарии HTML вставляются в код с помощью парного тега вида:

```
<!--  
    <текст комментария>  
-->
```

2.8 Графика

Современные веб-страницы невозможно представить без графического оформления. Графические изображения в HTML добавляются двумя способами: непосредственная вставка рисунка и использование рисунка в качестве фона (фоновый рисунок или, иначе говоря, текстура).

Большинство цветных изображений и фонов в Сети являются файлами в формате CompuServe GIF (*Graphics Interchange Format*). Формат GIF обладает двумя преимуществами – прозрачностью и анимацией, которая осуществляется циклическим чередованием изображений. Прозрачность (transparency) в GIF дает возможность использовать изображения неправильной формы на любом фоне. Прозрачность осуществляется за счет того, что какой-либо цвет (указываемый при экспорте файла) не показывается.

GIF целесообразно использовать для хранения:

- малоразмерных графических элементов: значков-ссылок, надписей



Кафедра
АГyMM

Начало

Содержание



Страница 31 из 187

Назад

На весь экран

Закрыть

и миниатюр;

- изображений, состоящих из больших областей однородной окраски и не имеющих множества цветов и тонких цветовых переходов.

Для полноцветных изображений в WWW используется формат JPEG (Joint Photographic Experts Group). JPEG сжимает графику фотографического качества и сохраняет в большой степени точность соответствия цветов. Это делает файлы JPEG меньшими по размеру, уменьшая время загрузки. Можно установить ту или иную степень сжатия JPEG-файла, но поскольку сжатие у данного формата приводит к потере качества, чем сильнее будет его степень, тем большее количество цветовой информации будет утеряно. JPEG целесообразно использовать для хранения полноцветных фотографий или образцов художественной графики, включающих тонкие градации цвета и имеющих размер более 200 пикселей по каждой из координат.

PNG (произносится как "пинг") (Portable Network Graphics). Этот формат предназначен специально для передачи изображений по сетям. Этот формат поддерживает сжатие без потерь, прозрачность цвета (transparency), хранение пользовательских данных (например, комментариев). Используется как заменитель GIF.

Чтобы поместить на страницу графическое изображение, следует использовать одинарный тег *IMG*.

Тег `` имеет обязательный атрибут `src="адрес файла картин-`



Кафедра
АГуММ

Начало

Содержание



Страница 32 из 187

Назад

На весь экран

Закрыть

ки", указывающий на местонахождения файла картинки. Если картинка находится в той же папке, что и html-файл, в который она добавляется, то можно указывать только полное имя файла (собственное имя файла и его тип; например: Вася.jpg или рисунок.gif). Если картинка находится не в одной папке с html-файлом, то необходимо указывать полный адрес картинки, например:

```
src="d:/distrib/Mou рисунки/car .jpg"
```

или

```
src="http://daemon.brsu.brest.by/images/webspid.gif" .
```

Желательно, при использовании нескольких графических файлов в процессе создания web-страницы, помещать данные файлы в подкаталог того каталога, в котором содержится сама html-страница. То есть, пусть у нас имеется папка А, в которой находится страница 1.htm. Создадим в этой папке А еще одну папку В, в которую и будем помещать все графические файлы, используемые в этой странице. Тогда тег вставки рисунка будет выглядеть следующим образом

```
<img SRC="B/имя файла" >.
```

Рассмотрим основные атрибуты тега ``.

- Атрибут `border` указывает ширину рамки вокруг картинки, задается в пикселах.



Кафедра
АГуММ

Начало

Содержание



Страница 33 из 187

Назад

На весь экран

Закрыть

- Атрибут *align* управляет горизонтальным выравниванием картинки на странице, этот же атрибут используется для вертикального выравнивания картинки относительно текста. Параметр "*выравнивание*" может принимать следующие значения:
 - *left* – горизонтальное выравнивание по левому краю окна;
 - *right* – горизонтальное выравнивание по правому краю окна;
 - *center* – горизонтальное выравнивание по центру;
 - *bottom* – вертикальное выравнивание по нижнему краю относительно текста абзаца;
 - *middle* – вертикальное выравнивание по центру относительно текста;
 - *top* – вертикальное выравнивание по верхнему краю изображения относительно текста абзаца.
- Атрибут *hspace* указывает горизонтальные отступы справа и слева от картинки. Задается в пикселах.
- Атрибут *vspace* указывает вертикальный отступ сверху и снизу от картинки. Задается в пикселах.
- Атрибут *alt* устанавливает сообщение, которую выведет браузер, если по каким-либо причинам картинка не загрузится. В качестве сообщения может выступать любая фраза.



Кафедра
АГyMM

Начало

Содержание



Страница 34 из 187

Назад

На весь экран

Заккрыть

- Атрибут *width* задает ширину картинки в пикселах или процентах.
- Атрибут *height* задает высоту картинки в пикселах или в процентах.

Применение *width* и *height* атрибутов ускоряет загрузку страницы. Когда применяются эти атрибуты, браузер знает сколько места оставить под картинку и сначала загружает текст. Так же эти атрибуты применяются для изменения размеров картинки. Если мы зададим только один параметр – высоту (ширину), – то картинка изменит свои размеры с сохранением пропорций относительно исходного размера. В частности, в процентах рекомендуется задавать один параметр. Данный вариант задания параметров высоты (ширины) оптимизирует картинку под любой размер окна браузера, то есть, в случае, если мы задали `height=50%`, наша картинка будет занимать половину окна браузера по высоте невзирая на размеры окна.

Напомним, что у тега *body* есть атрибут `BACKGROUND="имя файла"`, который помещает некоторый рисунок в качестве фона (заполняет множеством этих рисунков все пространство окна браузера). Правила задания имени фонового рисунка точно такие же, как и при использовании атрибута *src* в теге вставки картинки.



Кафедра
АГУММ

Начало

Содержание



Страница 35 из 187

Назад

На весь экран

Заккрыть

2.9 Гиперссылка

Как уже упоминалось в самом начале, сокращение HTML означает "язык гипертекстовой разметки". В отличие от обыкновенного текста, который можно читать только от начала к концу, гипертекст позволяет осуществлять мгновенный переход от одного фрагмента текста к другому. Системы помощи многих популярных программных продуктов устроены именно по гипертекстовому принципу. При нажатии левой кнопкой мыши на некоторый выделенный фрагмент текущего документа происходит переход к некоторому заранее назначенному документу или фрагменту документа.

Гиперссылка создается с помощью особого парного тега `<A>...`. Внутри этого тега помещается текст, который станет собственно гиперссылкой. Тег `<A>` имеет обязательный атрибут `HREF`, задающий адрес перехода:

`<A HREF="адрес перехода" произвольный текст `.

В качестве параметра "адрес перехода" может использоваться несколько типов аргументов. Самое простое – это задать имя другого HTML-документа, к которому нужно перейти. Например:

` Перейти к документу 1 `

Фраза *Перейти к документу 1* в документе будет выделена цветом и подчеркиванием. При нажатии на нее левой кнопкой мыши в текущее окно будет загружен документ 1.htm.



Кафедра
АГММ

Начало

Содержание



Страница 36 из 187

Назад

На весь экран

Закрыть

Обратите внимание: если в адресе перехода не указан путь к файлу, переход будет выполнен внутри той папки, в которой содержится исходный html-документ.

Можно создать ссылку перехода, скажем, на какой-либо графический файл. Для этого необходимо указать его в качестве адреса перехода. Например:

```
<a href="dog.jpg">Перейти к рисунку</a>
```

Кроме того, в качестве объекта ссылки может выступать не только какой-то текст, но и графическое изображение. Просто мы должны тег вставки картинки заключить в тег ссылки.

И это еще не все. Наиболее интересным является тот факт, что существует возможность создать гиперссылку используя какой-либо фрагмент графического изображения. Для этого используется парный тег `<MAP>...</MAP>`. Данный тег позволяет задать некоторую область, которую можно затем использовать в теге вставки картинки. При этом размеры картинки должны быть заданы в абсолютном виде, то есть в пикселах.

В теге гиперссылки появляется атрибут `shape=poly`. В данном случае в качестве гиперссылки будет выступать некая область, ограниченная замкнутой полилинией, координаты узлов которой определяются атрибутом `coords=x1,y1,x2,y2,...,xn,yn`. Отсчет производится от левой границы картинки (для x) и от верхней границы (для y). Соот-



Кафедра
АГММ

Начало

Содержание



Страница 37 из 187

Назад

На весь экран

Закрыть

ветственно, тег вставки графического объекта дополняется командой *useMap=#FPMapl*, где FPMapl – имя, заданное в теге <map> атрибутом *name*.

Кстати, область подобной гиперссылки визуально определить нельзя. Для ее нахождения необходимо поводить курсором мыши по картинке, и там, где он видоизменится в "палец", и будет область гиперссылки.

При необходимости можно задать переход не просто к некоторому документу, но и к определенному месту внутри этого документа. Для этого необходимо создать в документе, к которому будет задан переход, некоторую опорную точку, или *анкер*. Разберем это на примере.

Допустим, что необходимо осуществить переход из файла 1.htm к словам "Переход закончен" в файле 2.htm (файлы находятся в одном каталоге). Прежде всего, необходимо создать вот такой анкер в файле 2.htm:

```
<A NAME="AAA» Переход закончен </A>.
```

Слова *Переход закончен* при этом никак не будут выделены в тексте документа.

Затем в файле 1.htm (или в любом другом) можно определить переход на этот анкер:

```
<A HREF="2.htm#AAA» переход к анкеру AAA </A>.
```

Кстати говоря, переход к этому анкеру можно определить и внутри самого документа 2.htm – достаточно только включить в него вот такой фрагмент:



Кафедра
АГММ

Начало

Содержание



Страница 38 из 187

Назад

На весь экран

Заккрыть

`` переход к анкеру AAA ``.

На практике это очень удобно при создании больших документов. В начале документа можно поместить оглавление, состоящее из ссылок на анкеры, расположенные в заголовках разделов документа.

Во избежание недоразумений рекомендуется задавать имена анкеров латинскими буквами. Следите за написанием имен анкеров: большинство браузеров отличают большие буквы от маленьких. Если имя анкера определено как AAA, ссылка на анкер aaa или AaA не выведет Вас на анкер AAA, хотя документ, скорее всего, будет загружен корректно.

В теге гиперссылки может также присутствовать атрибут *target*, который указывает, куда должна загружаться вызываемая страничка. Если параметр *target*="Название окна" указать равным blank, то браузер загрузит вызываемую страничку в новое окно. Если этот атрибут опустить, то новая страничка загрузится в текущее окно. К этому атрибуту мы вернемся, при рассмотрении фреймов.

2.10 Списки

В HTML существует 3 вида списков: маркированные, нумерованные и многоуровневые (списки определений) списки.

- `... `



Кафедра
АГММ

Начало

Содержание



Страница 39 из 187

Назад

На весь экран

Закреть

Маркированный список. В маркированном списке каждый элемент списка выводится с новой строки, имеет некоторый отступ слева и помечен маркером. Отдельные элементы отделяются друг от друга тегом `...` (рис. 2.1).

<i>Пример:</i>	<i>В окне браузера:</i>
<code></code>	
<code>это</code>	● это
<code>маркированный</code>	● маркированный
<code>список</code>	● список
<code></code>	

Рис. 2.1: Маркероанный список

- `...`

Нумерованный список. Каждый элемент нумерованного списка имеет свой порядковый номер и выводится с новой строки. Нумерация строк происходит автоматически. Так же, как и для маркированного списка, отдельные элементы отделяются друг от друга тегом `...` и имеет место левый отступ (рис. 2.2).

- `<DL>...</DL>`

Список определений. Список определений отличается от других видов списков тем, что имеет несколько уровней отступов слева. Тер-



Кафедра
АГММ

Начало

Содержание



Страница 40 из 187

Назад

На весь экран

Закрыть

<i>Пример:</i>	<i>В окне браузера:</i>
<code></code>	
<code>это</code>	1. это
<code>нумерованный</code>	2. нумерованный
<code>список</code>	3. список
<code></code>	

Рис. 2.2: Нумерованный список

мины, которым нужно дать определения, заключаются в тег `<dt>...</dt>`, а сами определения заключены в тег `<dd>...</dd>` (рис. 2.3).

<i>Пример:</i>	<i>В окне браузера:</i>
<code><dl></code>	
<code><dt>первый термин</dt></code>	первый термин
<code><dd>это определение</dd></code>	это определение
<code><dt>второй термин</dt></code>	второй термин
<code><dd>это определение</dd></code>	это определение
<code><dt>третий термин</dt></code>	третий термин
<code><dd>это определение</dd></code>	это определение
<code></dl></code>	

Рис. 2.3: Список определений

Если определяемые термины достаточно коротки, для списков определений можно использовать атрибут *compact* (рис. 2.4).



Кафедра
АГММ

Начало

Содержание



Страница 41 из 187

Назад

На весь экран

Заккрыть

<i>Пример:</i>	<i>В окне браузера:</i>	
<code><dl compact></code>		
<code><dt>A</dt></code>		
<code><dd>Первая буква алфавита</dd></code>	A	Первая буква алфавита
<code><dt>B</dt></code>		
<code><dd>Вторая буква алфавита</dd></code>	B	Вторая буква алфавита
<code><dt>B</dt></code>		
<code><dd>Третья буква алфавита</dd></code>	B	Третья буква алфавита
<code></dl></code>		

Рис. 2.4: Список определений

Элемент любого списка может содержать в себе список любого вида. Число уровней вложенности, в принципе, не ограничено, однако злоупотреблять вложенными списками все же не следует.

2.11 Таблицы

До настоящего времени мы имели дело с документами, в которых существовал только один "поток" текста. На практике иногда очень хочется расположить текст в несколько колонок. Таблица может в этом помочь. Кроме того, таблица, состоящая из одной ячейки, может очень эффектно выделить фрагмент текста, на который Вы хотите обратить внимание читателя.

Таблица объявляется тегом `<TABLE>...</TABLE>`. Кроме того, метка `<table>` может включать несколько атрибутов:



Кафедра
АГуММ

Начало

Содержание



Страница 42 из 187

Назад

На весь экран

Закрыть



- Атрибут *ALIGN* устанавливает расположение таблицы по отношению к полям документа. Допустимые значения: *align=left* (выравнивание влево), *align=center* (выравнивание по центру), *align=right* (выравнивание вправо).
- Атрибут *WIDTH* задает Ширину таблицы. Ее можно задать в пикселах (например, *width=400*) или в процентах от ширины страницы (например, *width=80%*).
- Атрибут *BORDER* устанавливает ширину внешней рамки таблицы и ячеек в пикселах (например, *border=4*). Если атрибут не установлен, таблица показывается без рамки.
- Атрибут *CELLSPACING* устанавливает расстояние между рамками ячеек таблицы в пикселах (например, *cellspacing=2*).
- Атрибут *CELLPADDING* устанавливает расстояние между рамкой ячейки и текстом в пикселах (например, *cellpadding=10*).
- Атрибут *bordercolordark="цвет"* задает цвет верхней и левой стороны рамки таблицы.
- Атрибут *bordercolorlight="цвет"* задает цвет нижней и правой стороны таблицы.

- *BORDERCOLOR*. Если вместо `bordercolordark` и `bordercolorlight` поставить один атрибут `bordercolor="цвет"`, то вся рамка таблицы будет нарисована одним цветом.
- *BGCOLOR*. С помощью атрибута `bgcolor="цвет"` можно задать цвет фона для отдельной строки, ячейки или для всей таблицы.
- *BACKGROUND*. Если нам необходимо в качестве фона иметь картинку, то в этом нам поможет атрибут `background="имя графического файла"`. Этот атрибут можно использовать с тегами `<table>`, `<tr>` и `<td>`.

Таблица может иметь заголовок, который задается тегом

`<CAPTION>...</CAPTION>`.

Однако, заголовок не является обязательным. Метка `<caption>` может включать атрибут `align`. Допустимые значения: `<caption align=top>` (заголовок помещается над таблицей) и `<caption align=bottom>` (заголовок помещается под таблицей).

Каждая строка таблицы заключена в тег `<TR>...</TR>`. Он может включать следующие атрибуты:

- Атрибут *ALIGN* устанавливает выравнивание текста в ячейках строки. Допустимые значения: `align=left` (выравнивание влево),



Кафедра
АГУММ

Начало

Содержание



Страница 44 из 187

Назад

На весь экран

Заккрыть

align=center (выравнивание по центру), *align=right* (выравнивание вправо).

- Атрибут *VALIGN* устанавливает вертикальное выравнивание текста в ячейках строки. Допустимые значения: *valign=top* (выравнивание по верхнему краю), *valign=middle* (выравнивание по центру), *valign=bottom* (выравнивание по нижнему краю).

Каждая ячейка таблицы задается тегом `<TD>... </TD>`. Метка `<td>` может включать следующие атрибуты:

- *NOWRAP*. Присутствие этого атрибута означает, что содержимое ячейки должно быть показано в одну строку.
- *COLSPAN*. Данный атрибут устанавливает "размах" ячейки по горизонтали. Например, *colspan=3* означает, что ячейка простирается на три колонки.
- Атрибут *ROWSPAN* устанавливает "размах" ячейки по вертикали. Например, *rowspan=2* означает, что ячейка занимает две строки.
- Атрибут *ALIGN* устанавливает выравнивание текста в ячейке. Допустимые значения: *align=left* (выравнивание влево), *align=center* (выравнивание по центру), *align=right* (выравнивание вправо).



Кафедра
АГУММ

Начало

Содержание



Страница 45 из 187

Назад

На весь экран

Закрыть



- Атрибут *VALIGN* устанавливает вертикальное выравнивание текста в ячейке. Допустимые значения: *valign=top* (выравнивание по верхнему краю), *valign=middle* (выравнивание по центру), *valign=bottom* (выравнивание по нижнему краю).
- Атрибут *WIDTH* устанавливает ширину ячейки в пикселах (например, *width=200*).
- Атрибут *HEIGHT* устанавливает высоту ячейки в пикселах (например, *height=40*).

Если ячейка таблицы пуста, вокруг нее не рисуется рамка. Если ячейка пуста, а рамка нужна, в ячейку можно ввести символьный объект * *; (non-breaking space – неразрывный пробел). Ячейка попрежнему будет пустой, а рамка вокруг нее будет.

Следует отметить, что любая ячейка таблицы может содержать в себе другую таблицу.

Рассмотрим несколько примеров.

Пример 1. Управление размерами

```
<table border="10"width="100%">
  <tr>
    <td width="20%">width="20%"</td>
```

```
</tr>  
</table>
```

В браузере отобразится таблица, изображенная на рис. 2.6.

align="left"	align="center"	align="right"
valign="top"	valign="middle"	valign="bottom"

Рис. 2.6:

Пример 3. Отступы

```
<table border="1"width="100%"cellpadding="10"cellspacing="10">  
  <tr>  
    <td width="33%"> Ячейка 1 </td>  
    <td width="33%"> Ячейка 2 </td>  
    <td width="34%"> Ячейка 3</td>  
  </tr>  
</table>
```

В браузере отобразится таблица, изображенная на рис. 2.7.



Кафедра
АГуММ

Начало

Содержание



Страница 48 из 187

Назад

На весь экран

Закреть

Ячейка 1	Ячейка 2	Ячейка 3
----------	----------	----------

Рис. 2.7:

Пример 4. Сложные таблицы

```
<table border="1">  
  <tr>  
    <td rowspan="2">rowspan="2"</td>  
    <td colspan="2">colspan="2"</td>  
  </tr>  
  <tr>  
    <td>ячейка</td>  
    <td>ячейка</td>  
  </tr>  
  <tr>  
    <td>ячейка</td>  
    <td>ячейка</td>  
    <td>ячейка</td>  
  </tr>  
</table>
```

В браузере отобразится таблица, изображенная на рис. 2.8.



Кафедра
АГММ

Начало

Содержание



Страница 49 из 187

Назад

На весь экран

Закреть

rowspan="2"	colspan="2"	
	ячейка	ячейка
ячейка	ячейка	ячейка

Рис. 2.8:

Пример 5. Цветное оформление таблицы

```

<table cellspacing="10"border="5"
bordercolordark="black"
bordercolorlight="gray">
  <tr bgcolor="blue">
    <td>строка</td>
    <td>с голубым</td>
    <td>фоном</td>
  </tr>
  <tr>
    <td bgcolor="red">красный фон</td>
    <td bgcolor="yellow">желтый фон</td>
    <td bgcolor="green">зеленый фон</td>
  </tr>
  <tr>

```



Кафедра
АГММ

Начало

Содержание



Страница 50 из 187

Назад

На весь экран

Закрыть

```
<td colspan="3"background="Wood2-small.jpg">Ячейка с фо-  
новым рисунком </td>  
</tr>  
</table>
```

В браузере отобразится таблица, изображенная на рис. 2.9.

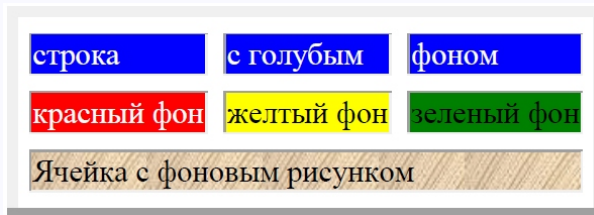


Рис. 2.9:

2.12 Фреймы

2.12.1 Описание фреймов

В HTML существует возможность разделить основное окно, в котором отображаются веб-страницы, на несколько частей, которые называются *фреймами*. Фрейм, по сути, представляет собой окно, в которое загружается отдельная страница. В результате в одном окне отображается сразу несколько страниц. В принципе, для того, чтобы создать окно



Кафедра
АГyMM

Начало

Содержание



Страница 51 из 187

Назад

На весь экран

Закрыть

с фреймовой структурой, необходимо создать новый html-документ, который и будет объявлять эту структуру. Для описания структуры фреймоф используется тег `<frameset>`. Записывается структура фреймов в заголовке html-документа (внутри тега `<head>...</head>`).

```
<html>
  <head>
    <frameset rows="20%,*">
      <frame name="frame1"src="1.htm">
      <frame name="frame2"src="2.htm">
    </frameset>
  </head>
</html>
```

Данный пример описывает фреймовую структуру, изображенную на рис. 2.10

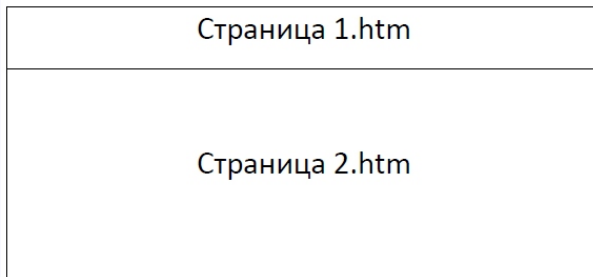


Рис. 2.10:



Кафедра
АГММ

Начало

Содержание



Страница 52 из 187

Назад

На весь экран

Заккрыть

Для тега `<frameset>` обязателен один из двух атрибутов:

- `rows="список размеров фреймов"` делит окно по вертикали;
- `cols="список размеров фреймов"` делит окно по горизонтали.

Размеры в параметре `"список размеров фреймов"` могут задаваться в процентах или в пикселах. Размеры перечисляются через запятую. Сколько размеров столько и фреймов. Если вместо размера указать * (задается в качестве последнего размера), то соответствующий фрейм займет все оставшееся пространство.

Тег `<frame>` определяет свойства отдельного фрейма, на которые делится окно браузера. Используется с обязательным атрибутом `src="файл-источник фрейма"`, который устанавливает источник для фрейма. Атрибут `name="имя фрейма"` устанавливает имя фрейма, которое потом может использоваться с атрибутом `target` тега гиперссылки.

Если имя фрейма `"frame1"`, то гиперссылка, загружающая страничку в этот фрейм будет выглядеть так:

```
<a href="123.htm"target="frame1">
```

ссылка, загружающая страницу в первый фрейм

```
</a>
```

Данная запись ссылки возможна в том случае, когда страница, содержащая эту гиперссылку, включена во фреймовую структуру, то есть



Кафедра
АГММ

Начало

Содержание



Страница 53 из 187

Назад

На весь экран

Закрыть

сама загружается в один из фреймов. Если же указать *target="top"*, то новая страница загрузится в текущее окно, поверх всех фреймов.

2.12.2 Сложные фреймовые структуры

Фреймы могут быть вложены друг в друга. Это позволяет создавать сложную структуру фреймов с вертикальным и горизонтальным делением. Например:

```
<html>
<head>
  <frameset rows="20%, 60%, *">
    <frame name="frame1"src="1.htm">
      <frameset cols="50%,50%">
        <frame name="frame2"src="2.htm">
          <frame name="frame3"src="3.htm">
        </frameset>
      <frame name="frame4"src="4.htm">
    </frameset>
  </head>
</html>
```

Данная фреймовая структура изображена на рис. 2.11.

Теперь разберем некоторые атрибуты, используемые в сочетании с тегами-объявителями фреймов.



Кафедра
АГММ

Начало

Содержание



Страница 54 из 187

Назад

На весь экран

Заккрыть



Рис. 2.11:

- *MARGINWIDTH*="отступ по горизонтали".

Этот атрибут задает значение горизонтальных отступов от края страницы, для страниц, отображаемых в заданном фрейме. Параметр отступ по горизонтали задается в пикселах. Используется с тегами: `<frame>`, `<iframe>`.

- *MARGINHEIGHT*="отступ по вертикали".

То же самое, что и *marginwidth*, но действует по вертикали.

- *SCROLLING*="режим".

Атрибут, управляющий отображением линеек прокрутки (скроллингов).

Параметр режим может принимать следующие значения:

- *yes* – всегда изображать скроллинги;



Кафедра
АГyMM

Начало

Содержание



Страница 55 из 187

Назад

На весь экран

Закрыть

- *no* – никогда не изображать скроллинги;
- *auto* – автоматическое изображение скроллингов.

Используется с тегами: `<frame>`, `<iframe>`.

- ***NORESIZE***.

Устанавливает запрет на изменение размеров фреймов с помощью мышки. Используется с тегами: `<frame>`, `<iframe>`.

- ***FRAMEBORDER*** = "*режим*".

Управляет отображением границ фреймов. Параметр *режим* может принимать следующие значения:

- *yes* или *1* – отображать границу фреймов,
- *no* или *0* – не отображать границу фреймов

- ***FRAMESPACING*** = "*ширина*".

Задание ширины границ фреймов. Параметр *ширина* задается в пикселах. Используется с тегами: `<frameset>`, `<iframe>`.

Тег `<iframe>` создает плавающий фрейм, который находится внутри обычного документа, он позволяет загружать в область заданных размеров любые другие независимые документы.

Тег `<iframe>...</iframe>` является контейнером, содержание которого игнорируется браузерами, не поддерживающими данный тег. Для



Кафедра
АГуММ

Начало

Содержание



Страница 56 из 187

Назад

На весь экран

Закрыть

таких браузеров можно указать альтернативный текст, который увидят пользователи. Он должен располагаться между элементами `<iframe>` и `</iframe>`.

Рассмотрим некоторые атрибуты тега `<iframe>`:

- *align* определяет как фрейм будет выравниваться по краю, а также способ обтекания его текстом;
- *allowtransparency* устанавливает прозрачный фон фрейма, через который виден фон страницы;
- *frameborder* устанавливает, отображать границу вокруг фрейма или нет;
- *height* задает высоту фрейма;
- *width* задает ширину фрейма;
- *hspace* – горизонтальный отступ от фрейма до окружающего контента;
- *vspace* – вертикальный отступ от фрейма до окружающего контента.



Кафедра
АГyMM

Начало

Содержание



Страница 57 из 187

Назад

На весь экран

Заккрыть

2.13 Формы

2.13.1 Задание формы

Форма – это особый элемент страницы, реализующий ввод, подготовку и отправку данных другой странице или специальной программе, обрабатывающей введенные посетителем данные. Форма (или Web-форма) содержит набор элементов управления – полей ввода, списков, флажков, переключателей и пр., – в которые вводятся данные.

Форма открывается меткой `<form>` и заканчивается меткой `</form>`. HTML-документ может содержать в себе несколько форм, однако формы не должны находиться одна внутри другой. HTML-текст, включая метки, может размещаться внутри форм без ограничений.

Метка `<form>` может содержать три атрибута, один из которых является обязательным.

- *ACTION*.

Обязательный атрибут. Определяет, где находится обработчик формы.

- *METHOD*.

Определяет, каким образом данные из формы будут переданы обработчику. Допустимые значения: *method=post* и *method=get*. Если значение атрибута не установлено, по умолчанию предполагается *method=get*.



Кафедра
АГуММ

Начало

Содержание



Страница 58 из 187

Назад

На весь экран

Заккрыть

- *ENCTYPE*.

Определяет, каким образом данные из формы будут закодированы для передачи обработчику. Если значение атрибута не установлено, по умолчанию предполагается *enctype=application/x-www-form-urlencoded*.

2.13.2 Элементы управления

Формы передают информацию программам-обработчикам в виде пар *[имя переменной]=[значение переменной]*. Имена переменных следует задавать латинскими буквами. Значения переменных воспринимаются обработчиками как строки, даже если они содержат только цифры.

Для того, чтобы запустить процесс передачи данных из формы обработчику, нужен какой-то орган управления.

Элементы управления создаются с помощью разных тегов HTML. Это тег *INPUT*, применяемый для создания большинства элементов управления, теги *TEXTAREA*, *SELECT* и *OPTION*.

Рассмотрим подробнее элементы управления, предлагаемые языком HTML, и теги которые их создают.

Кнопка

Создается с помощью тега *<INPUT TYPE=SUBMIT>*. Встретив такую строчку внутри формы, браузер нарисует на экране кнопку с над-



Кафедра
АГyMM

Начало

Содержание



Страница 59 из 187

Назад

На весь экран

Закреть

писью *Submit*, при нажатии на которую все имеющиеся в форме данные будут переданы обработчику, определенному в метке `<form>`.

Атрибуты:

- *VALUE* – задает надпись на кнопке (от английского "значение").

```
<input type=submit value="Жми сюда">
```

Данный пример создаст на форме кнопку "Жми сюда" (Рис. 2.12)

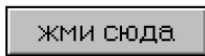


Рис. 2.12:

Простейшую форму можно использовать в качестве гиперссылки. Подобная форма не будет собирать никаких данных, а просто приведет нас к заданной странице.

```
<html>
  <head>
    <title>Пример</title>
  </head>
  </body>
    <h1>Простейшая форма </h1>
```



Кафедра
АГММ

Начало

Содержание



Страница 60 из 187

Назад

На весь экран

Заккрыть



```
<form action="1.htm"><!--Это начало формы-->  
<input type=submit value="Перейти на страницу №1">  
</form> <!--Это конец формы-->  
</body>  
</html>
```

- *NAME* - позволяет надпись, нанесенную на кнопку, передать обработчику путем введения в определение кнопки атрибута, например:

```
<input type=submit name=button value="жми сюда">
```

При нажатии на такую кнопку обработчик вместе со всеми остальными данными получит и переменную *button* со значением *жми сюда*.

В форме может быть несколько кнопок типа *submit* с различными именами и/или значениями. Обработчик, таким образом, может действовать по-разному в зависимости от того, какую именно кнопку *submit* нажал пользователь.

Существуют и другие типы элементов *<input>*. Каждый элемент *<input>* должен включать атрибут *name=имя*, определяющий имя элемента (и, соответственно, имя переменной, которая будет передана обработчику). Имя должно задаваться только латинскими буквами. Большинство элементов *<input>* должны включать атрибут *value="значение"*,

определяющий значение, которое будет передано обработчику под этим именем. Для элементов `<input type=text>` и `<input type=password>`, однако, этот атрибут не обязателен, поскольку значение соответствующей переменной может вводиться пользователем с клавиатуры.

Поле ввода

Поле ввода текста создается с помощью одинарного тега `INPUT TYPE=`*text*. Может содержать дополнительные атрибуты:

- `SIZE=число` – задает ширину окна ввода в символах;
- `MAXLENGTH=число` – задает максимально допустимую длину вводимой строки в символах.

```
<input type=text size=20 name=user value="имя">
```

Определяет окно шириной 20 символов для ввода текста (рис. 2.13). По умолчанию в окне находится текст имя, который пользователь может редактировать. Отредактированный (или неотредактированный) текст передается обработчику в переменной user.



A screenshot of a text input field. The field is rectangular with a thin border and contains the text "ИМЯ" in a simple, sans-serif font. The text is positioned at the beginning of the field, and there is a small vertical cursor or focus indicator at the start of the text.

Рис. 2.13:



Кафедра
АГУММ

Начало

Содержание



Страница 62 из 187

Назад

На весь экран

Закреть

Поле ввода пароля

Поле для ввода пароля создается тегом `<INPUT TYPE=password>`. Абсолютно аналогичен типу `text`, только вместо символов вводимого текста показывает на экране звездочки (*).

```
<input type=password name=pw size=20 maxlength=10>
```

Данный пример определяет окно шириной 20 символов для ввода пароля. Максимально допустимая длина пароля – 10 символов. Введенный пароль передается обработчику в переменной `pw`.

Переключатель

Переключатель задается одинарным тегом `<INPUT TYPE=radio>`. Может содержать дополнительный атрибут `checked`, который показывает, что кнопка помечена. В группе радиокнопок с одинаковыми именами может быть только одна помеченная радиокнопка.

```
<input type=radio name=modem value="9600"checked> 9600 бум/с  
<input type=radio name=modem value="14400">14400 бум/с  
<input type=radio name=modem value="28800">28800 бум/с
```

В данном примере определяется группа из трех радиокнопок (рис. 2.14), подписанных 9600 бит/с, 14400 бит/с и 28800 бит/с. Первоначально помечена первая из кнопок. Если пользователь не отметит другую кнопку,



Кафедра
АГММ

Начало

Содержание



Страница 63 из 187

Назад

На весь экран

Закрыть

обработчику будет передана переменная `modem` со значением 9600. Если пользователь отметит другую кнопку, обработчику будет передана переменная `modem` со значением 14400 или 28800.

9600 бит/с 14400 бит/с 28800 бит/с

Рис. 2.14:

Флажок

Для создания флажков применяется тег `<INPUT TYPE=checkbox>`. Может содержать дополнительный атрибут `checked`, который показывает, что квадрат помечен. В отличие от радиокнопок, в группе квадратов с одинаковыми именами может быть несколько помеченных квадратов.

`<input type=checkbox name=comp value="pc">` персональные компьютеры

`<input type=checkbox name=comp value="ws"checked>` рабочие станции

`<input type=checkbox name=comp value="lan">` серверы локальных сетей

`<input type=checkbox name=comp value="is"checked>` серверы интернет



Кафедра
АГММ

Начало

Содержание



Страница 64 из 187

Назад

На весь экран

Заккрыть

В данном примере определяется группа из четырех квадратов (флажков) (рис. 2.15). Первоначально помечены второй и четвертый квадраты. Если пользователь не произведет изменений, обработчику будут переданы две переменные: *comp=ws* и *comp=is*.

<input type="checkbox"/>	персональные компьютеры	<input checked="" type="checkbox"/>	рабочие станции
<input type="checkbox"/>	серверы локальных сетей	<input checked="" type="checkbox"/>	серверы интернет

Рис. 2.15:

Скрытое поле

Определяет скрытый элемент данных, который не виден пользователю при заполнении формы и передается обработчику без изменений. Скрытое поле задается тегом `<INPUT TYPE=hidden>`. Такой элемент, как скрытое поле, иногда полезно иметь в форме, которая время от времени подвергается переработке, чтобы обработчик мог знать, с какой версией формы он имеет дело.

```
<input type=hidden name=version value="1.1">
```

Данный пример определяет скрытую переменную *version*, которая передается обработчику со значением 1.1.



Кафедра
АГММ

Начало

Содержание



Страница 65 из 187

Назад

На весь экран

Закреть

Кнопка сброса формы

Кнопка сброса формы - это кнопка, при нажатии на которую форма возвращается в исходное состояние. Она задается тегом `<INPUT TYPE=reset>`. Поскольку при использовании этой кнопки данные обработчику не передаются, кнопка типа `reset` может и не иметь атрибута `name`.

```
<input type=reset value="очистить форму">
```

В примере определяется кнопка *Очистить форму*, при нажатии на которую форма возвращается в исходное состояние.

Как отмечалось выше, помимо элементов `<input>`, формы могут содержать меню `<select>` и поля для ввода текста `<textarea>`.

Меню (список)

Списки, как обычные, так и раскрывающиеся, создаются с помощью парного тега `<SELECT>...</SELECT>`. Метка `<select>` содержит обязательный атрибут `name`, определяющий имя переменной, которую генерирует меню.

Элемент меню определяется тегом `<OPTION>`. Обязательный атрибут `value` устанавливает значение, которое будет передано обработчику, если выбран этот элемент меню. Тег `<option>` может включать атрибут `checked`, показывающий, что данный элемент отмечен по умолчанию.



Кафедра
АГУММ

Начало

Содержание



Страница 66 из 187

Назад

На весь экран

Заккрыть

```
<select name="имя">  
<option value="значение 1"> текст 1  
<option value="значение 2"> текст 2  
...  
<option value="значение n"> текст n  
</select>
```

Данный пример генерирует выпадающий список (рис. 2.16).

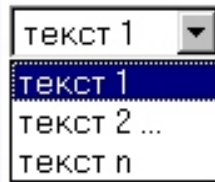


Рис. 2.16:

Метка `<select>` может также содержать атрибут *multiple*, присутствие которого показывает, что из меню можно выбрать несколько элементов. Большинство браузеров показывают меню `<select multiple>` в виде окна, в котором находятся элементы меню (высоту окна в строках можно задать атрибутом *size=число*). Меню `<select>` в большинстве случаев показывается в виде выпадающего меню.



Кафедра
АГyMM

Начало

Содержание



Страница 67 из 187

Назад

На весь экран

Заккрыть

Область редактирования

Область редактирования, задается парным тегом `<TEXTAREA>...</TEXTAREA>`.

```
<textarea name=address rows=5 cols=50>
```

а здесь – ваш адрес...

```
</textarea>
```

Все атрибуты обязательны.

- Атрибут *name* определяет имя, под которым содержимое окна будет передано обработчику.
- Атрибут *rows* устанавливает высоту окна в строках).
- Атрибут *cols* устанавливает ширину окна в символах.

Текст, заключенный в теге `<textarea>...</textarea>`, представляет собой содержимое окна по умолчанию. Пользователь может его отредактировать или просто стереть.

Важно знать, что русские буквы в окне `<textarea>` при передаче обработчику могут быть конвертированы в соответствующие им символьные объекты.



Кафедра
АГyMM

Начало

Содержание



Страница 68 из 187

Назад

На весь экран

Закрыть

РАЗДЕЛ 3

КАСКАДНЫЕ ТАБЛИЦЫ СТИЛЕЙ

3.1 CSS

Каскадные таблицы стилей (Cascading Style Sheets, CSS) — это стандарт, который определяет представление данных в браузере. Если HTML предоставляет информацию о структуре документа, то таблицы стилей сообщают как он должен выглядеть.

Основным понятием в CSS является – стиль. *Стиль* – это набор инструкций форматирования, которые можно применить к целому документу или к его части. Стиль включает все типы элементов дизайна: шрифт, фон, текст, цвета ссылок, поля и расположение объектов на странице.

Таблица стилей — это совокупность стилей, применимых к гипертекстовому документу.

Каскадирование — это порядок применения различных стилей к веб-странице. Браузер, который поддерживает таблицы стилей, последовательно применяет их в соответствии со следующим приоритетом:

- связанные стили;
- внедренные стили;
- встроенные стили.



Кафедра
АГуММ

Начало

Содержание



Страница 69 из 187

Назад

На весь экран

Заккрыть

Другим аспектом каскадирования является *наследование*, суть которого заключается в том, что если не указано иное, то конкретный стиль будет применен ко всем дочерним элементам гипертекстового документа. Например, если вы примените определенный цвет текста в теге `<div>`, то все теги внутри этого блока будут отображаться этим же цветом.

Использование каскадных таблиц дает возможность разделить содержимое и его представление и гибко управлять отображением гипертекстовых документов путем изменения стилей.

Таблицы стилей строятся в соответствии с определенным синтаксисом (рис. 3.1).

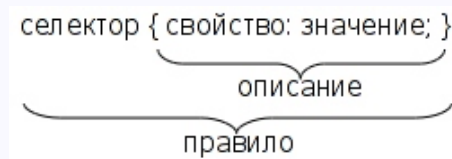


Рис. 3.1: Синтаксис описания стиля CSS

Селектор — это элемент, к которому будут применяться назначаемые стили. Селектором может быть тег, класс или идентификатор объекта html-документа.

Свойство определяет одну или несколько характеристик селектора. Свойства задают формат отображения селектора: отступы, шрифты, выравнивание, размеры и т.д.



Кафедра
АГУММ

Начало

Содержание



Страница 70 из 187

Назад

На весь экран

Закрыть

Значение — это фактические числовые или строковые константы, определяющие свойство селектора.

Совокупность свойств и их значений называется *описанием*. Полное описание стиля (селектор + описание) называется *правилом*.

Таким образом, таблица стилей — это набор правил, задающих значения свойств селекторов, перечисленных в этой таблице. Общий синтаксис описания правила выглядит так:

селектор {свойство: значение;}

При описании стилей регистр символов значения не имеет.

Приведем несколько примеров написания правил:

1. Основной текст с выравниванием по ширине, абзацный отступ 30px, гарнитура (шрифт) — Serif, кегль (размер шрифта) — 14px:

```
p {  
    text-align: justify;  
    text-indent: 30px;  
    font-family: Serif;  
    font-size: 14px;  
}
```

Это правило будет применено ко всем тегам <p>.

2. Синий цвет для заголовков с первого по третий уровень:



Кафедра
АГММ

Начало

Содержание



Страница 71 из 187

Назад

На весь экран

Заккрыть

```
h1, h2, h3 {  
  color: blue;  
}
```

Стандарт CSS представляет возможности создания именованных стилей — стилевых классов. Это позволяет применить разные стили к одному и тому же селектору.

Предположим, что в документе вам нужны два различных вида основного текста — один без отступа, второй — с левым отступом и шрифтом красного цвета. Для этого нужно создать правила для каждого из них. Например:

```
p {margin-left: 0;}  
p.warn {margin-left: 40px; color: FF00;}  
}
```

Для применения созданного класса его имя нужно указать в атрибуте *class* для выбранных абзацев:

```
<p class="warn" > Красный текст с отступом слева </p>
```

Общий синтаксис описания класса:

селектор.имя_класса {описание}



Кафедра
АГУММ

Начало

Содержание



Страница 72 из 187

Назад

На весь экран

Закрыть

При создании класса селектор можно не указывать (*.имя_класса {описание}*), тогда это правило можно применять к любому селектору, поддерживающему тот же набор свойств.

В качестве селектора может выступать идентификатор элемента гипертекста, указанный в атрибуте *id*. Для назначения стилей таким элементам используется синтаксис, аналогичный описанию классов, но вместо точки ставится знак # ("решетка"). Например:

```
div#content {  
    position: absolute;  
    top: 10px;  
    left: 10  
    right: 10  
    border: solid 1px silver;  
}  
...  
<div id="content" > Текст </div>
```

Следует помнить, что идентификаторы элементов должны быть уникальны в пределах документа.

Значения родственных свойств можно объединять (группировать). При этом таблица стилей становится более компактной, но предъявляются более жесткие требования к описанию правил. Группировка может



Кафедра
АГУММ

Начало

Содержание



Страница 73 из 187

Назад

На весь экран

Заккрыть

применяться для таких свойств, как *padding*, *font*, *border*, *background* и других.

Рассмотрим пример обычного стиля, задающего отступы:

```
div {  
    margin-left: 10px;  
    margin-top: 5px;  
    margin-right: 40px;  
    margin-bottom: 15px;  
}
```

Это же правило можно переписать с группировкой в следующем виде:

```
div {margin: 5px 40px 15px 10px; } /*порядок: top right bottom left*/
```

Оба стиля будут отображаться одинаково.

3.2 Использование CSS в HTML-страницах

Существует три способа применения таблицы стилей к HTML-документу:

- *Встраивание*. Этот метод позволяет применить стиль к заданному тегу. Встроенный стиль применяется к любому тегу с помощью атрибута *style*. Например:



Кафедра
АГуММ

Начало

Содержание



Страница 74 из 187

Назад

На весь экран

Закреть

`<p style="font: 12pt Courier" > Это текст с кеглем 12 точек и гарнитурой Courier </p>`

Встроенные стили полезны, когда необходима тонкая настройка отображения некоторого элемента страницы или небольшой web-страницы.

- *Внедрение*. Внедрение позволяет управлять стилями страницы целиком. Внедренные стили используют парный тег `<style>...</style>`, который обычно размещают в заголовке html-документа (`<head>...</head>`):
-

```
<html>
  <head>
    ...
    <style>
      правила CSS
    </style>
    ...
  </head>
  <body>
    ...
```



Кафедра
АГММ

Начало

Содержание



Страница 75 из 187

Назад

На весь экран

Заккрыть

- *Связывание*. Связанные, или внешние таблицы стилей — наиболее удобное решение, когда речь идет об оформлении целого сайта. Описание правил помещается в отдельный файл (обычно с расширением *.css*). С помощью тега `<link>` выполняется связывание этой таблицы стилей с каждой страницей, где ее необходимо применить. Например:

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

Любая страница, содержащая такую связь, будет оформлена в соответствии со стилями, указанными в файле *sample.css*.

3.3 Основные свойства CSS

Перечислим некоторые часто используемые свойства CSS и их назначение.

Параметры фона

- *background-color* — управление цветом фона. Здесь можно задать либо сам цвет в виде RGB-кода или символического наименования, либо значение *transparent*, чтобы задать элементу страницы прозрачный фон.



Кафедра
АГУММ

Начало

Содержание



Страница 76 из 187

Назад

На весь экран

Закреть

- *background-image* – позволяет использовать в качестве фона графическое изображение. Здесь можно задать либо адрес файла, либо значение *none*, убирающее графический фон (это поведение по умолчанию).
- *background-repeat* – повторение фоновой картинки. Здесь доступны следующие значения:
 - *repeat* – фоновое изображение будет повторяться по горизонтали и по вертикали (поведение по умолчанию);
 - *repeat-x* – фоновое изображение будет повторяться только по горизонтали;
 - *repeat-y* – фоновое изображение будет повторяться только по вертикали;
 - *no-repeat* – фоновое изображение не будет повторяться; часть фона элемента страницы останется не занятой изображением.
- *background-position-x* – позволяет задать горизонтальную позицию фонового изображения. Возможные значения:
 - *left* – фоновое изображение прижимается к левому краю элемента страницы;
 - *center* – фоновое изображение располагается по центру;



Кафедра
АГММ

Начало

Содержание



Страница 77 из 187

Назад

На весь экран

Заккрыть

- *right* – фоновое изображение прижимается к правому краю.
- *background-position-y* – позволяет задать вертикальную позицию фонового изображения. Возможные значения:
 - *top* – фоновое изображение прижимается к верхнему краю элемента страницы;
 - *center* – фоновое изображение располагается по центру;
 - *bottom* – фоновое изображение прижимается к нижнему краю.

Параметры шрифта

- *font-family* – задает начертание шрифта текста. Имена шрифтов задаются в виде "*Arial*", "*Times New Roman*", "*Verdana*" (кавычки обязательны).
- *font-size* – определяет размер шрифта. Размер шрифта может быть задан в пикселах *px*, в пунктах *pt*, в сантиметрах *cm*, в миллиметрах *mm* и др.
- *font-style* – задает начертание шрифта. Доступны три значения:
 - *normal* – обычный шрифт (значение по умолчанию);
 - *italic* – курсив;
 - *oblique*.



Кафедра
АГММ

Начало

Содержание



Страница 78 из 187

Назад

На весь экран

Заккрыть

- *font-weight* – устанавливает "жирность" шрифта. Здесь доступны девять абсолютных значений от 100 до 900, представляющих различную "жирность". При этом обычный шрифт будет иметь "жирность" 400 (или *normal*), а полужирный – 700 (*bold*).
- *color* – задает цвет текста. Значение цвета может быть задано как RGB-код, так и символьным наименованием цвета.
- *text-decoration* – задает "украшение", которое будет применено к тексту. Здесь будут доступны следующие значения:
 - *none* – убирает все "украшения" (это поведение по умолчанию);
 - *underline* – подчеркивает текст;
 - *overline* – "надчеркивает" текст;
 - *line-through* – зачеркивает текст.

Параметры абзаца

- *text-indent* – задает отступ для "красной строки".
- *text-align* – задает горизонтальное выравнивание текста. Доступны следующие значения:
 - *left* – выравнивание по левому краю (поведение по умолчанию);
 - *right* – выравнивание по правому краю;



Кафедра
АГММ

Начало

Содержание



Страница 79 из 187

Назад

На весь экран

Закрыть

- *center* – выравнивание по центру;
- *justify* – полное выравнивание.

Параметры списков

- *list-style-type* – задает вид маркера списка. Доступны следующие значения:
 - *disc* – черный кружок (поведение по умолчанию для маркированных списков);
 - *circle* – светлый кружок;
 - *square* – светлый квадратик;
 - *decimal* – арабская цифра (поведение по умолчанию для нумерованных списков);
 - *lower-roman* – маленькая римская цифра;
 - *upper-roman* – большая римская цифра;
 - *lower-alpha* – маленькая латинская буква;
 - *upper-alpha* – большая латинская буква;
 - *none* – маркер и нумерация отсутствуют (поведение по умолчанию для обычных абзацев).
- *list-style-image* – позволяет задать в качестве маркера пунктов списка какое-либо графическое изображение.



Кафедра
АГумМ

Начало

Содержание



Страница 80 из 187

Назад

На весь экран

Заккрыть

Параметры курсора

- *cursor* – устанавливает форму курсора мыши при наведении его на данный элемент страницы. Доступны следующие значения:
 - *hand* – "указывающий перст";
 - *crosshair* – перекрестье, "прицел";
 - *text* – текстовый курсор;
 - *wait* – песочные часы, обозначающие, что Windows "думу думает";
 - *default* – обычная стрелка;
 - *help* – обычная стрелка с вопросительным знаком справа;
 - *e-resize* – стрелка вправо, "на восток";
 - *ne-resize* – стрелка вверх-вправо, "на северо-восток";
 - *n-resize* – стрелка вверх, "на север";
 - *nw-resize* – стрелка вверх-влево, "на северо-запад";
 - *w-resize* – стрелка влево, "на запад";
 - *sw-resize* – стрелка вниз-влево, "на юго-запад";
 - *s-resize* – стрелка вниз, "на юг";
 - *se-resize* – стрелка вниз-вправо, "на юго-восток";



Кафедра
АГyMM

Начало

Содержание



Страница 81 из 187

Назад

На весь экран

Закреть

- *auto* – управление формой курсора передается web-обозревателю (поведение по умолчанию).

Параметры рамки

- Атрибуты *border-left-width*, *border-top-width*, *border-right-width* и *border-bottom-width* задают толщину рамки.
- Атрибуты *border-left-color*, *border-top-color*, *border-right-color* и *border-bottom-color* задают цвет рамки.
- Атрибуты *border-left-style*, *border-top-style*, *border-right-style* и *border-bottom-style* задают стиль линий, которыми будет нарисована рамка, или отсутствие рамки. Доступны следующие значения:
 - *none* – рамка отсутствует (поведение по умолчанию);
 - *dotted* – пунктирная линия;
 - *dashed* – штриховая линия;
 - *solid* – сплошная линия;
 - *double* – двойная линия;
 - *groove* – "вдавленная" трехмерная линия;
 - *ridge* – "выпуклая" трехмерная линия;
 - *inset* – трехмерная "выпуклость";
 - *outset* – трехмерное "углубление".



Кафедра
АГyMM

Начало

Содержание



Страница 82 из 187

Назад

На весь экран

Закрыть

РАЗДЕЛ 4

ЯЗЫК JAVASCRIPT

Web-сценарий (или просто *сценарий*) – это программа, помещенная прямо в HTML-код Web-страницы и выполняющая какие-либо действия над ней. Добавить Web-сценарий в HTML-документ можно с помощью элемента `<SCRIPT>` двумя способами:

- определить встроенный сценарий, который располагается непосредственно между парой тегов `<SCRIPT>` и `</SCRIPT>` (встроенный сценарий);
- подключить внешний файл с JavaScript-кодом (внешний сценарий).

В первом случае сценарий располагается непосредственно между парой тегов `<SCRIPT>` и `</SCRIPT>`:

```
<html>
  <head>
    <title>Название документа</title>
    <SCRIPT>
      // JavaScript-код
    </SCRIPT>
  </head>
</body>
```



Кафедра
АГУММ

Начало

Содержание



Страница 83 из 187

Назад

На весь экран

Закреть

```
// ...  
</body>  
</html>
```

или

```
<html>  
  <head>  
    <title>Название документа</title>  
  </head>  
<body>  
  <SCRIPT>  
    // JavaScript-код  
  </SCRIPT>  
</body>  
</html>
```

Во втором случае web-сценарий располагается во внешнем файле – *файле сценариев* (текстовый файл с расширением *.js*). Подключение внешнего файла выполняется с помощью атрибута *src* тега `<SCRIPT>` следующим образом:

```
<SCRIPT SRC="адрес файла сценария"></SCRIPT>
```

Внутри файла сценариев web-сценарий не нужно располагать между тегами `<SCRIPT>` и `</SCRIPT>`.



Кафедра
АГуММ

Начало

Содержание



Страница 84 из 187

Назад

На весь экран

Закреть

4.1 Типы данных

В JavaScript существует несколько основных типов данных.

Числовые данные

Числовые данные (или *числа*) – это обычные числа, над которыми можно производить все арифметические действия, извлекать из них квадратный корень и вычислять тригонометрические функции. Числа могут быть как целыми, так и дробными (целая и дробная части разделяются *точкой*).

```
var n = 123;  
n = 12.345;
```

Существуют специальные числовые значения *Infinity* (*бесконечность*) и *NaN* (*ошибка вычислений*). Эти значения формально принадлежат типу «число», хотя числами в их обычном понимании не являются.

Строковые данные

Строковые данные (или *строки*) – это последовательности букв, цифр, пробелов, знаков препинания и прочих символов, заключенные в одинарные или двойные кавычки.

```
var str = "Мама мыла раму";
```



Кафедра
АГММ

Начало

Содержание



Страница 85 из 187

Назад

На весь экран

Заккрыть

str = 'Одинарные кавычки тоже подойдут';

В JavaScript одинарные и двойные кавычки равноправны. Можно использовать или те или другие.

Логический тип

Логическая величина может принимать только два значения: *true* и *false* – "истина" и "ложь".

Специальный тип null

Данный тип может принимать единственное значение *null* и применяется в особых случаях. Это просто специальное значение, которое имеет смысл «ничего» или «значение неизвестно».

Специальный тип undefined

Принимает единственное значение *undefined* и указывает на то, что переменной не было присвоено никакое значение.

Рассмотренные выше типы называют *примитивными*.

Объекты

Объекты - это сложный тип данных, включающий в себя множество переменных – *свойств* – и набор функций для манипулирования значениями этих переменных – *методов*.



Кафедра
АГУММ

Начало

Содержание



Страница 86 из 187

Назад

На весь экран

Закреть

Переменные

Каждая переменная должна иметь имя, которое однозначно ее идентифицирует. В имени переменной могут присутствовать только латинские буквы, цифры и символы подчеркивания (`_`), причем первый символ должен быть либо буквой, либо символом подчеркивания. При именовании переменных следует помнить, что язык JavaScript чувствителен к регистру.

Перед использованием переменной в коде сценария рекомендуется выполнить ее объявление. Для этого используется ключевое слово `var`.

```
var x;  
var y=1234;  
var z,w;
```

Значение переменной можно присвоить прямо при ее объявлении.

4.2 Операторы

4.2.1 Арифметические операторы

Арифметические операторы служат для выполнения арифметических операций:



Кафедра
АГуММ

Начало

Содержание



Страница 87 из 187

Назад

На весь экран

Заккрыть

Оператор	Описание
+	сложение
-	вычитание
*	умножение
/	деление
%	взятие остатка от деления
++	инкремент (увеличение на единицу)
--	декремент (уменьшение на единицу)

4.2.2 Оператор присваивания

Присваивание производится с помощью знака =.

$$a=2; b=c=3; a=a+b;$$

4.2.3 Операторы сравнения

Операторы сравнения сравнивают два операнда и возвращают логическое значение. Если условие сравнения выполняется, возвращается значение *true*, если не выполняется – *false*. Операторы сравнения приведены в следующей таблице:



Кафедра
АГyMM

Начало

Содержание



Страница 88 из 187

Назад

На весь экран

Закреть

Оператор	Описание
<	Меньше
>	Больше
==	Равно
<=	Меньше или равно
>=	Больше или равно
!=	Не равно
===	Строго равно
!==	Строго не равно

4.2.4 Логические операторы

Логические операторы выполняют действия над логическими значениями.

- НЕ (логическая инверсия): `!`.
- И (логическое умножение): `&&`.
- ИЛИ (логическое сложение): `//`.

Результатом выполнения логических операторов является логическое значение: *true* или *false*.

Основная область применения логических операторов – выражения сравнения. Приведем пример такого выражения:

$$a=(b>0) \ \&\& \ (c+1!=d);$$



Кафедра
АГММ

Начало

Содержание



Страница 89 из 187

Назад

На весь экран

Закреть

4.2.5 Оператор получения типа typeof

Оператор получения типа *typeof* возвращает строку описывающую тип данных операнда. Все значения, которые он может вернуть:

- "string";
- "number";
- "boolean";
- "object";
- "function";
- "null";
- "undefined".

4.2.6 Условный оператор "if"

Условный оператор позволяет нам выполнить одно из двух входящих в него выражений в зависимости от выполнения или невыполнения какого-либо условия.

Формат условного оператора if:

```
if (<условие>
  <блок "то" >
```



Кафедра
АГуММ

Начало

Содержание



Страница 90 из 187

Назад

На весь экран

Закреть

else

<блок "иначе" >

Если *условие* имеет значение *true*, то выполняется <блок "то" >. Если же *условие* имеет значение *false*, то выполняется <блок "иначе" >.

Заметим, то <блок "иначе" >, может отсутствовать:

if (<условие>)

<блок "то" >

Рассмотрим пример:

```
if (x==1) {  
    a="Единица";  
    b=1;  
}  
else {  
    a="Не единица";  
    b=2; }
```

4.2.7 Условный оператор "?"

Условным оператором *?* можно воспользоваться, если условное выражение совсем простое. Формат условного оператора *?*:

<условие> ? <выражение "то" > : <выражение "иначе" >;



Кафедра
АГyMM

Начало

Содержание



Страница 91 из 187

Назад

На весь экран

Заккрыть

Пример:

```
a = (x==1) ? "Единица": "Не единица";
```

4.2.8 Оператор выбора "switch"

Оператор выбора *"switch"* – это фактически несколько условных операторов, объединенных в одном. Его формат:

```
switch ( <исходное выражение> ) {  
    case <значение 1> :  
        <блок 1> ;  
    case <значение 2> :  
        <блок 2> ;  
    <... другие секции case>  
    default :  
        <блок, исполняемый для остальных значений>  
}
```

Результат вычисления *исходного выражения* последовательно сравнивается со *значением 1*, *значением 2* и т.д. и, если такое сравнение увенчалось успехом, выполняется соответствующий блок кода (*блок 1*, *блок 2* и т.д.). Если же ни одно сравнение не увенчалось успехом, выполняется блок кода, находящийся в секции *default* (если она присутствует).

Рассмотрим пример:



Кафедра
АГyMM

Начало

Содержание

◀ ▶

◀◀ ▶▶

Страница 92 из 187

Назад

На весь экран

Закрыть

```
switch (a) {  
    case 1 :  
        b="Единица";  
    case 2 :  
        b="Двойка" ;  
    case 3 :  
        b="Тройка" ;  
    default :  
        b="Другое число" ;
```

4.2.9 Цикл со счетчиком

Цикл со счетчиком используется, если какой-то код нужно выполнить строго определенное число раз. Цикл со счетчиком записывается так:

```
for (<выражение инициализации> ; <условие> ; <приращение>)  
    <тело цикла>
```

Выражение инициализации выполняется самым первым и всего один раз. Оно присваивает переменной, называемой счетчиком цикла, некое начальное значение. Следующий шаг проверка условия. Если условие возвращает *true*, выполняется тело цикла, в противном случае цикл завер-



HTML

Кафедра
АГММ

Начало

Содержание



Страница 93 из 187

Назад

На весь экран

Закрыть

шается. После прохода *тела цикла* выполняется выражение *приращения*, изменяющее значение счетчика. Далее снова проверяется *условие*. Если условие возвращает *true*, выполняется *тело цикла*, в противном случае цикл завершается и т.д.

Пример цикла со счетчиком:

```
for (i=1; i<11; i++) {  
    a+=3;  
    b=i*2+1;  
}
```

4.2.10 Цикл с предусловием

Формат цикла с предусловием:

```
while <условие>  
    <тело цикла>
```

Сначала проверяется *<условие>*. Если оно имеет значение *true*, то выполняется *<тело цикла>*. Если условие ложно изначально, цикл не выполняется ни разу.

Пример цикла с предусловием:

```
while (a<100) {  
    a=a*i+2;
```



Кафедра
АГуММ

Начало

Содержание



Страница 94 из 187

Назад

На весь экран

Закреть

```
i=++i;  
}
```

4.2.11 Цикл с постусловием

В *цикле с постусловием* сначала выполняется тело цикла, а затем проверяется условие. Формат цикла с постусловием:

```
do  
    <тело цикла>  
while (<условие>);
```

Цикл с постусловием выполняется хотя бы один раз, даже если его условие изначально ложно.

Пример цикла с постусловием:

```
do {  
    a=a*i+2;  
    i=++i;  
} while (a<100);
```

4.2.12 Операторы прерывания и перезапуска цикла

Иногда нужно прервать волнение цикла. Для этого используется оператор *break*.



Кафедра
АГММ

Начало

Содержание



Страница 95 из 187

Назад

На весь экран

Заккрыть

Оператор перезапуска *continue* позволяет перезапустить цикл, т.е. оставить невыполненными все последующие выражения, входящие в тело цикла и запустить выполнение цикла с самого его начала: проверка условия, выполнение приращения и тела.

4.3 Функции

Функция – это особым образом написанный и оформленный фрагмент кода JavaScript, который можно вызвать из любого сценария на данной странице. В JavaScript функции можно разделить на "написанные" пользователем и встроенные.

4.3.1 Объявление и вызов функций

Прежде чем функция будет использована где-то в сценарии, ее нужно объявить. Объявление функции производится с помощью ключевого слова *function*

```
function <имя функции> (/<список параметров>/  
    <тело функции>
```

Имя функции должно быть уникальным в пределах страницы. Для имен функций действуют те же правила, что и для имен переменных.

Список параметров представляет собой набор переменных, в которых при вызове функции будут помещены значения переданных ей парамет-



Кафедра
АГММ

Начало

Содержание



Страница 96 из 187

Назад

На весь экран

Закреть

ров (эти параметры называются *формальными*).

В пределах *тела функции* над принятыми ее параметрами (если они есть) и другими данными выполняются некоторые действия и вырабатывается результат. Чтобы вернуть результат из функции в выражение, из которого она была вызвана, используется *оператор возврата return*:

```
return переменная или выражение>;
```

Здесь переменная должна соержжать возвращаемое значение, а выражение должно его вычислять.

Пример объявления функции:

```
function divine(a,b) {  
    var c;  
    c=a/b;  
    return c;  
}
```

После объявления функции ее можно вызвать из любого сценария, присутствующего на этой же странице. Для этого используется следующий формат:

```
<имя функции> ([<список фактических параметров, разделенных запятыми>])
```

Здесь указывается имя нужной функции и в круглых скобках перечисляются фактические параметры, над которыми нужно выполнить



Кафедра
АГуММ

Начало

Содержание



Страница 97 из 187

Назад

На весь экран

Заккрыть

соответствующие действия.

4.3.2 Встроенные функции JavaScript

Язык JavaScript предоставляет целый набор встроенных функций. Эти функции объявлены и реализованы в самом языке. Рассмотрим некоторые из них, поддерживаемые всеми WEB-обозревателями:

parseInt(<Строка>[, <основание системы счисления>]) – преобразует строку *<строка>*, содержащее число, в числовое значение. Вторым параметром указывается число, являющееся основанием системы счисления, в которой представлено число в строке. Если этот параметр не указан, то предполагаем, что число представлено в десятичной системой счисления.

Функция *parseInt* успешно преобразует в числовой тип строку, содержащую только число или число, за которым следуют символы, не являющиеся цифрами. В противном случае функция *parseInt* возвращает значение NaN.

<i>Пример использования функции parseInt</i>	<i>Результат</i>
<i>parseInt("71");</i>	<i>71</i>
<i>parseInt("371a6");</i>	<i>371</i>
<i>parseInt("cdc");</i>	<i>NaN</i>



Кафедра
АГММ

Начало

Содержание



Страница 98 из 187

Назад

На весь экран

Заккрыть

parseFloat(*<строка>*) – преобразует строку в число с плавающей запятой. Данная функция ведет себя аналогично *parseInt*.

<i>Пример использования функции parseFloat</i>	<i>Результат</i>
<i>parseFloat("7.2");</i>	<i>7,2</i>
<i>parseFloat("1.7a");</i>	<i>1,7</i>
<i>parseFloat("Float");</i>	<i>NaN</i>

isNaN(*<выражение>*) – возвращает *false*, если *<выражение>* представляет собой правильное число, и *true* – в противном случае. Как правило, в качестве параметра *<выражение>* указывается математическое выражение, корректность результатов вычисления которых надо проверить.

<i>Пример использования функции isNaN</i>	<i>Результат</i>
<i>x=7; y=0;</i> <i>a = isNaN(x*y)</i> <i>b = isNaN(x/y)</i> <i>isNaN(parseFloat("Float"));</i>	<i>в переменную a запишется значение false</i> <i>в переменную b запишется значение true</i> <i>возвращает true</i>

eval(*<выражение JavaScript в виде строки>*) - возвращает результат выполнения выражения JavaScript, записанного в виде строки.



Кафедра
АГyMM

Начало

Содержание



Страница 99 из 187

Назад

На весь экран

Закрыть

Пример использования f(x) eval	Результаты
a = eval("2*3")	a = 6
b = eval("2+3")	b = 5

4.3.3 Рекурсия

В JavaScript функции могут вызывать и сами себя. Такой прием программирования иногда может быть очень полезен и называется *рекурсией*.

В обычных условиях, если функция вызовет сама себя, то она войдет в бесконечный цикл вызовов, из которого нет выхода (так называемая *бесконечная рекурсия*). Поэтому функция, предназначенная для рекурсивного вызова, должна предусматривать возможность выхода из этого цикла вызовов. Стандартного способа сделать это не существует, поэтому решение для каждого конкретного случая нужно искать особо.

Рассмотрим пример функции, правильно использующей рекурсию:

```
function factorial(a)    if (a == 0)    return 1;    else    return
(a * factorial(a - 1));
```

Эта функция вычисляет факториал числа a. Она вызывает сама себя для того, чтобы получить факториал числа a - 1. Заметим также, что в теле функции выполняется проверка условия равенства a нулю; если это условие истинно, возвращается единица, дальнейших рекурсивных вызовов не происходит.



Кафедра
АГММ

Начало

Содержание



Страница 100 из 187

Назад

На весь экран

Закрыть

4.4 Объекты

Объект – это сложный тип данных, включающий в себя множество переменных(свойств) и набор функций для манипуляции значениями этих переменных(методы).

Каждый объект должен иметь уникальное имя, по которому к нему можно обратиться. К именам объектов предъявляются те же требования, что и к именам переменных.

Заметим, что объект описывает только набор доступных в нем свойств и методов. Его нельзя использовать напрямую. Для этого необходимо с помощью оператора **new** создать экземпляр объекта:

```
new <имя объекта>([список параметров]);
```

Список параметров может отсутствовать.

```
var Red;  
Red = new Color();
```

В данном примере создается новый экземпляр Red некоего объекта Color.

После создания экземпляра объекта можно обращаться к его свойствам и методам, используя так называемый синтаксис "с точной":

```
<имя экземпляра объекта>.<имя свойства>  
<имя экземпляра объекта>.<имя свойства>([список параметров])
```



Кафедра
АГММ

Начало

Содержание



Страница 101 из 187

Назад

На весь экран

Закрыть

Если метод не принимает параметров, то указываются пустые скобки.

```
Red.prop = 1;  
Red.metod();
```

Объект может содержать внутри себя другие объекты (внутренние объекты). Для обращения к свойствам и методам внутренних объектов используется практически тот же синтаксис:

```
onterObj.InnerObj.prop = 1;
```

где *onterObj* – экземпляр внешнего объекта, *InnerObj* – экземпляр внутреннего объекта.

Экземпляры объектов удалять не нужно – их удалит JavaScript, как только страница со сценариями, создавшими экземпляры объектов, будет выгружена WEB-обозревателем из памяти.

Все объекты, доступные в JavaScript, делятся на три вида:

- предоставляемые языком JavaScript;
- предоставляемые web-обозревателем;
- созданные программистом.

4.4.1 Объект строки String

Объект *String* представляет обычную строку, то есть константу или переменную строкового типа. Например:



Кафедра
АГyMM

Начало

Содержание



Страница 102 из 187

Назад

На весь экран

Закреть

```
var str;  
str = "Интернет-технологии";
```

Экземпляр этого объекта также может быть создан с помощью оператора *new*:

```
new String([<строковая константа>]);
```

Например:

```
var str;  
str = new String("Интернет-технологии");
```

Объект `String` поддерживает единственное свойство *length*, возвращающее длину строки в символах:

```
l1 = str.length;  
l2 = "Интернет-технологии".length
```

Рассмотрим основные методы объекта `String`.

- Метод *charAt(<номер символа>)* возвращает символ строки, номер которого передан в качестве параметра.
- Метод *charCodeAt(<номер символа>)* возвращает код в кодировке Unicode символа строки, номер которого передан в качестве параметра.



Кафедра
АГyMM

Начало

Содержание



Страница 103 из 187

Назад

На весь экран

Закреть



- Метод *concat*(*<добавляемая строка>*) добавляет к текущей строке ту, что передана в качестве параметра, и возвращает результат.
- Метод *indexOf*(*<искомая строка>*[, *<начальная позиция поиска>*]) возвращает номер позиции в текущей строке, в которой находится строка, переданная в качестве параметра. Первым параметром передается сама искомая строка (подстрока поиска). Вторым, необязательным, параметром может быть передан номер символа текущей строки, с которого начнется поиск подстроки поиска. По умолчанию поиск начинается с первого символа (то есть символа с номером 0). Если подстрока поиска не была найдена, возвращается значение -1 .
- Метод *lastIndexOf* аналогичен методу *indexOf*, но ведет поиск подстроки, начиная с конца текущей строки.

```
n1 = "Интернет-технологии".lastIndexOf("a");
```

- Метод *link*(*<интернет-адрес гиперссылки>*) возвращает HTML-код, создающий гиперссылку. Текущая строка станет текстом этой гиперссылки, а строка, переданная в качестве параметра, — ее интернет-адресом.
- Метод *slice*(*<номер первого символа>*[, *<номер последнего символа>*]) возвращает заданный фрагмент текущей строки. Первым параметром передается номер первого символа возвращаемого фраг-

мента строки. Второй, необязательный, параметр задает номер его последнего символа. Если он опущен, возвращаемый фрагмент будет содержать все оставшиеся символы строки.

- Метод *split*(*<символ-разделитель>*) разбивает текущую строку на фрагменты, формирует массив, элементами которого станут полученные фрагменты строки, и возвращает его.
- Метод *sub* заключает текущую строку в парный тег `<SUB>` (нижний индекс) и возвращает ее. Он не принимает параметров.
- Метод *sup* заключает текущую строку в парный тег `<SUP>` (верхний индекс) и возвращает ее. Он не принимает параметров.
- Метод *toLowerCase* преобразует текущую строку в нижний регистр и возвращает ее. Он не принимает параметров.
- Метод *toUpperCase* преобразует текущую строку в верхний регистр и возвращает ее. Он не принимает параметров.

4.4.2 Объект числа **Number**

Объект *Number* представляет обычное число – константу или переменную числового типа. Его экземпляр может быть создан двумя способами.

Первый способ:



Кафедра
АГУММ

Начало

Содержание



Страница 105 из 187

Назад

На весь экран

Закреть

```
var num;  
num = 12;
```

Второй способ с помощью оператора *new*:

```
new Number(<числовая константа>);
```

Например:

```
var num;  
num = new Number(12);
```

Объект `Number` поддерживает несколько свойств:

- свойство `MAX_VALUE` возвращает максимальное допустимое в JavaScript число;
- свойство `MIN_VALUE` возвращает минимальное допустимое в JavaScript число;
- свойство `NaN` возвращает значение NaN;
- `NEGATIVE_INFINITY` – математическая "минус бесконечность";
- `POSITIVE_INFINITY` – математическая "плюс бесконечность".

Рассмотрим некоторые методы объекта `Number`.



Кафедра
АГУММ

Начало

Содержание



Страница 106 из 187

Назад

На весь экран

Заккрыть

- Метод *toExponential*(*<количество знаков после запятой>*) возвращает экспоненциальное представление числа в строковом виде.
- Метод *toFixed*(*<количество знаков после запятой>*) возвращает обычное представление числа в строковом виде. Единственный параметр этого метода задает количество знаков после запятой. Если заданное количество знаков меньше, чем присутствует в числе, оно будет округлено.
- Метод *toPrecision*(*<общее количество знаков>*) возвращает обычное представление числа в строковом виде с заданным количеством знаков. Единственный параметр задает общее количество знаков числа (разделитель целой и дробной части числа также считается за знак).

4.4.3 Объект логического значения Boolean

Объект *Boolean* представляет логическое значение логического типа. В следующем примере задается экземпляр этого объекта

```
var b; c = (a == b);
```

Экземпляр объекта *Boolean* может быть создан с помощью оператора *new*:

```
new Boolean(<логическая константа>);
```



Кафедра
АГММ

Начало

Содержание



Страница 107 из 187

Назад

На весь экран

Закреть

Например:

```
var b;  
b = new Boolean(a == b);
```

Объект `Boolean` поддерживает два метода: `toString` и `valueOf`. Первый метод возвращает строковое представление текущего значения объекта в виде строки `"true"` или `"false"`, второй — само его значение.

4.4.4 Объект значения даты `Date`

Объект `Date` служит для хранения значений даты и времени (время также включает миллисекунды). Такие значения "внутри" JavaScript представляют собой обычные числа, указывающие количество миллисекунд, прошедших с полуночи 1 января 1970 года.

Экземпляр объекта `Date` создается только с помощью оператора `new`:

```
new Date([<год>, <месяц>, <число>[, <часы>, <минуты>, <секунды>]])
```

Рассмотрим методы объекта `Date`.

- Метод `getDate` возвращает число из текущего значения даты.
- Метод `getDay` возвращает номер месяца недели. Здесь число 0 обозначает воскресенье, число 1 — понедельник, 2 — вторник и т.д.



Кафедра
АГУММ

Начало

Содержание



Страница 108 из 187

Назад

На весь экран

Заккрыть

- Метод *getFullYear* возвращает номер года в виде числа из четырех цифр.
- Метод *getHours* возвращает количество часов.
- Метод *getMilliseconds* возвращает количество миллисекунд.
- Метод *getMinutes* возвращает количество минут.
- Метод *getMonth* возвращает номер месяца. Здесь число 0 обозначает январь, число 1 — февраль и т. д.
- Метод *getSeconds* возвращает количество секунд.
- Метод *getTime* возвращает "числовое" представление значения даты и времени в виде количества миллисекунд, прошедших с полуночи 1 января 1970 года.
- Метод *setDate* изменяет число у значения даты и возвращает новое значение даты и времени.
- Метод *valueOf* возвращает текущее значение даты и времени.
- Метод *toString* возвращает строковое представление даты.

Рассмотренные выше методы не принимают параметров.



Кафедра
АГУММ

Начало

Содержание



Страница 109 из 187

Назад

На весь экран

Заккрыть

4.4.5 Объект массив Array

Массив – это пронумерованный набор переменных (элементов), ведущий себя как одна переменная. Доступ к отдельному элементу массива выполняется по его порядковому номеру, называемому индексом. А общее число элементов массива называется его размером.

Объект *Array* служит для представления массивов. Для создания его экземпляра нужно просто присвоить любой переменной список его элементов, разделенных запятыми и заключенных в квадратные скобки:

```
var arr;  
arr = [1, 2, 3, 4];
```

Экземпляр объекта *Array* может быть создан с помощью оператора *new*:

```
new Array([ <список значений элементов массива, разделенных запятыми>]);
```

Например:

```
var arr;  
arr = new Array(1, 2, 3, 4);
```

Объект *Array* поддерживает единственное свойство *length*, возвращающее размер массива.

Рассмотрим методы объекта *Array*.



Кафедра
АГММ

Начало

Содержание

◀ ▶

◀◀ ▶▶

Страница 110 из 187

Назад

На весь экран

Закреть



Кафедра АГММ

Начало

Содержание



Страница 111 из 187

Назад

На весь экран

Закрыть

- Метод *concat*(*<список добавляемых в новый массив элементов, разделенных запятыми>*) создает новый массив, содержащий элементы текущего массива и значения всех переданных данному методу параметров, и возвращает его. В качестве параметров могут приниматься массивы.
- Метод *join*(*<символ-разделитель>*) возвращает строку, полученную путем объединения значений всех элементов текущего массива; эти значения разделяются заданным символом-разделителем.
- Метод *shift*() удаляет первый элемент текущего массива и возвращает его значение. Он не принимает параметров.
- Метод *pop*() удаляет последний элемент текущего массива и возвращает его значение. Он не принимает параметров.
- Метод *reverse*() меняет порядок следования элементов текущего массива на противоположный и возвращает копию обновленного массива. Он не принимает параметров.
- Метод *slice*(*<индекс первого элемента>*, *<индекс последнего элемента>*) извлекает заданные элементы текущего массива, формирует из них новый массив и возвращает его. Если последний параметр опущен, извлекаются все оставшиеся элементы массива.
- Метод *sort*() сортирует текущий массив и возвращает его копию уже в отсортированном виде.

4.4.6 Объект Math

Объект *Math* представляет тригонометрические и логарифмические функции, а также математические константы.

Поддерживаемые объектом Math свойства:

- свойство *E* возвращает константу e ;
- свойство *LN10* возвращает $\ln 10$;
- свойство *LN2* возвращает $\ln 2$;
- свойство *LOG10E* возвращает $\lg e$;
- свойство *LOG2E* возвращает $\log_2 e$;
- свойство *PI* возвращает константу π ;
- свойство *SQRT1_2* возвращает $\sqrt{\frac{1}{2}}$;
- свойство *SQRT2* возвращает $\sqrt{2}$.

Рассмотрим методы, поддерживаемые объектом Math.

- Метод *abs(<параметр>)* возвращает абсолютное значение параметра.
- Метод *acos(<параметр>)* возвращает арккосинус параметра в радианах.



Кафедра
АГyMM

Начало

Содержание



Страница 112 из 187

Назад

На весь экран

Заккрыть



Кафедра АГyMM

Начало

Содержание



Страница 113 из 187

Назад

На весь экран

Закрyть

- Метод $\text{asin}(\langle \text{параметр} \rangle)$ возвращает арксинус параметра в радианах.
- Метод $\text{atan}(\langle \text{параметр} \rangle)$ возвращает арктангенс параметра в радианах.
- Метод $\text{atan2}(\langle X \rangle, \langle Y \rangle)$ возвращает угол в радианах между горизонтальной осью и прямой, проведенной через начало координат и точку с координатами X, Y .
- Метод $\text{ceil}(\langle \text{параметр} \rangle)$ возвращает ближайшее целое число, большее или равное параметру.
- Метод $\text{cos}(\langle \text{параметр} \rangle)$ возвращает косинус параметра, заданного в радианах.
- Метод $\text{exp}(\langle \text{параметр} \rangle)$ возвращает значение $e^{\text{параметр}}$.
- Метод $\text{floor}(\langle \text{параметр} \rangle)$ возвращает ближайшее целое число, меньшее параметра или равное ему.
- Метод $\text{log}(\langle \text{параметр} \rangle)$ возвращает натуральный логарифм параметра.
- Метод $\text{max}(\langle \text{параметр } 1 \rangle, \langle \text{параметр } 2 \rangle)$ возвращает наибольший из параметров.
- Метод $\text{min}(\langle \text{параметр } 1 \rangle, \langle \text{параметр } 2 \rangle)$ возвращает наименьший из параметров.



- Метод $\text{pow}(\langle \text{основание} \rangle, \langle \text{порядок} \rangle)$ возвращает значение $\text{основание}^{\text{порядок}}$.
- Метод $\text{random}()$ возвращает псевдослучайное число от 0 включительно до 1 исключительно.
- Метод $\text{round}(\langle \text{параметр} \rangle)$ возвращает значение параметра, округленное до ближайшего целого.
- Метод $\text{sin}(\langle \text{параметр} \rangle)$ возвращает синус параметра, заданного в радианах.
- Метод $\text{sqrt}(\langle \text{параметр} \rangle)$ возвращает квадратный корень из параметра.
- Метод $\text{tan}(\langle \text{параметр} \rangle)$ возвращает тангенс параметра, заданного в радианах.

4.5 Общие принципы написания Web-сценариев

4.5.1 Внутреннее представление страницы (Document Object Model)

Загружая страницу, WEB-обозреватель создает экземпляр объекта HTMLDocument. Данный экземпляр предоставляет загружаемую страницу и документ под именем document. Далее по мере загрузки страницы web-обозреватель считывает код HTML и создает внутренние пред-

ставление экземпляров страницы в виде экземпляров соответствующих объектов. Так для гиперссылки создается экземпляр объекта HTMLLinkElement, для изображения – экземпляр объекта HTMLImageElement, для абзаца – экземпляр объекта HTMLParagraphElement, для формы – экземпляр объекта HTMLFormElement и др. Кроме того, текстовое содержание каждого тега представляется как экземпляр объекта Text.

Все объекты, представляющие элементы страницы, являются потомками объекта HTMLElement. Поэтому этот объект представляет свойства и методы, общее для всей элементов страницы.

Однако, в отличие от экземпляра объекта HTMLDocument, который представляет саму веб-страницу, экземпляры объектов, представляющих отдельные элементы этой страницы, изначально недоступны в сценариях. Чтобы получить в сценарии доступ к определенному элементу, нужно дать ему имя. Для этого достаточно вставляющий элемент тег необязательный атрибут *ID*. В качестве значения этого атрибута указывается нужное имя:

`<p ID="p1"> Абзац с именем p1 </p>`

Имена элементов страницы должны удовлетворять следующим трем правилам:

- имена должны содержать только латинские буквы, цифры и знаки подчеркивания (`_`);



Кафедра
АГММ

Начало

Содержание



Страница 115 из 187

Назад

На весь экран

Закрыть

- первым символом имени должна быть буква или знак подчеркивания;
- имена должны быть уникальными в пределах данной web-страницы.

Для доступа к элементу страницы используется такой же синтаксис, как и для доступа к обычному экземпляру объекта JavaScript:

<имя элемента>.<имя свойства или метода>[список фактических параметров].

Существует и другой способ обращения к элементам страницы – через коллекции. Коллекция JavaScript – это ассоциативный массив, представляющий список экземпляров каких-либо объектов и сам являющийся особым объектом.

Рассмотрим некоторые коллекции, которые поддерживает объект HTMLDocument:

- *all* – все элементы страницы (включая теги <HTML>, <HEAD>, <TITLE> и <BODY>);
- *anchors* – все якоря страницы;
- *forms* – все web-формы;
- *images* – все графические изображения;
- *links* – все гиперссылки (включая горячие области);
- *styleSheets* – все таблицы стилей.



Кафедра
АГММ

Начало

Содержание



Страница 116 из 187

Назад

На весь экран

Закреть

Перечисленные коллекции, за исключением `styleSheets`, являются экземплярами объекта `HTMLCollection`. Все они поддерживают свойство `length`, которое возвращает количество элементов в коллекции.

Получить доступ к каждому элементу страницы можно указав строковый индекс (совпадает с именем элемента страницы):

```
parObj=document.all["n1"];
```

или по числовому индексу (нумерация коллекции начинается с нуля):

```
var i, linkObj;  
for (i=0; i<document.links.length; i++) {  
linkObj=document.links[i]; }
```

В данном примере просматриваем коллекции `links`, чтобы "обойти" все гиперссылки на странице.

Объектной моделью документа (Document Object Model или DOM) называется совокупность экземпляров различных объектов, представляющих саму страницу и все ее элементы.

4.5.2 Работа с окнами Web-обозревателя

Текущее окно Web-обозревателя представляется как экземпляр объекта `Window`, созданный самим Web-обозревателем и хранящийся в переменной `window`. Этот объект поддерживает очень много свойств, мето-



Кафедра
АГММ

Начало

Содержание



Страница 117 из 187

Назад

На весь экран

Заккрыть

дов и событий, позволяющих выполнять над окнами Web-обозревателя различные манипуляции. Рассмотрим некоторые из них.

Свойство *name* устанавливает/возвращает имя окна.

Свойство *frames* возвращает список фреймов текущего окна. Перечисляются только прямые фреймы, без их подфреймов. Список фреймов не является Array, но похож на массив в том смысле, что у него есть свойство *length*, и его элементы доступны по индексу [i].

Свойство *status* позволяет устанавливать и получать статусную строку окна.

Метод *open* создает новое окно браузера.

Метод *close* закрывает окно браузера.

В JavaScript имеется объект, который отвечает за документ в целом. Этот объект является свойством объекта Window. Этот объект называется *document*. Через объект Document в JavaScript мы можем получить доступ к любому элементу нашей страницы.

Объект *document* обладает свойством *title*, которое позволяет изменить название страницы.

Метод *write* (*<вставляемая строка>*) позволяет изменить содержимое страницы. Метод *write* в качестве единственного параметра принимает строку с HTML-кодом. Web-обозреватель обработает этот HTML-код и вставит результат его обработки в то место кода страницы, где встретился вызов данного метода.



Кафедра
АГУММ

Начало

Содержание



Страница 118 из 187

Назад

На весь экран

Заккрыть

Приведем пример:

```
write(«p>Это новый абзац.</p>");
```

4.5.3 События элементов страницы и их обработка

События мыши

Событие *onClick* возникает при щелчке мышью на элементе страницы.

Событие *onDbClick* возникает при двойном щелчке мышью на элементе страницы.

Событие *onMouseMove* возникает при перемещении курсора мыши над элементом страницы.

Событие *onMouseOut* возникает, когда курсор мыши уходит за границы элемента страницы.

Событие *onMouseMove* возникает при перемещении курсора мыши над элементом страницы.

События клавиатуры

Событие *onKeyDown* возникает, когда посетитель нажимает любую клавишу клавиатуры. Если клавиша удерживается нажатой, возникает последовательно.



Кафедра
АГУММ

Начало

Содержание



Страница 119 из 187

Назад

На весь экран

Закрыть

Событие *onKeyPress* возникает, когда посетитель нажимает алфавитно-цифровую клавишу. Если клавиша удерживается нажатой, возникает последовательно.

Событие *onKeyUp* возникает, когда посетитель отпускает нажатую ранее клавишу.

4.5.4 Работа с web-формами и элементами управления

Для доступа к формам страницы можно использовать коллекцию WEB.

Форму представляет объект `HTMLFormElement`, созданный на основе объекта `HTMLElement` и наследующий его свойства, методы и события. Кроме того этот объект представляет свои методы и события.

Метод *submit* выполняет отправку данных серверной программе или другой странице. Данный метод аналогичен нажатию кнопки отправки данных.

Метод *reset* аналогичен нажатию кнопки сброса формы. Он очищает форму, подставляя во все ее элементы управления изначальные значения.

Объект `HTMLFormElement` поддерживает события *onReset* и *onSubmit*. Событие `onReset` возникает при очистке формы в результате нажатия кнопки сброса и вызова метода `reset` объекта `HTMLFormElement`. Событие `onSubmit` возникает при отправке данных в результате вызова



Кафедра
АГуММ

Начало

Содержание



Страница 120 из 187

Назад

На весь экран

Закреть

метода `submit`.

Сценарии реализующие обработку введенных данных, работают непосредственно с элементами управления.

Свойства, методы и события элементов управления условно можно разделить на две группы:

- поддерживаемые всеми элементами управления;
- специфичные для определенных элементов управления.

Рассмотрим некоторые общие свойства, методы и события поддерживаемые всеми элементами управления.

Свойство *disabled*. Если установлено значение `true`, то элемент управления недоступен. Если установлено значение `false`, то элемент управления доступен для ввода. Данное свойство не доступно для скрытого поля, надписи и группы.

Свойство *type* задает или возвращает тип элемента управления, задаваемый атрибутом `TYPE`.

Методов, общих для всех элементов управления, существует всего два. Метод *focus* активизирует текущий элемент управления, а метод *blur* делает его неактивным (при этом активизирует следующий элемент управления).

События общие для всех элементов управления:



Кафедра
АГyMM

Начало

Содержание



Страница 121 из 187

Назад

На весь экран

Закрыть

- *onBlur* возникает, когда текущий элемент управления перестает быть активным;
- *onFocus* возникает когда текущий элемент управления становится активным.

Теперь рассмотрим свойства, методы и события, специфичные для определенных элементов управления.

Поле ввода

Данный объект поддерживает свойство *value*, которое задает или возвращает текущее значение, введенное в поле ввода.

Свойство *defaultValue* задает или возвращает изначальное значение, которое должно присутствовать в поле ввода при загрузке страницы или после выполнения сброса формы. Свойства *maxLength* и *size* соответствуют одноименным атрибутам тега `<input>`.

Метод *select* выделяет все содержимое поля ввода. Он не принимает параметров и не возвращает значения.

Событие *onChange* возникает при любом изменении значения в поле ввода посетителем.

Событие *onselect* возникает при выделении посетителем значения в поле ввода.

Область редактирования



Кафедра
АГyMM

Начало

Содержание



Страница 122 из 187

Назад

На весь экран

Закреть

Прежде всего, этот объект поддерживает свойства *defaultValue* и *value*, уже знакомые нам по полю ввода. Свойство *type* для области редактирования всегда возвращает строку *"textarea"*.

Свойство *cols* соответствует одноименному атрибуту тега `<textarea>` и задает или возвращает ширину области редактирования в символах в виде числа. Свойство *rows* также соответствует одноименному атрибуту тега `<textarea>`; оно задает или возвращает высоту области редактирования в строках в виде числа.

Метод *select* также знаком нам по полю ввода. Он выделяет все содержимое области редактирования.

Данный элемент управления поддерживает два события: *onChange* и *onSelect*. Они знакомы нам по полю ввода.

Поле ввода пароля

Этот объект поддерживает свойства *defaultValue*, *maxLength*, *size* и *value*, метод *select*, знакомые нам по обычному полю ввода.

Кнопка отправки данных

Данный элемент управления поддерживает свойство *value*, задающее или возвращающее надпись на кнопке в виде строки.

Метод *click* позволяет имитировать щелчок на кнопке. Он не принимает параметров и не возвращает значения.



Кафедра
АГММ

Начало

Содержание



Страница 123 из 187

Назад

На весь экран

Закреть

Если нужно отследить щелчок на ней, обрабатывают либо событие *onClick* самой кнопки, либо событие *onSubmit* формы, в которой находится эта кнопка.

Кнопка сброса формы

Данный элемент управления поддерживает те же самые свойства, методы и события, что и рассмотренная ранее кнопка отправки данных. Для отслеживания щелчка на этой кнопке обрабатывают либо ее событие *onClick*, либо событие *onReset* формы, в которой находится эта кнопка.

Кнопка

Данный элемент управления поддерживает те же самые свойства, методы и события, что и рассмотренные ранее кнопки отправки данных и сброса формы. Для отслеживания щелчка на ней обрабатывают событие *onClick* этой кнопки.

Флажок

Флажок поддерживает свойство *value*, задающее или возвращающее значение флажка, и метод *click*, имитирующий щелчок на флажке.

Свойство *checked* задает или возвращает текущее состояние флажка, то есть включен он или выключен. Значение *true* обозначает, что флажок включен, значение *false* — что флажок выключен.



Кафедра
АГУММ

Начало

Содержание



Страница 124 из 187

Назад

На весь экран

Заккрыть

Свойство *defaultChecked* задает или возвращает изначальное состояние флажка, заданное в его HTML-коде атрибутом `checked`. Здесь значение `true` также обозначает, что флажок изначально включен, значение `false` — что флажок изначально выключен.

Для отслеживания щелчка на нем обрабатывается событие *onClick*.

Переключатель

Переключатель поддерживает свойства *defaultChecked*, *value* и метод *click*, знакомые нам по флажку.

Для отслеживания щелчка на нем обрабатывается "универсальное" событие *onClick*.

Список

Данный элемент управления поддерживает свойство *type*. В случае списка с возможностью выбора только одного пункта оно всегда возвращает строку *"select-one"*. Если же список позволяет выбрать сразу несколько пунктов, оно всегда возвращает строку *"select-multiple"*.

Свойство *size* соответствует одноименному атрибуту тега `<select>` и задает или возвращает размер списка в пунктах (позициях) в виде числа.

Свойство *multiple* соответствует одноименному атрибуту тега `<select>` и указывает, поддерживает ли список возможность выбора сразу нескольких пунктов. Значение `true` указывает, что в списке можно выбрать несколько пунктов одновременно, а значение `false` — только один пункт.



Кафедра
АГyMM

Начало

Содержание



Страница 125 из 187

Назад

На весь экран

Закрыть

Свойство *selectedIndex* задает или возвращает номер выбранного в списке пункта в виде числа. При этом:

- если список позволяет выбирать одновременно только один пункт, возвращается номер именно этого пункта;
- если список позволяет выбирать сразу несколько пунктов, возвращается номер первого выбранного пункта;
- если ни один пункт в списке не выбран, возвращается значение -1 .

Свойство *selected* задает или возвращает состояние пункта — выбран ли он в данный момент или нет. Значение *true* обозначает, что пункт выбран, значение *false* — не выбран.

Свойство *defaultSelected* задает или возвращает изначальное состояние пункта списка, заданное в HTML-коде атрибутом *selected* тега `<option>`. Значение *true* обозначает, что пункт изначально выбран, значение *false* — изначально не выбран.

Свойство *value* соответствует одноименному атрибуту тега `<OPTION>` и задает или возвращает значение пункта списка в виде строки.

Свойство *text* задает или возвращает текст пункта в виде строки.

Свойство *index* возвращает номер пункта в списке в виде числа.

Метод *add* добавляет пункт в список.

Метод *remove* удаляет заданный пункт из списка.



Кафедра
АГУММ

Начало

Содержание



Страница 126 из 187

Назад

На весь экран

Закрыть

4.5.5 Управление графикой и мультимедийными элементами

Каждое графическое изображение, помещенное на страницу, представляется Web-обозревателем как экземпляр объекта *HTMLImageElement*, производного от объекта *HTMLElement*.

Рассмотрим четыре специфических свойства, поддерживаемых объектом *HTMLImageElement*.

Свойство *src* задает или возвращает интернет-адрес файла, где хранится графическое изображение. С его помощью мы можем сменить изображение, отображаемое на странице, другим.

Свойство *alt* задает или возвращает текст замены, заданный для текущего изображения.

Свойство *lowsrc* задает или возвращает интернет-адрес файла, где хранится графическое изображение пониженного качества.

Если нам нужно быстро узнать, загружено ли изображение, мы можем воспользоваться свойством *complete*. Это свойство возвращает *true*, если изображение загружено и выведено на страницу, и *false* в противном случае.

Специфических событий объект *HTMLImageElement* поддерживает три:

- *onAbort* - возникает при прерывании загрузки изображения;
- *onError* - Возникает при неудачной загрузке изображения;



Кафедра
АГММ

Начало

Содержание



Страница 127 из 187

Назад

На весь экран

Закрыть

- *onLoad* - Возникает после завершения загрузки изображения.

Кроме того, объект `HTMLImageElement` поддерживает все события объекта `HTMLElement`, поскольку является его "наследником". Как правило, именно события объекта `HTMLElement` и обрабатываются.



*Кафедра
АГyMM*

Начало

Содержание



Страница 128 из 187

Назад

На весь экран

Закрyть

РАЗДЕЛ 5

СОВРЕМЕННЫЕ СРЕДСТВА СОЗДАНИЯ WEB-СТРАНИЦ

5.1 Скриптовые языки программирования

HTML. Язык разметки гипертекста (Hypertext Markup Language), или, как его чаще называют, HTML, – это компьютерный язык, лежащий в основе World Wide Web (Всемирной Паутины). Благодаря языку HTML любой текст можно разметить, преобразовав его в гипертекст с последующей публикацией в Web. Язык HTML имеет собственный набор символов, с помощью которых Web-браузеры отображают страницу. Одной из отличительных особенностей HTML-документов является то, что сам документ содержит только текст, а все остальные объекты встраиваются в документ в момент его отображения браузером с помощью специальных тэгов и хранятся отдельно. При сохранении HTML-файла в месте размещения документа создается папка, в которую помещаются сопутствующие ему графические элементы оформления.

PHP. В первую очередь PHP используется для создания скриптов, работающих на стороне сервера, для этого его, собственно, и придумали. PHP способен обрабатывать данные html-форм, динамически генерировать html страницы и т.п. Но есть и другие области, где может использоваться PHP. Вторая область – это создание скриптов, выполняющихся



Кафедра
АГуММ

Начало

Содержание



Страница 129 из 187

Назад

На весь экран

Закрыть

в командной строке. То есть с помощью PHP можно создавать такие скрипты, которые будут исполняться, вне зависимости от web-сервера и браузера, на конкретной машине.

Ajax. Ажак расшифровывается как Asynchronous Javascript And XML (Асинхронные Javascript и XML) и технологией в строгом смысле слова не является. Если в стандартном web-приложении обработкой всей информации занимается сервер, тогда как браузер отвечает только за взаимодействие с пользователем, передачу запросов и вывод поступившего HTML, то в Ajax-приложении между пользователем и сервером появляется еще один посредник – движок Ajax. Он определяет, какие запросы можно обработать "на месте", а за какими необходимо обращаться на сервер.

JavaScript. Язык программирования JavaScript разработан фирмой Netscape для создания интерактивных HTML-документов. Это объектно-ориентированный язык разработки встраиваемых приложений, выполняющихся как на стороне клиента, так и на стороне сервера. Синтаксис языка очень похож на синтаксис Java – поэтому его называют – Java-подобным.

VBScript. Язык создания сценариев VBScript разработан фирмой Microsoft, является подмножеством достаточно распространенного в среде программистов языка Visual Basic разработки прикладных программ Windows-приложений. Как и его родитель, язык VBScript достаточно прост



Кафедра
АГУММ

Начало

Содержание



Страница 130 из 187

Назад

На весь экран

Заккрыть

и лёгок в изучении. Преимуществом его применения для создания сценариев является возможность использования, с небольшими корректировками, ранее написанных процедур на языках Visual Basic и Visual Basic for Application. Функциональные возможности сценариев, написанных на VBScript, ничем не отличаются от возможностей сценариев JavaScript: динамическое создание документа или его частей, перехват и обработка событий и т.д.

5.2 Средства создания Web-страниц

Web-редакторы

Текстовый редактор *Блокнот (Notepad)* – самый простой, входящий в стандартные средства любой ОС серии Windows. Требуется больших знаний о языке HTML, хорошего понимания структуры Web-страниц.

HomeSite – очень хороший HTML-редактор. Обладает массой достоинств, среди них: подсветка тэгов, расширенный поиск и замена сразу в нескольких файлах, встроенный редактор стилей, проверка целостности ссылок, поддержка проектов и др.

UltraEdit – мощный текстовый редактор для программистов. Удобен для одновременного редактирования нескольких файлов. Обладает некоторыми специфическими и полезными качествами: чтение и запись файлов в формате UNIX, Mac; поиск/замена в группе файлов, подсветка



Кафедра
АГyMM

Начало

Содержание



Страница 131 из 187

Назад

На весь экран

Заккрыть

тэгов, поддержка макросов, нумерация строк, неограниченный размер файла и многое-многое другое...

Графические Web-редакторы

Графические Web-редакторы – специализированные системы визуального программирования на языке HTML. Такие редакторы являются лишь вспомогательными средствами, ускоряющими процесс написания программ. В окне редакторов такого типа документ отображается практически также, как потом его покажет браузер.

FrontPage. Является наиболее популярным HTML-редактором на рынке, главным образом за счёт популярности пакета Microsoft Office. В программе предлагаются мощные функции и привлекательный пакет программ, включая Image Composer, который предназначен для создания изображений. Во FrontPage также содержатся WEB-компоненты, необходимые для добавления интерактивных свойств web-узлу. Одним из основных преимуществ программы является большое количество имеющихся шаблонов, позволяющих пользователю не ломать голову над дизайном своего проекта. Выбрав необходимый шаблон, можем приступить непосредственно к наполнению страницы контентом.

Macromedia Dreamweaver – самый лучший на сегодняшний день визуальный HTML-редактор. В отличие от остальных визуальных редакторов, создает гораздо более корректный код и очень удобен. Особенно полезен при написании динамического HTML.



Кафедра
АГуММ

Начало

Содержание



Страница 132 из 187

Назад

На весь экран

Закреть

Macromedia Flash – векторный редактор и аниматор, который позволяет создавать и встраивать в web-странички эффектные анимационные ролики.

Microsoft Word. Не смотря на то, что Microsoft Word является текстовым редактором и по существу не считается HTML-редактором. Он имеет возможность сохранять файлы в формате HTML. В результате мы можем столкнуться со страницами которые тем или иным способом были созданы в Microsoft Word. Проблемы в HTML-коде, полученном из редактора Word аналогичны проблемам FrontPage: обе эти программы создают избыточный код, который не во всём соответствует HTML-стандарту. Огромное количество людей, использующих Word в своей повседневной работе, становятся потенциальными разработчиками HTML-документов. Создать web-станицу в Word можно двумя способами: с помощью Мастера, либо преобразовав существующий документ Word в формат HTML. При этом Word сам генерирует тэги HTML, хотя и не оптимальным образом. Преобразование, естественно, приводит к тому, что какие-то элементы оформления документа будут утрачены или изменены. При необходимости вставить на HTML-страницу собственные тэги, в Word предусмотрено непосредственное редактирование HTML-кода.



*Кафедра
АГММ*

Начало

Содержание



Страница 133 из 187

Назад

На весь экран

Заккрыть

5.3 Создание Web-страницы с помощью FrontPage

Создание Web-страниц часто производится с помощью WEB-редактора Microsoft FrontPage, который является составной частью офисного пакета Microsoft Office. Заметим, что данный редактор можно приобрести как отдельную программу. FrontPage подходит как для обучения HTML-программированию, так и для разработки собственных Web-сайтов (рис. 5.1).

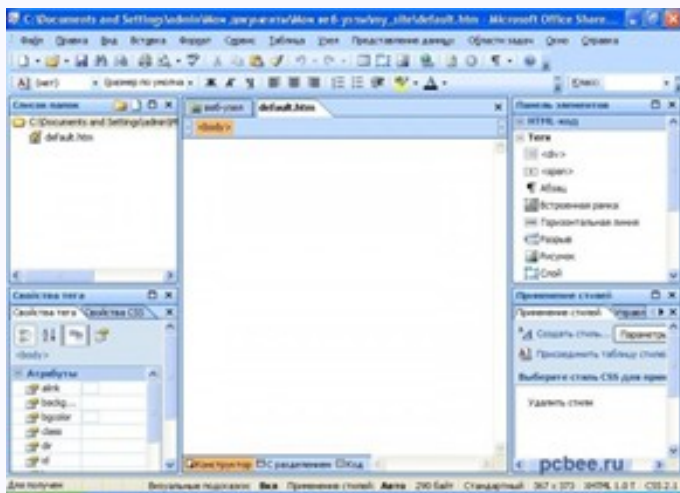


Рис. 5.1: WEB-редактор Microsoft FrontPage

К основным функциям FrontPage относятся:

1) создание и сохранение Web-страниц на винчестере компьютера и



*Кафедра
АГуММ*

Начало

Содержание



Страница 134 из 187

Назад

На весь экран

Закрыть

непосредственно в Интернет;

- 2) загрузка Web-страницы из Интернет и редактирование их;
- 3) просмотр и администрирование Web-страницы;
- 4) разработка сложного дизайна;
- 5) применение готовых HTML-тэгов;
- 6) использование готовых рисунков;
- 7) применение в Web-страницах элементов управления ActiveX и сценариев.

Создать новую Web страницу в редакторе FrontPage можно двумя способами:

- 1) выполнить команду *File -> New -> Page*;
- 2) нажать комбинацию клавиш *Ctrl+N*.

На экране появится диалоговое окно *New*. В этом окне следует либо выбрать необходимый шаблон страницы или осуществить переход на вкладку *Frames Pages (Фреймы)*. Новую страницу можно также создать с помощью кнопки *New* на стандартной панели инструментов. В этом случае новая страница будет создана по шаблону *Normal Page*.

Сохранение Web-страниц производится с помощью команды *Save* меню *File* или при нажатии комбинации клавиш *Ctrl + S*. В появившееся диалоговое окно вводится название страницы, списке *Save as type* определяется ее тип, а также можно указать ее местоположение.

Свойства созданной Web-страницы показаны в диалоговом окне *Page Properties*, которое открывается командой *File -> Properties*.



Кафедра
АГУММ

Начало

Содержание



Страница 135 из 187

Назад

На весь экран

Закрыть

Чтобы опубликовать Web-страницы, следует выбрать команду *File* – *Publish Web* или нажать одноименную кнопку стандартной панели инструментов. В образовавшемся диалоговом окне необходимо указать место размещения Web-страницы, опции опубликования измененных или всех страниц и опции защиты. При нажатии кнопки Publish созданные Web-страницы появятся в Интернет.

5.3.1 Вставка таблиц в редакторе FrontPage

Во FrontPage существует несколько способов создать таблицу.

Первый способ. Для вставки таблицы необходимо в меню *Таблица* (*Table*), выбрать пункт *Вставить* (*Insert*). В выпадающем меню нужно выбрать пункт *Таблица* (*Table*) (рис. 5.2).

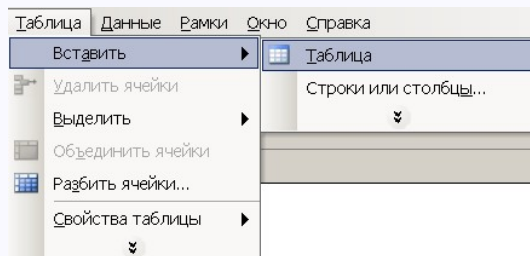


Рис. 5.2: Создания таблицы в FrontPage — с помощью меню *Таблица*

После выбора этой команды откроется окно, в котором можно указать различные параметры создаваемой таблицы (рис. 5.3).



Кафедра
АГММ

Начало

Содержание



Страница 136 из 187

Назад

На весь экран

Закрыть

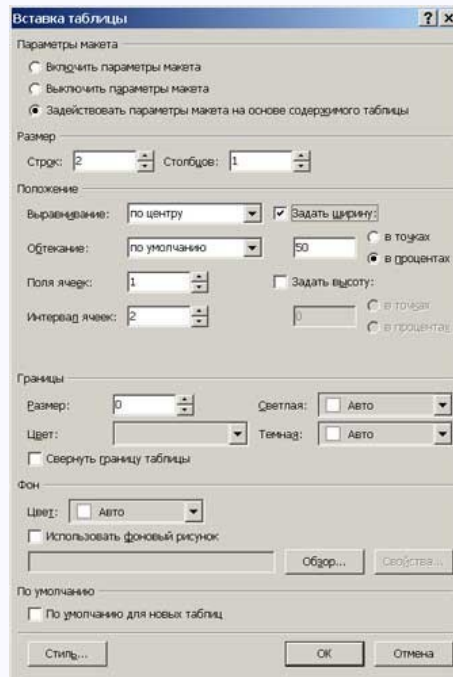


Рис. 5.3: Диалоговое окно вставки таблицы

В области *Размер* можно выбрать необходимое количество строк (*rows*) и колонок (*columns*).

В области *Положение* (*Layout*) можно настроить:

- поле *Выравнивание* (*Alignment*) служит для указания расположение таблицы на странице; доступны следующие значения:
- *По умолчанию* (*Default*) - тогда таблица разместится по усмотрению



Кафедра
АГyMM

Начало

Содержание



Страница 137 из 187

Назад

На весь экран

Закрыть

браузера пользователя, скорее всего у левого края;

- Слева (*Left*);
- Справа (*Right*);
- По центру (*Center*);
- поле *Поля ячеек (Cell Padding)* служит для установления расстояния между границей ячейки и ее содержимым;
- в поле *Интервал ячеек (Cell Spasing)* указывается расстояние между ячейками таблицы;
- для указания ширины таблицы служит флажок *Точная ширина (Specify width)*. Если он снят, размер таблицы будет зависеть только от ее содержимого. Если же флажок стоит, то можно обозначить ширину в пикселах (in pixels) или в процентах (in percent).

В области *Границы (Border)* можно настроить:

- в поле *Размер границ (Border size)* указывается ширина границ таблицы;
- можно указать цвет, светлую и темную стороны таблицы.

В области *Фон (Background)* задается цвет фона таблицы или фоновый рисунок.

Второй способ. На панели инструментов *Стандартная (Standard)* левой кнопкой мыши необходимо нажать значок *Добавить таблицу*



Кафедра
АГММ

Начало

Содержание



Страница 138 из 187

Назад

На весь экран

Заккрыть

(*Insert Table*). После этого откроется небольшой макет таблицы (рис. 5.4).



Рис. 5.4: Создания таблицы, используя кнопку Добавить таблицу (*Insert Table*)

Удерживая нажатой левую кнопку мыши необходимо выделить нужное количество строк и колонок. После того отпущения кнопки мыши, моментально появится сама таблица.

Недостатком этого способа является то, что здесь нельзя предварительно указать параметры создаваемой таблицы. Менять ее параметры можно будет только после ее создания.

Третьим способом создания таблица является рисование таблицы мышью. В строке командного меню *Таблица (Table)* нужно выбрать команду *Таблица (Table)*, после чего откроется выпадающее меню (рис. 5.5).

В этом меню нужно выбрать пункт *Нарисовать таблицу (Draw Table)*. Откроется следующее небольшое окно (рис. 5.6).

Рисование таблицы состоит из нескольких этапов: прорисовывается



Кафедра
АГУММ

Начало

Содержание



Страница 139 из 187

Назад

На весь экран

Закрыть

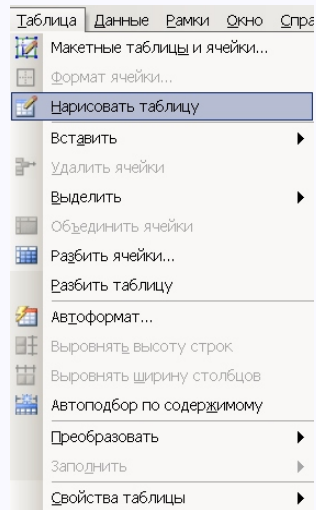


Рис. 5.5: Меню команды Таблица



Рис. 5.6: Панель инструментов Таблица

внешний прямоугольный контур таблицы, проводятся линии, разделяющие таблицу на строки и колонки, лишние линии стираются.

Вставить строку в таблицу можно одним из следующих способов:
поставить курсор на одну из строк таблицы. На панели инструментов *Таблица (Table)* нажать на кнопку *Добавить строку (Insert row)*. Новая



Кафедра
АГУММ

Начало

Содержание



Страница 140 из 187

Назад

На весь экран

Закрыть

строка появится перед строкой, на которой стоял курсор.

нужно выделить одну из существующих строк. Затем щелкнуть по выделенной строке правой кнопкой мыши и в открывшемся меню выбрать пункт *Добавить строки (Insert row)* (рис. 5.7).

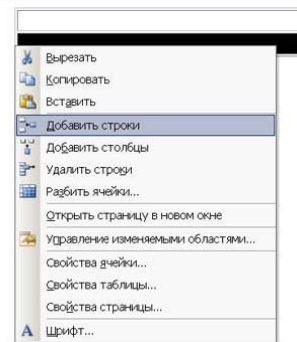


Рис. 5.7: Путь для вставки новой строки

Установить курсор в строку над (под) которой вы собираетесь добавить новые строки. Затем в меню *Таблица (Table)* выбрать *Вставка (Insert)* → *Строки или Колонки (Rows or Columns)*. Откроется следующее окно, в котором указывается количество строк, которое необходимо вставить, а также выбирается расположение новой строки — *над выделенной строкой (above)* или *под ней (below)* (рис. 5.8).

Для форматирования таблицы нужно ее выделить и выполнить команду *Таблица (Table)* → *Выделить (Select)* → *Таблица (Table)* либо нажать правой кнопкой мыши таблице и в появившемся меню выбрать



Кафедра
АГММ

Начало

Содержание



Страница 141 из 187

Назад

На весь экран

Закрыть

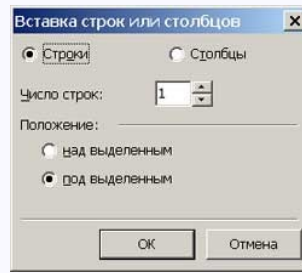


Рис. 5.8: Диалоговое окно добавления новых строк/колонок

пункт *Свойства таблицы (Table Properties)*. Откроется окно, которое практически полностью повторяет окно вставки новой таблицы. В нем можно поменять существующие настройки на новые.

Кстати, в программе FrontPage рамку таблицы можно также сделать «двойной», состоящей из множества квадратиков, с тенью и т.д. Все это доступно в верхнем меню *Формат (Format)* → *Границы и заливка (Borders and Shading)* (рис. 5.9).

В первой закладке — *Рамки (Borders)* можно поменять вид границ таблицы. Во второй закладке — *Заливка (Shading)* можно установить фоновый рисунок для таблицы а также выбрать цвет фона и текста.

5.3.2 Добавление ссылок и изображений

Для вставки изображения на страницу необходимо установить курсор на место, в которое нужно вставить картинку, и в командной строке



Кафедра
АГУММ

Начало

Содержание



Страница 142 из 187

Назад

На весь экран

Закрыть

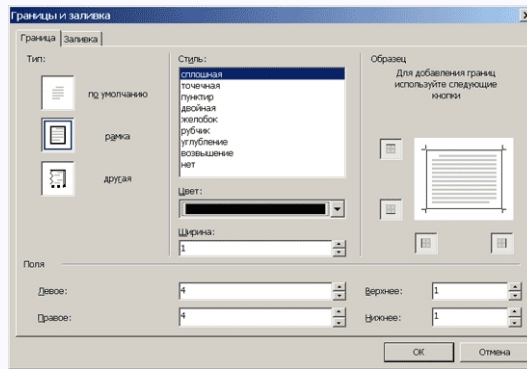


Рис. 5.9: Диалоговое окно изменения границ таблицы

выбрать меню *Вставка (Insert)* → *Рисунок (Image)* → *Из файла (From the file)*. После этого откроется окно, в котором нужно указать путь к картинке. Для задания свойств картинки нужно нажать правую кнопку мыши на картинку, и в открывшемся меню выбрать пункт *Свойства рисунка (Image Properties)*. Чтобы сократить эти действия, можно просто нажать $\langle \text{Alt} \rangle + \langle \text{Enter} \rangle$ (рис. 5.10).

Для создания гиперссылки в редакторе FrontPage нужно выделить слово, которое будет иметь гиперссылку и нажать по нему правой кнопкой мыши (не снимая выделения). В открывшемся меню нужно выбрать пункт *Гиперссылка (Hyperlink)*. Заметим, что после выделения текста можно нажать комбинацию клавиш $\langle \text{Ctrl} \rangle + \langle \text{K} \rangle$. В любом случае откроется следующее окно: (рис. 5.11).



Кафедра
АГММ

Начало

Содержание



Страница 143 из 187

Назад

На весь экран

Закрыть

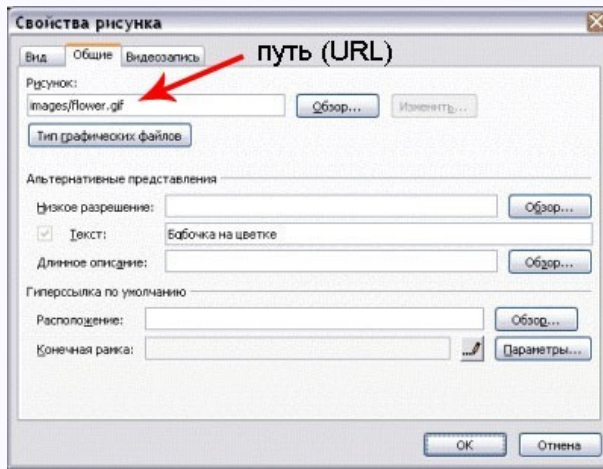


Рис. 5.10: Диалоговое окно Свойства рисунка

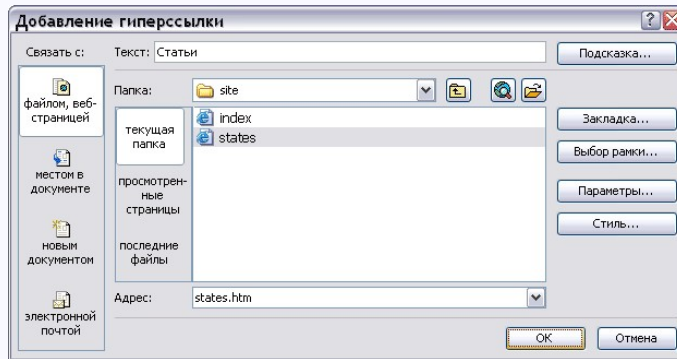


Рис. 5.11: Окно создания гиперссылки



Кафедра
АГММ

Начало

Содержание



Страница 144 из 187

Назад

На весь экран

Закрыть

РАЗДЕЛ 6

ОСНОВНЫЕ КОМПОНЕНТЫ WEB-СТРАНИЦЫ И СПОСОБЫ ИХ ВИЗУАЛЬНОГО ПРЕДСТАВЛЕНИЯ НА СТРАНИЦАХ САЙТА

6.1 Эргономика WEB-сайта

Эргономика (от греч. *ergon* / работа + *nomos* / закон) — научная дисциплина, изучающая трудовые процессы с целью создания оптимальных условий труда, что способствует увеличению его производительности, а также обеспечивает необходимые удобства и сохраняет силы, здоровье и работоспособность человека.

Эргономика в целом характеризуется двумя принципами:

- комфорт во время использования, который состоит из уменьшения физической и психологической усталости;
- безопасность, которая предполагает выбор подходящих решений для защиты пользователя.

Основные критерии эргономики в процессе создания сайта:

1. Лаконичность.

- *Простота.* Лаконичный сайт укрепит доверие к организации.
- *Не перегруженность.* Не рекомендуется использование анимированных изображений. Анимированные изображения должны



Кафедра
АГММ

Начало

Содержание



Страница 145 из 187

Назад

На весь экран

Закреть

быть закреплены за важными сообщениями, так как они привлекают внимание пользователя.

2. Четкость.

- *Ясность.* Доказано, что печатный текст труднее читать с монитора, чем с бумаги (занимает на 25% больше времени). Таким образом, текст должен быть достаточно разграничен.
- *Структурированность.* Текст должен быть структурирован по параграфам и заглавиям для облегчения чтения.
- *Расположение.* Информация должна быть расположена по степени важности. Самая важная информация сверху страницы.
- *Видимость адреса.* URL страницы должно быть видимым и понятным, что позволяет знать, где ты находишься и легко вернуться на эту страницу.
- *Осязаемость информации.* Информация должна иметь такие данные, как имя автора, дату последнего обновления и дату публикации. Плохая идея при создании сайта на главной странице размещать ссылки на страницы, которые еще не сделаны.
- *Однородность структуры.* Элементы, используемые для просмотра должны быть расположены на том же месте на каждой странице и должны иметь те же формы презентации с одной страницы на другую, если это возможно.

3. Скорость.



Кафедра
АГММ

Начало

Содержание



Страница 146 из 187

Назад

На весь экран

Закрыть



- *Время загрузки.* Время загрузки страницы должно быть как можно короче, потому что никто не будет ждать более 15 секунд загрузки. Этот фактор в основном зависит размера страницы, изображений на ней и возможностей сервера.
- *Оптимизированные изображения.* Хорошей идеей является оптимизация размера изображений, выбирая подходящий размер и цвета. Рекомендация — не загружать изображения более 30-40 Кб (можно предложить скачать изображения более высокого качества).

4. Взаимодействие.

- *Гипертекстовые ссылки.* Интерактивность - это возможные взаимодействия между пользователем и веб-сайтом. Гипертекстовые ссылки предлагают пользователям широкие возможности в этой области и дают посетителям несколько путей, которые они могут использовать, как им нравится. Веб-дизайнеры советуют размещать достаточно ссылок между различными страницами. Тем не менее, слишком много ссылок может сделать чтение трудным.
- *Сегментация информации.* Для того, чтобы посетители могли получать информацию более легко и, в некоторых случаях, для того, чтобы возбудить любопытство, будет хорошей идеей сегментировать информацию. В частности, сегментирование инфор-

мации может быть сделано при наличии гипертекстовых ссылок, которые ведут к остальной части этой статьи.

- *Содействие взаимодействию.* Хорошей идеей будет создание сайта легкого для взаимодействия с посетителями, особенно для того, чтобы высказать свои впечатления и мысли по работе сайта. Как минимум, должна быть обратная связь, благодаря которой можно связаться с администратором по электронной почте, заполнив контактную информацию.

5. Адаптивность. Адаптивность характеризуется способностью веб-сайта автоматически становится персонализированным без вмешательства пользователя.

- *Изменение размера шрифта.* Текст сайта должен предпочтительно не использовать шрифты, размеры которых не выражаются по абсолютной величине. Таким образом, пользователь может изменить размер шрифта, если он того пожелает.
- *Доступ ко всему.* Доступность означает возможность всем типам пользователей, включая слабовидящих и слепых людей, иметь доступ к веб-сайту. Следует придерживаться определенных правил доступности, чтобы дать доступ большему количеству людей, независимо от их компьютеров или физических недостатков.
- *Взаимодействие.* Соблюдение стандартов, в частности, рекомендации W3C, помогает гарантировать хороший уровень взаимо-



Кафедра
АГУММ

Начало

Содержание



Страница 148 из 187

Назад

На весь экран

Закрыть

действия, то есть способность для посетителей, использующих различное программное обеспечение иметь доступ к веб-сайту.

6. Доступность.

- *Принцип прозрачности.* Предпочтительно, чтобы форматы, используемые для создания сайта были прозрачными, то есть их можно было прочитать в текстовом режиме. Таким образом, предпочтительно использовать HTML вместо Flash. Кроме того, изображения и анимация не должны отвлекать внимание от текстовой информации. Таким образом, графические иллюстрации должны быть использованы только для визуального дополнения текста.
- *Подпись.* Вместо изображений должна присутствовать подпись (используйте атрибут alt) . Это необходимо для того, чтобы слабовидящие люди поняли смысл изображения.
- *Выбор цвета.* Цвета должны быть подобраны таким образом, чтобы даже дальтоники могли точно различить цвета.
- *Правильное использование стилей.* Информация должна быть доступна даже без стилей.
- *Контраст.* Между цветом фона и текстом должен быть достаточный контраст, чтобы слабовидящие люди могли прочитать текст.
- *Изменяемый размер шрифта.* Размер шрифта должен быть



Кафедра
АГММ

Начало

Содержание



Страница 149 из 187

Назад

На весь экран

Заккрыть

адаптирован таким образом, чтобы пользователи могли увеличить размер шрифта, если это необходимо.

Кроме того, шрифты не должны быть настолько малы, чтобы утомлять глаза и делать текст неразборчивым.

Рассмотрим основные правила.

1. Оптимальное значение времени ожидания пользователя — 2 секунды.
2. Основная мысль на любой странице раскрывается в начале. Необходимо наличие элемента навигации, представляющего собой путь по сайту от его «корня» до текущей страницы.
3. Обязательны заметные точки начала (реализуют возможность всегда вернуться на предыдущие страницы).
4. Между главной страницей и второстепенными должно быть визуальное сходство.
5. Требуются описание цели и назначения сайта.

6.2 Требования к иллюстрациям в Internet

Общие пожелания для подготовки изображений:

- графические элементы должны быть представлены в формате GIF или PNG;



*Кафедра
АГyMM*

Начало

Содержание



Страница 150 из 187

Назад

На весь экран

Закреть

- фотографии должны быть представлены в формате JPEG;
- необходимо обеспечить показ изображений с малым разрешением, пока идет загрузка больших изображений;
- используются GIF-изображения с чересстрочным форматом;
- указываются атрибуты height и width элемента img;
- необходимо сократить количество анимации;
- по возможности используются пиктограммы изображений;
- обязательно создается «альтернативный» текст (атрибут alt).

6.3 Обзор типов файлов иллюстраций

Чтобы не затеряться на просторах Internet и привлечь к себе внимание пользователей, никак нельзя обойтись без графического оформления Web-страниц и узлов. Однако здесь на пути разработчиков возникает проблема: графические технологии для Web не успевают в своем развитии за другими технологиями, и возможности в данной области остаются весьма ограниченными.

В самом деле, два наиболее популярных в настоящее время графических формата Internet – GIF и JPEG – являются уже довольно старыми. Конечно, не-удачными назвать их никак нельзя, ведь сам факт столь длительного их существования – свидетельство этому. Но, с другой сто-



*Кафедра
АГММ*

Начало

Содержание



Страница 151 из 187

Назад

На весь экран

Закреть

роны, вряд ли можно поспорить с тем, что возможности данных форматов не отвечают современным требованиям в области графики. Так, формат GIF поддерживает только 256-битовый цвет, а в случае применения формата JPEG при большой степени сжатия существенно снижается качество изображения. Кроме того, еще в 1995 г. возможность свободного использования GIF оказалась под вопросом, когда компании Unisys, которой принадлежит реализованный в этом формате алгоритм сжатия LZW, и CompuServe, разработавшей сам формат, собрались взимать лицензионные отчисления с каждой программы, использующей его.

В сложившейся ситуации группа независимых разработчиков Internet приняла решение о разработке формата, который соответствовал бы или даже превосходил по своим возможностям GIF, но был при этом простым в создании и полностью мобильным. Новый формат получил название Portable Network Graphics (PNG) и был одобрен консорциумом W3C в 1996 г.

Формат PNG поддерживает 48-битовые цветные и 16-битовые черно-белые изображения и обеспечивает более быструю их загрузку, чем формат GIF. Он также включает в себя немало дополнительных возможностей, например альфа-каналы (alpha channel), позволяющие устанавливать уровень прозрачности для каждого пиксела, и гамма-коррекцию. Механизм сжатия изображения в PNG реализован на базе фильтров, позволяющих оптимизировать данные перед сжатием, и алгоритма LZ77, применяемого в ZIP-архиваторах.



*Кафедра
АГуММ*

Начало

Содержание



Страница 152 из 187

Назад

На весь экран

Закрыть

Однако, несмотря на ряд преимуществ PNG пока не удалось стать реальной альтернативой GIF и JPEG. Виной тому было отсутствие поддержки со стороны разработчиков браузеров. Правда, к сегодняшнему дню в данном направлении произошли существенные сдвиги: начиная с Internet Explorer 4.0 и Netscape Navigator 4.04 поддержка PNG реализована непосредственно в браузерах; до этого она обеспечивалась за счет встраиваемых компонентов. Следующим по популярности растровым форматом для Web можно назвать FlashPix, разработанный группой компаний: Kodak, Hewlett–Packard, Microsoft и Live Picture. Он базируется на принципах JPEG–компрессии, но содержит ряд усовершенствований, которые позволяют уменьшить степень искажения изображений. Основное преимущество данного формата – многоуровневая организация файла. В начале загружается изображение с самым низким разрешением и впоследствии, по мере надобности, подкачивается более качественная версия.

Интересной разработкой обладает компания Iterated Systems, которая создала свой формат на основе фрактальной компрессии (Fractal Image Format, FIF), а также выпустила программу преобразования основных форматов в FIF и плагины для просмотра сжатых по фрактальному алгоритму изображений в основных браузерах. К сожалению, фрактальная компрессия, как и JPEG, имеет существенный недостаток: согласно этим алгоритмам, для анализа изображение перед сжатием разбивается на отдельные блоки, что затрудняет его постепенную прорисовку при



*Кафедра
АГyMM*

Начало

Содержание



Страница 153 из 187

Назад

На весь экран

Закрыть

загрузке с Web-сайта.

Наиболее перспективные – растровые форматы, основанные на алгоритмах wavelet-сжатия. В этой области ведут разработки практически все компании, которые занимаются созданием графических форматов. Самым многообещаемым является, безусловно, JPEG 2000. Работа над ним еще не завершена, но заявленные параметры впечатляют: 256 каналов цвета, что позволит формату работать с любым цветовым пространством и поддерживать множество альфа-каналов; встраивание ICC-профилей; неограниченное поле для метаданных. Но главное преимущество wavelet-технологии – потоковость. Wavelet-поток можно прервать в любое время, при этом изображение все равно воспроизводится, только качество его будет зависеть от количества загруженных данных.

Компания AT & T разработала и собственный формат на основе wavelet-компрессии – DjVu. Его главная особенность – распознавание текста при ком-прессии содержащих его изображений и сжатие отдельно графического и текстового слоя. По утверждению компании, основным предназначением этого формата и является публикация в Web сканированных документов. На сайте AT & T по адресу djvu.research.att.com можно получить бесплатный плагин для просмотра DjVu –файлов, а также целую библиотеку, опубликованную в этом формате.

Как бы ни были хороши вышеперечисленные форматы, всех их объединяет один недостаток – растр. Например, реализованные с их помощью изображения довольно сложно модифицировать и даже масшта-



*Кафедра
АГММ*

Начало

Содержание



Страница 154 из 187

Назад

На весь экран

Закрыть

бировать. Кроме того, несмотря на использование различных методов сжатия, они все-таки имеют немалый размер, а следовательно, и относительно большое время загрузки, что для Web-графики является особенно критичным.

Векторная графика основана не на хранении информации о каждом пикселе, а на командах рисования линий и заполнения форм. Используется она уже довольно давно, но в отличие от традиционных замкнутых форматов векторные форматы для Web построены на базе открытых стандартов, главным образом языков маркировки, в которых для определения тегов и других элементов применяется обычный текст, что значительно упрощает манипулирование свойствами изображений. Преимуществами векторной графики на основе языков маркировки являются также возможности выбора, индексирования и поиска элементов изображения и привязки ее к другим элементам.

Однако говорить о массовом внедрении векторной графики в Web пока еще рано, в первую очередь из-за отсутствия единого формата.

Наиболее распространенным в данный момент является формат, разработанный компанией Macromedia, – Flash. Благодаря своим уникальным возможностям его последняя версия очень быстро завоевала популярность. Flash 3 поддерживает анимацию по кейфреймам, морфинг, прозрачные объекты, гиперссылки, встраивание звуковых и видеофайлов. Средства для его создания достаточно просты в пользовании, хорошо документированы, плагины для просмотра распространяются бесплат-



*Кафедра
АГуММ*

Начало

Содержание



Страница 155 из 187

Назад

На весь экран

Закрыть

но, а размер выходных файлов крайне мал.

Но все его преимущества, к сожалению, блекнут перед одним единственным недостатком, который заставил Macromedia отказаться от дальнейшей разработки формата. Этот недостаток – закрытость, ведь файл Flash – двоичный. Таким образом, его можно редактировать только в специальной программе. Поэтому в последнее время различными компаниями и организациями предложен целый ряд языковых форматов, и каждый из них претендует на роль единого стандарта. В число таких форматов входят Web Schematics, DrawML, PGML и VML.

Web Schematics представляет собой язык гипертекстовой маркировки для создания чертежей и диаграмм. Его разработчики попытались создать аналог функций рисования, используемых в базовых графических средствах систем воспроизведения документов, таких, как Adobe FrameMaker и Microsoft Word. Данный формат использует модели рендеринга и представлений HTML и CSS1.

6.4 Форматы JPEG, GIF, PNG

GIF – формат, использующий алгоритм сжатия без потерь информации LZW. Максимальная глубина цвета – 8 бит (256 цветов). В нём также есть возможность записи анимации. Поддерживает прозрачность пикселей (двухуровневая – полная прозрачность, либо полная непрозрачность). Данный формат широко применяется при создании Web-страниц.



Кафедра
АГУММ

Начало

Содержание



Страница 156 из 187

Назад

На весь экран

Закрыть

GIF–формат позволяет записывать изображение «через строчку», благодаря чему, имея только часть файла, можно увидеть изображение целиком, но с меньшим разрешением. Его выгодно применять для изображений с малым количеством цветов и резкими границами (например, для текстовых изображений).

JPEG (JPG) – формат, использующий алгоритм сжатия с потерями информации, который позволяет уменьшить размер файла в сотни раз. Глубина цвета – 24 бит. Не поддерживается прозрачность пикселей. При сильном сжатии в области резких границ появляются дефекты. Формат JPEG хорошо применять для сжатия полноцветных фотографий. Учитывая то, что при повто-ном сжатии происходит дальнейшее ухудшение качества, рекомендуется сохранять в JPEG только конечный результат работы. JPEG широко применяется при создании Web–страниц, а также для хранения больших коллекций фотографий.

Сравнение GIF и JPEG

- GIF – формат удобен при работе с рисованными картинками;
- JPEG – формат лучше использовать для хранения фотографий и изображений с большим количеством цветов;
- для создания анимации и изображений с прозрачным фоном применяется GIF–формат.

BMP – это формат графического редактора Paint. В нём не применяется сжатие. Он хорошо подходит для хранения очень маленьких изображений – таких как иконки на рабочем столе. Большие же файлы



Кафедра
АГММ

Начало

Содержание



Страница 157 из 187

Назад

На весь экран

Закрыть

в этом формате занимают слишком много места.

PNG – разработан с целью заменить формат *GIF*. Использует алгоритм сжатия *Deflate* без потерь информации (усовершенствованный *LZW*). Максимальная глубина цвета – 48 бит. Поддерживает каналы градиентных масок прозрачности (256 уровней прозрачности). *PNG* – относительно новый формат, и поэтому ещё не очень распространён. В основном используется в *Web*-дизайне.

6.5 WEB-оптимизация изображений

Оптимизация изображений – это то, чем пренебрегать категорически не рекомендуется. Это простейший способ ускорить веб-страницу в несколько раз, и позаботиться о посетителях своего сайта. Различают два вида оптимизации.

Первый вид: *оптимизация без потерь*. Представляет собой процесс, итогом которого является получение оптимизированного изображения, ничем не отличающейся от оригинала.

Второй вид: *оптимизация с потерями*. В результате такой оптимизации изображение имеет более низкое качество и выглядит заметно хуже.

Существует достаточно большое количество инструментов, как в виде специализированных программ, так и онлайн-приложений, которые работают быстро, качественно и без лишнего шума оптимизировать изображения. Рассмотрим некоторые из них.



Кафедра
АГуММ

Начало

Содержание



Страница 158 из 187

Назад

На весь экран

Заккрыть

Irfan View. Достаточно функциональная программа, имеющая достаточно удобный «просмотрщик» изображений. Достаточно быстро работает, имеет большое количество функций. Также в данной программе предусмотрена пакетная обработка фотографий.

Riot (Radical Image Optimization Tool). Если Вы не хотите разбираться в настройках, то эта программа - то, что вам нужно. Программа может: оптимизировать картинки, поворачивать, сжимать до определенного объема, сменять размеры в точках и многое другое.

Smush.it Популярный сервис от Yahoo, который позволяет удалять ненужные байты из картинок. Не смотря на то, что язык у данного сервера английский, пользоваться им весьма просто.

Dynamicdrive. Данный инструмент позволяет осуществлять не только контроль уровня сжатия, но и может конвертировать один формат изображения в другой.

PS Photoshop. Оптимизировать картинку можно и с помощью PS Photoshop. Для этого необходимо открыть картинку в PS Photoshop и выполнить *Images -> Adjustments -> Posterize*. Появится окно с ползунком, который реагирует на различные цветовые области изображения. Для каждой картинку ползунок имеет свои параметры, поэтому его положение определяется только опытным путём. В завершение нужно сохранить изображение через *File -> Save for web*.



Кафедра
АГММ

Начало

Содержание



Страница 159 из 187

Назад

На весь экран

Закрыть

ПРАКТИЧЕСКИЙ РАЗДЕЛ

Лабораторная работа №1

1. Изучите теоретический материал пунктов **2.1–2.10**.
2. Создайте HTML-документ с автобиографией. Документ **должен** содержать: ФИО, Ваше фото, а также прочую информацию, необходимую для автобиографии. Найдите применение тегам `<H1>...</H1>`, `<PRE>...</PRE>`, `...`, `<I>...</I>`, `<U>...</U>`, `<S>...</S>`. Использовать ссылки, метки, списки, фон документа, а также свойства шрифта.

Лабораторная работа №2

1. Изучите теоретический материал пункта **2.11**.
2. Используя тэг `<TABLE>`, создайте следующую таблицу:



Кафедра
АГММ

Начало

Содержание



Страница 160 из 187

Назад

На весь экран

Закреть

Вариант 1.

Вариант 2.

Вариант 3.



*Кафедра
АГyMM*

Начало

Содержание



Страница 161 из 187

Назад

На весь экран

Закрyть

Вариант 4.

Вариант 5.

Вариант 6.



Кафедра
АГyMM

Начало

Содержание



Страница 162 из 187

Назад

На весь экран

Закрyть

Вариант 7.

Вариант 8.

Вариант 9.



Кафедра
АГyMM

Начало

Содержание



Страница 163 из 187

Назад

На весь экран

Закрyть

Вариант 10.

Лабораторная работа №3

1. Изучите теоретический материал пункта **2.12**.
2. Создайте документ html со следующими фреймами:
 - 1) шапка сайта (содержит: название сайта, слоган и прочее);
 - 2) меню (содержит ссылки: автобиография, контактная информация);
 - 3) содержимое (в этом фрейме отображается соответственно автобиография либо контактная информация);
 - 4) информация о разработчике (фамилия, имя, группа, год разработки, номер зачетной книжки).

Пояснения по выполнению:

В каждый фрейм информация подгружается из html-документов. Т.о. у Вас должно быть 6 html-файлов:

- шапка сайта;



Кафедра
АГММ

Начало

Содержание



Страница 164 из 187

Назад

На весь экран

Закреть

- меню;
- автобиография (см. Лабораторная работа №1);
- контактная информация;
- информация о разработчике;
- html-документ с фреймами, в которые загружаются предыдущие пункты.

Фреймы 1, 2 и 4 не меняют своего содержания. При нажатии по ссылке во фрейме с меню (фрейм 2) фрейм 3 меняет свое содержание.

Вариант 1.

1	
2	3
4	

Вариант 2.

1
2
3
4



*Кафедра
АГММ*

Начало

Содержание



Страница 165 из 187

Назад

На весь экран

Закреть

Вариант 3.

1	2	4
	3	

Вариант 4.

1	
2	3
4	

Вариант 5.

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



Кафедра
АГyMM

Начало

Содержание



Страница 166 из 187

Назад

На весь экран

Заккрыть

Вариант 6.

1	4
2	
3	

Вариант 7.

4	
3	
1	2

Вариант 8.

1		
2	3	2
4		



Кафедра
АГyMM

Начало

Содержание



Страница 167 из 187

Назад

На весь экран

Закрyть

Вариант 9.

1	
2	
2	3
4	

Вариант 10.

1	4
2	
3	
2	

Лабораторная работа №4

1. Изучите теоретический материал раздела 3.
2. Дано 5 гиперссылок. При прохождении курсора мыши над каждой из них гиперссылки видоизменяются:

Вариант 1. Увеличиваются.



Кафедра
АГyMM

Начало

Содержание



Страница 168 из 187

Назад

На весь экран

Закреть

Вариант 2. Изменяется цвет гиперссылок.

Вариант 3. Изменяется жирность текста гиперссылок.

Вариант 4. Изменяется курсивность текста гиперссылок.

Вариант 5. Изменяется фон гиперссылок.

Вариант 6. Текст гиперссылок зачеркивается.

Вариант 7. Текст гиперссылок подчеркивается.

Вариант 8. Изменяется начертание шрифта гиперссылок (cursive/arial).

Вариант 9. Текст гиперссылок подчеркивается пунктирной линией.

Вариант 10. Уменьшаются.

Примечания:

`itmja.className='itmja_classa'` — задать другой стиль;

`onmouseover` — событие, наступающее при наведении курсора мыши на элемент;

`onmouseout` — событие, наступающее при уведении курсора мыши с элемента.

Лабораторная работа №5

1. Изучите теоретический материал раздела 3.
2. Создайте стилевой файл содержащий оформление web-сайта. Обязательно определение оформления следующих элементов:

- заголовки 1-4 уровней;



Кафедра
АГуММ

Начало

Содержание



Страница 169 из 187

Назад

На весь экран

Закреть

- пункты меню 1-4 уровней;
- гиперссылки;
- таблицы и ячейки таблиц;
- основной текст.

3. Для оформления страницы в графическом редакторе PhotoShop разработайте:

- баннер;
- разделители;
- списковые элементы (буллеты);
- пиктограммы и иконки (домой, назад, и т.д.).

Лабораторная работа №6

1. Изучите теоретический материал пунктов **2.13**, **4.1**, **4.2** и **4.4**.

2. Создайте html-страницу, содержащую 3 поля text и 1 button, как на рисунке:

При нажатии на кнопку, в text 3 должен выводиться результат (обозначим для удобства text 1 – a, text 2 – b):

Вариант 1. $\sqrt{a} + b^2$.

Вариант 2. $|a| + \sin b$.



*Кафедра
АГуММ*

Начало

Содержание



Страница 170 из 187

Назад

На весь экран

Закреть

Вариант 3. $\sqrt{a^2 + b^2}$.

Вариант 4. $a^2 - b^2$.

Вариант 5. a^b .

Вариант 6. Максимальное число из a и b .

Вариант 7. Минимальное число из a и b .

Вариант 8. $e^a + e^b$.

Вариант 9. a и b (например, если $a = 3435$, а $b = 232$, то результат равен 3435232).

Вариант 10. $\frac{a}{b^2}$.

Лабораторная работа №7

1. Изучите теоретический материал пунктов **4.1** и **4.2**.
2. При наведении курсора мыши на одну из шести гиперссылок в



HTML

Кафедра
АГММ

Начало

Содержание



Страница 171 из 187

Назад

На весь экран

Заккрыть

строке состояния появляется ее текст. При уведении со ссылки:

Вариант 1. Бегущая строка слева направо "Нажмите на ссылку!" в строке состояния;

Вариант 2. Мерцающая строка слева направо "Нажмите на ссылку!" в строке состояния;

Вариант 3. Бегущая строка справа налево "Не жмите на ссылку!!!" в строке состояния;

Вариант 4. Выбегающая слева строка "Нажмите на ссылку!" в строке состояния ;

Вариант 5. Выбегающая справа строка "Жми на ссылку!" в строке состояния ;

Вариант 6. Через каждые 5 секунд текст в строке состояния обновляется: случайным образом выбирается из массива пяти строк в строке состояния;

Вариант 7. Бегущая строка слева направо "Нажмите на ссылку!" в титульной строке;

Вариант 8. Мерцающая строка слева направо "Нажмите на ссылку!" в титульной строке;

Вариант 9. Через каждые 3 секунды текст в титульной строке обновляется: случайным образом выбирается из массива пяти строк;

Вариант 10. Выбегающая слева строка "Нажмите на ссылку!" в титульной строке;



Кафедра
АГММ

Начало

Содержание



Страница 172 из 187

Назад

На весь экран

Закрыть

Вариант 11. Выбегающая справа строка "Жми на ссылку!" в титульной строке;

Примечания:

t.substr(1,t.length-1) — метод класса String. Возвращает фрагмент строки заданной величины. Если второй параметр пропущен, выбираются все символы до конца строки;

t.charAt(0) — метод класса String. Возвращает символ, номер которого передан в качестве параметра;

document.title=' ' — помещает текст в титульную строку;

window.status=' ' — помещает текст в строку состояния;

Math.ceil() — возвращает ближайшее целое число, большее или равное аргументу;

Math.random() — возвращает псевдослучайное число от 0 включительно до 1 исключительно;

onmousemove — событие, наступающее при прохождении курсора над элементом.

Лабораторная работа №8

1. Изучите теоретический материал пункта 4.4.
2. Создать html файл с необходимыми объектами для решения поставленных задач:



Кафедра
АГММ

Начало

Содержание



Страница 173 из 187

Назад

На весь экран

Закреть



Начало

Содержание



Страница 174 из 187

Назад

На весь экран

Закреть

Вариант 1. Сколько минут и секунд прошло с начала текущей пары (время начала пары вводится пользователем);

Вариант 2. Сколько осталось времени до конца текущей пары (время окончания пары пользователь вводит самостоятельно);

Вариант 3. Определить, сколько осталось дней, часов, минут и секунд до Парада на 9 Мая (начало Парада - 10.00);

Вариант 4. В каком из знаков зодиака "расположен" текущий день;

Вариант 5. Сколько дней, часов, минут и секунд осталось до дня рождения Вашего друга;

Вариант 6. Определить количество дней между двумя введенными пользователем датами;

Вариант 7. Сколько дней, часов, минут и секунд прошло после наступления Нового года;

Вариант 8. Сколько дней, часов, минут и секунд осталось до наступления Нового года;

Вариант 9. Определить, сколько осталось дней, часов, минут и секунд до Вашего дня рождения;

Вариант 10. Сколько дней, часов, минут и секунд осталось до даты, случайно выбранной компьютером;

Вариант 11. Сколько дней, часов, минут и секунд осталось до даты, введенной пользователем.

Лабораторная работа №9

1. Изучите теоретический материал раздела 4.
2. Создайте игру.

Вариант 1. На какой-нибудь страничке реализовать игру "Угадай цвета". Описание игры: наборы цветов записаны в текстовом поле с ниспадающим меню. Компьютер случайным образом загадывает четыре цвета. Задача пользователя, выбирая различные цвета из ниспадающего меню, угадать загаданный компьютером набор. Если в каком-то поле цвет уже совпал с загаданным, компьютер сообщает об этом пользователю.

Вариант 2. Указать автора книги (из поля с ниспадающим меню), названной компьютером. Оценивать выбранный ответ.

Вариант 3. Рулетка.

Вариант 4. Указать исполнителя песни, автора слов и музыки.

Вариант 5. Быки и коровы.

Вариант 6. Указать актера, сыгравшего роль, загаданную компьютером. Оценивать знания.

Вариант 7. Угадай слово (компьютер загадывает слово, переставляет в нем буквы, пользователь угадывает).

Вариант 8. Реализовать на страничке развлекательную игру "Составление жуткой истории". Описание игры: на начальной странице игры располагается несколько строк ввода текста с пояснениями, что вво-



Кафедра
АГУММ

Начало

Содержание



Страница 175 из 187

Назад

На весь экран

Закрыть

дить в эти строки. Пользователь после внесения информации в строки нажимает кнопку "Рассказать" после чего в новом окне появляется готовый рассказ, так называемая "жуткая история" в которой используется введенная пользователем информация.

Вариант 9. Угадай слово, загаданное компьютером ("Поле чудес").

Вариант 10. Угадай число из интервала, загаданное компьютером.

Требования:

- 1) предусмотреть модальное окно для выбора параметров игры;
- 2) временной контроль за игрой;
- 3) использование элементов управления.

Лабораторная работа №10

Реализуйте авторский бизнес-проект. Создайте сайт для некоторой вымышленной организации (например, интернет-магазин). Проект требует самостоятельного внедрения новых идей и оценки существующей информации.



*Кафедра
АГММ*

Начало

Содержание



Страница 176 из 187

Назад

На весь экран

Закреть

ТЕСТЫ

ТЕСТ №1. Тест по теме «Язык разметки гипертекста HTML»

1. Каким тегом объявляется web-страница?

- 1) `<html> ... </html>`
- 2) `<head> ... </head>`
- 3) `<title> ... </title>`
- 4) `<body> ... </body>`

2. Каким тегом объявляется заголовок web-страницы?

- 1) `<html> ... </html>`
- 2) `<head> ... </head>`
- 3) `<title>...</title>`
- 4) `<body> ... </body>`

3. В какой тег заключается основное содержание web-страницы?

- 1) `<html> ... </html>`
- 2) `<head> ... </head>`
- 3) `<title> ... </title>`
- 4) `<body> ... </body>`

4. В какой тег заключается название web-страницы?

- 1) `<html> ... </html>`
- 2) `<head> ... </head>`



Кафедра
АГуММ

Начало

Содержание



Страница 177 из 187

Назад

На весь экран

Закреть

3) `<title> ... </title>`

4) `<body> ... </body>`

5. Какой код для пустой web-страницы правильный?

1) `<html> <head> <title> </head> <body> </body> </html>`

2) `<html> <head> <title> </title> </head> <body> </body>`

`</html>`

3) `<html> <head> <title> <body> </body> </html>`

4) `<html> <head> <title> </title> </head> <body> </body>`

6. В каком коде абзац "Системы счисления" выровнен по центру

1) `<body> <h3 align = "center" > Системы счисления </h3>`

`</body>`

2) `<body> <p align = "center" > Системы счисления </p>`

`</body>`

3) `<body> Системы счисления `

`</body>`

7. Какой html-код задает вывод текста в две строки

1) `<p> Информационные
 технологии </br> </p>`

2) `<p>
 Информационные технологии </br> </p>`

3) `<p> Информационные
 технологии </p>`

8. Каким тегом задается вставка изображения на web-страницу?

1) ` `

2) ``



Кафедра
АГУММ

Начало

Содержание



Страница 178 из 187

Назад

На весь экран

Закрыть

3) ` `

4) ``

9. Каким тегом задается вставка гиперссылки на web-страницу?

1) ` `

2) ``

3) ` `

4) ``

10. Каким тегом задается цвет текста на web-странице?

1) ` `

2) ``

3) ` `

4) ``

11. Каким тегом задается метка на web-странице?

1) ` `

2) ``

3) ` `

4) ``



Кафедра
АГyMM

Начало

Содержание



Страница 179 из 187

Назад

На весь экран

Закрyть

ТЕСТ №2. Тест на знание Javascript

1. JavaScript – это

- 1) *Объектно-ориентированный скриптовый язык программирования.*
- 2) *Скриптовой язык программирования.*
- 3) *Прототипно-ориентированный скриптовый язык программирования.*
- 4) *Все выше перечисленное верно.*

2. Где в документе может располагаться тэг script?

- 1) *В HEAD или в BODY*
- 2) *До HTML*
- 3) *Только в BODY*
- 4) *Только в HEAD*
- 5) *Не имеет значения*

3. JavaScript используемым для разработки

- 1) *Веб-приложений на стороне сервера*
- 2) *Веб-приложений на стороне клиента*
- 3) *Оба варианта верны*

```
4. var a = 1;  
var b = ++a+a;  
alert(b);
```



Кафедра
АГиММ

Начало

Содержание



Страница 180 из 187

Назад

На весь экран

Закреть

Результат работы кода - сообщение с текстом

- 1) 1
- 2) 2
- 3) 3
- 4) 4
- 5) *Этот код не будет работать*

```
5. var a = 1;  
var b = a++;  
alert(b);
```

Результат работы кода - сообщение с текстом

- 1) 1
- 2) 2
- 3) *Другое*

```
6. var a = []  
a[1] = 1  
a[5] = 2  
alert(a.length)
```

Результат работы кода - сообщение с текстом

- 1) 1
- 2) 2
- 3) 3
- 4) 4



*Кафедра
АГyMM*

Начало

Содержание



Страница 181 из 187

Назад

На весь экран

Закрyть

5) 5

6) 6

7) Другое

```
7. var a = []
```

```
a[1] = 1
```

```
a[3] = 2
```

```
delete a[3]
```

```
alert(a.length)
```

Результат работы кода - сообщение с текстом

1) 1

2) 2

3) 3

4) 4

5) Другое

8. Какие из этих вариантов не задает массив из элементов "a", "b"?

1) `var a = new Array("a "b")`

2) `var a = ["a "b"]`

3) `var a = ("a "b")`

4) Все подойдут

5) Ни один не задает

9. Как считать свойство b объекта a?

1) `a[b]`



Кафедра
АГУММ

Начало

Содержание



Страница 182 из 187

Назад

На весь экран

Закреть

- 2) *a::b*
- 3) *a->b*
- 4) *a.b*

10. function c()

```
var a = 1;
```

```
if(a == 1)
```

```
var b = 2;
```

```
return b;
```

```
alert(c());
```

- 1) *0*
- 2) *2*
- 3) *none*
- 4) *undefined*

ОТВЕТЫ

ОТВЕТЫ на тест №1:

1) 1; 2) 2; 3) 4; 4) 3; 5) 2; 6) 2; 7) 3; 8) 2; 9) 3; 10) 1; 11) 4.

ОТВЕТЫ на тест №2:

1) 3; 2) 1; 3) 3; 4) 4; 5) 1; 6) 6; 7) 4; 8) 3; 9) 4; 10) 2.



*Кафедра
АГyMM*

Начало

Содержание



Страница 183 из 187

Назад

На весь экран

Закреть

Вопросы к зачету

1. Основные термины и определения Web-технологий.
2. Язык HTML. Понятие тега. Структура HTML документа.
3. Раздел документа HEAD. Раздел документа BODY.
4. Форматирование текста HTML-документа.
5. Теги логического форматирования текста HTML документа.
6. Физическое форматирование текста HTML документа.
7. Организация текста внутри HTML-документа (вложенные списки).
8. HTML-теги для добавления картинок и гиперссылок.
9. HTML-теги для создания таблиц.
10. Фреймы.
11. HTML-формы. Основные элементы формы.
12. Каскадные таблицы стилей (CSS). Правила CSS.
13. Каскадные таблицы стилей (CSS). Методы применения каскадных таблиц стилей.
14. Общие сведения о JavaScript. Добавление сценария JavaScript на Web-страницу.
15. Переменные в JavaScript. Типы данных в JavaScript.
16. Операторы в JavaScript.
17. Использование функций в JavaScript.
18. Функции работы со временем в JavaScript. Отображение текущего времени на Web-странице.



Кафедра
АГуММ

Начало

Содержание



Страница 184 из 187

Назад

На весь экран

Закреть

19. Определение и вызов пользовательских функций в JavaScript.
20. Массивы и строки в JavaScript.
21. Объект Date в JavaScript.
22. Объект Array в JavaScript.
23. Создание и редактирование страниц с помощью редактора WEB-страниц FrontPage. Шаблоны WEB-узлов.
24. Вставка таблиц в редакторе FrontPage.
25. Добавление ссылок и рисунков в редакторе FrontPage.
26. Обзор типов файлов иллюстраций. Форматы JPEG, GIF, PNG.
27. Требования к иллюстрациям в Internet.
28. Эргономика WEB-сайта.



*Кафедра
АГyMM*

Начало

Содержание



Страница 185 из 187

Назад

На весь экран

Закрyть

Список использованной и рекомендованной литературы

1. Дронов, В. А. JavaScript и AJAX в Web-дизайне / В. А. Дронов. – СПб.: БХВ-Петербург, – 2008. – 736 с.
2. Кент, П. Интернет / П. Кент. – М.: АСТ – Апрель, 2004. – 334 с.
3. Концевой, М. П. Основы создания Web-страниц: Учеб.-метод. Указания / Сост. М. П. Концевой, И. В. Секержицкий. – Брест: БрГУ имени А.С. Пушкина, – 2002. – 32 с.
4. Мэтьюс, М. FrontPage 2000 для пользователя / М. Мэтьюс, Э. Полсен. – Киев: Изд-во «Ирина», – 2000. – 336 с.
5. Норенков, И. П. Телекоммуникационные технологии и сети / И. П. Норенков, В. А. Трудоношин. – М.: Изд-во МГТУ им. Баумана, 2000. – 248 с.
6. Олифер, В. Г. Компьютерные сети. Принципы, Технологии, Протоколы / В. Г. Олифер, Н. А. Олифер. – СПб.: Питер, 2001. – 672с.
7. Петюшкин, А. HTML. Экспресс-курс / А. Петюшкин. – СПб.: БХВ-Петербург, 2003. – 325 с.
8. Райтман, М. Adobe Dreamweaver CS6. Официальный учебный курс / М. Райтман. - М.: Эскмо, 2013. - 496 с.
9. Сейш, Т. Дизайн и архитектура современного Web-сайта. Опыт профессионалов / Т. Сейш, Г. Мак-Клейн: Пер. с англ. – М.: Издательский дом «Вильяме», 2001. – 304 с.



Кафедра
АГумМ

Начало

Содержание



Страница 186 из 187

Назад

На весь экран

Закрыть

10. Скляр, Дж. Актуальные принципы Web-дизайна / Дж. Скляр; [пер. с англ., ред. А. А.Борисенко]. – М.: Эксмо, 2007. – 464 с.
11. Чиртик, А. А. HTML / А. А. Чиртик. – 2-е изд. – СПб.: Питер, 2008. – 256 с.



*Кафедра
АГyMM*

Начало

Содержание



Страница 187 из 187

Назад

На весь экран

Закрyть