

06.34

Муниципальное общеобразовательное учреждение
«Средняя общеобразовательная школа №1 п. Пангоды»

Принята на заседании
Педагогического совета
Протокол № 9 от «28» мая 2021 года

УТВЕРЖДАЮ
Директор МОУ «Средняя
общеобразовательная школа №1 п. Пангоды»
Е.Р. Тимяшева /
Приказ №143/1 от «31» мая 2021 г.



Дополнительная общеобразовательная общеразвивающая программа
Технической направленности
Объединение «Вебдизайн и разработка» (программа JuniorSkills)
Уровень: Средний
Возраст обучающихся: 14 – 18 лет (9-11 классы)
Срок реализации: 1 год

«

Автор – составитель:
Зайцев Никита Михайлович,
учитель химии

п. Пангоды

Пояснительная записка

В современном развивающемся мире все большее значение для всех сфер социальной жизни приобретает глобальная сеть Интернет, особенно ее часть – Всемирная паутина. Поддержка функционирования паутины требует большого числа различных специалистов. Но основными специалистами, осуществляющими наполнение содержанием Всемирной паутины, являются люди, владеющие технологиями веб-дизайна. Такие специалисты востребованы на рынке труда, так как все большее число мелких и средних организаций желают иметь свой корпоративный Web -сайт. Огромное число пользователей Всемирной паутины тоже со временем хотят расширить свой кругозор и иметь не только свой блог или аккаунт в социальной сети, но и внедрить в него готовые виджеты, а особенно html-виджеты, которые требуют начальных знаний Web -дизайна.

Направленность программы

Курс включает в себя практическое освоение техники создания веб-страниц, тематических сайтов. Курс служит средством внутривузовской специализации в области новых информационных технологий, что способствует созданию дополнительных условий для построения индивидуальных образовательных траекторий учащихся, дает возможность профессиональных проб в области сайтостроения.

Умение представлять информацию в виде, удобном для восприятия и использования другими людьми, – одно из условий образовательной компетентности учащихся. Web-сайт – наиболее популярное и доступное для учащихся средство представления текстовой, графической и иной информации в сети Интернет.

Осознание и присвоение учащимися достигаемых результатов происходят с помощью рефлексивных заданий. Такой подход гарантирует повышенную мотивацию и результативность обучения.

Цель программы:

Научить обучающихся продуктивно действовать в информационном Интернет-пространстве для реализации своих коммуникативных, технических и эвристических способностей в ходе проектирования и конструирования сайтов.

Задачи программы

ОБУЧАЮЩИЕ (развитие познавательного интереса к чему-либо, включение в познавательную деятельность, приобретение определенных знаний, умений, навыков, развитие мотивации к определенному виду деятельности и т.п.):

- сформировать у учащихся целостное представление об информационной картине мира средствами сети Интернет, научить способам представления информации в сети Интернет;

- познакомить с видами веб-сайтов, их функциональными, структурными и технологическими особенностями;
- сформировать навыки элементарного проектирования, конструирования веб-сайта;
- дать базовые навыки программирования на языках HTML, CSS, JavaScript;
- реализовать коммуникативные, технические и эвристические способности учащихся в ходе проектирования и конструирования сайтов;
- сформировать отношение к компьютеру как инструменту для творчества, созидания, реализации своих потребностей;
- сформировать навыки элементарного проектирования, конструирования, размещения и сопровождения мультимедийного проекта.

РАЗВИВАЮЩИЕ (связаны с творчеством ребенка, формированием потребностей в самопознании и саморазвитии):

- развивать пространственное, образное мышление учащегося, являющегося важнейшей частью его интеллектуального развития в целом;
- развивать творческий подход к делу и поиску нестандартных решений в процессе создания Web-сайта;
- развивать эстетический вкус и дизайнерское мышление;
- развивать творческие авторские начала через создание самостоятельных проектов, участие в конкурсах и олимпиадах;
- способствовать обогащению жизненного опыта и формированию системы ценностей;
- повышать грамотность учащихся в области Интернет-технологий.

ВОСПИТАТЕЛЬНЫЕ (формирование у обучающихся социальной активности, гражданской позиции, культуры общения и поведения в социуме, навыков здорового образа жизни и т.п.)

- воспитать умение эффективно работать в команде и индивидуально, над решением нестандартных задач.

С самого начала изучения данного курса для учащегося создается среда, позволяющая наиболее полно проявить свои творческие способности.

Учебно – тематический план.

№ п/п	Название раздела, темы	Количество часов			Формирование контроля
		Всего	Теор.	Прак.	
Модуль 1: Основные элементы HTML кода					
1.	Вводное занятие. Техника безопасности на учебных занятиях	2	2	0	Вводный контроль, диагностические задания
2.	Введение в компьютерную графику	10	6	4	Текущий контроль

3.	HTML – язык гипертекстовой разметки	14	4	10	Промежуточный контроль, самостоятельная работа
4.	Создание интерактивных элементов	14	6	8	Промежуточный контроль, самостоятельная работа
5.	Основные элементы оформления HTML-страниц	14	4	10	Промежуточный контроль, самостоятельная работа
Модуль 2: Разработка вида страницы					
6.	Работа со шрифтами.	12	4	8	Промежуточный контроль, самостоятельная работа
7.	Помещение Web-графики на Web-страницу.	12	4	8	Промежуточный контроль, самостоятельная работа
8.	Обработка изображений с помощью специальных программ.	12	2	10	Промежуточный контроль, самостоятельная работа
9.	Оптимизация кода HTML для Web.	20	6	14	Промежуточный контроль, самостоятельная работа
10.	Анимация, звук и видео.	30	4	26	Промежуточный контроль, самостоятельная работа
11.	Встраивание видео на Web-страницу.	4	2	2	Промежуточный контроль, самостоятельная работа
ИТОГО		144	36	108	

Календарный учебный график.

№ п/п	Месяц	Число	Время проведения занятия	Форма занятия	Количество часов	Тема занятия	Место проведения	Форма контроля
Модуль 1: Основные элементы HTML кода (54 часа)								
1.	сентябрь	05	15.00-15.45 15.55-16.40	лекция	2	Вводное занятие. Техника безопасности на учебных занятиях	кабинет 203	собеседование
2.	сентябрь	06	15.00-15.45 15.55-16.40	лекция	2	Введение в компьютерную графику	кабинет 203	собеседование
3.	сентябрь	12	15.00-15.45 15.55-16.40		2		кабинет 203	собеседование
4.	сентябрь	13	15.00-15.45 15.55-16.40		2		кабинет 203	самоконтроль, взаимопроверка, мини-проекты
5.	сентябрь	19	15.00-15.45 15.55-16.40		2			
6.	сентябрь	20	15.00-15.45 15.55-16.40	практическая работа	2	HTML – язык гипертекстовой разметки	кабинет 203	зачёт
7.	сентябрь	26	15.00-15.45 15.55-16.40	лекция	2			
8.	сентябрь	27	15.00-15.45 15.55-16.40	практическая работа	2		кабинет 203	собеседование
9.	октябрь	03	15.00-15.45 15.55-16.40		2		кабинет 203	самоконтроль, взаимопроверка, мини-проекты
10.	октябрь	04	15.00-15.45 15.55-16.40		2			
11.	октябрь	10	15.00-15.45 15.55-16.40		2			
12.	октябрь	11	15.00-15.45 15.55-16.40		2			
13.	октябрь	17	15.00-15.45 15.55-16.40		2		кабинет 203	зачёт
14.	октябрь	18	15.00-15.45 15.55-16.40	лекция	2	Создание интерактивных элементов	кабинет 203	собеседование
15.	октябрь	24	15.00-15.45 15.55-16.40		2			

16.	октябрь	25	15.00-15.45 15.55-16.40		2		кабинет 203	самоконтроль, взаимопроверка, мини-проекты	
17.	октябрь	31	15.00-15.45 15.55-16.40	практическая работа	2				
18.	ноябрь	01	15.00-15.45 15.55-16.40		2				
19.	ноябрь	07	15.00-15.45 15.55-16.40		2				
20.	ноябрь	08	15.00-15.45 15.55-16.40		2				кабинет 203
21.	ноябрь	14	15.00-15.45 15.55-16.40	лекция	2	Основные элементы оформления HTML-страниц	кабинет 203		
22.	ноябрь	15	15.00-15.45 15.55-16.40		2				
23.	ноябрь	21	15.00-15.45 15.55-16.40	практическая работа	2		кабинет 203		зачёт
24.	ноябрь	22	15.00-15.45 15.55-16.40		2				
25.	ноябрь	28	15.00-15.45 15.55-16.40		2				
26.	ноябрь	29	15.00-15.45 15.55-16.40		2				
27.	декабрь	05	15.00-15.45 15.55-16.40		2				
Модуль 2: Разработка вида страницы (90 часов)									
28.	декабрь	06	15.00-15.45 15.55-16.40	лекция	2	Работа со шрифтами.	кабинет 203		
29.	декабрь	12	15.00-15.45 15.55-16.40		2				
30.	декабрь	13	15.00-15.45 15.55-16.40	практическая работа	2		кабинет 203		зачёт
31.	декабрь	19	15.00-15.45 15.55-16.40		2				
32.	декабрь	20	15.00-15.45 15.55-16.40		2				
33.	декабрь	26	15.00-15.45		2				

			15.55-16.40					
34.	декабрь	27	15.00-15.45 15.55-16.40	практическая работа	2	Помещение Web-графики на Web- страницу	кабинет 203	
35.	январь	16	15.00-15.45 15.55-16.40		2			
36.	январь	17	15.00-15.45 15.55-16.40		2			
37.	январь	23	15.00-15.45 15.55-16.40		2			
38.	январь	24	15.00-15.45 15.55-16.40		2			
39.	январь	30	15.00-15.45 15.55-16.40		2		кабинет 203	зачёт
40.	январь	31	15.00-15.45 15.55-16.40	лекция	2	Обработка изображений с помощью специальных программ.	кабинет 203	
41.	февраль	06	15.00-15.45 15.55-16.40	практическая работа	2		кабинет 203	зачёт
42.	февраль	07	15.00-15.45 15.55-16.40		2			
43.	февраль	13	15.00-15.45 15.55-16.40		2			
44.	февраль	14	15.00-15.45 15.55-16.40		2			
45.	февраль	20	15.00-15.45 15.55-16.40		2		кабинет 203	
46.	февраль	21	15.00-15.45 15.55-16.40	лекция	2	Оптимизация кода HTML для Web.	кабинет 203	
47.	февраль	27	15.00-15.45 15.55-16.40		2			
48.	февраль	28	15.00-15.45 15.55-16.40		2			
49.	март	06	15.00-15.45 15.55-16.40	практическая работа	2		кабинет 203	Зачёт
50.	март	07	15.00-15.45 15.55-16.40		2			
51.	март	13	15.00-15.45		2			

			15.55-16.40					
52.	март	14	15.00-15.45 15.55-16.40		2			
53.	март	20	15.00-15.45 15.55-16.40		2			
54.	март	21	15.00-15.45 15.55-16.40		2			
55.	март	27	15.00-15.45 15.55-16.40		2		кабинет 203	
56.	март	28	15.00-15.45 15.55-16.40	лекция	2	Анимация, звук и видео.	кабинет 203	зачёт
57.	апрель	03	15.00-15.45 15.55-16.40		2			
58.	апрель	04	15.00-15.45 15.55-16.40	практическая работа	2		кабинет 203	
59.	апрель	10	15.00-15.45 15.55-16.40		2			
60.	апрель	11	15.00-15.45 15.55-16.40		2			
61.	апрель	17	15.00-15.45 15.55-16.40		2			
62.	апрель	18	15.00-15.45 15.55-16.40		2			
63.	апрель	24	15.00-15.45 15.55-16.40		2			
64.	апрель	25	15.00-15.45 15.55-16.40		2			
65.	май	08	15.00-15.45 15.55-16.40		2			
66.	май	09	15.00-15.45 15.55-16.40		2			
67.	май	15	15.00-15.45 15.55-16.40		2			
68.	май	16	15.00-15.45 15.55-16.40		2			
69.	май	22	15.00-15.45		2			

			15.55-16.40					
70.	май	23	15.00-15.45 15.55-16.40		2		кабинет 203	
71.	май	29	15.00-15.45 15.55-16.40	лекция	2	Встраивание видео на Web- страницу.	кабинет 203	зачёт
72.	май	30	15.00-15.45 15.55-16.40	практическая работа	2	Встраивание аудио на Web- страницу.	кабинет 203	

Содержание курса.

Вводное занятие. Техника безопасности на учебных занятиях – 2 часа.

Цель и задачи объединения. Режим работы. План занятий.

Организация рабочего места. Правильное положение рук и туловища во время работы. Правила техники безопасности.

ТЕМА 1. Введение в компьютерную графику

Теоретические сведения.

ТЕМА 2. HTML – язык гипертекстовой разметки

Теоретические сведения.

ТЕМА 3. Создание интерактивных элементов

Список интерактивных элементов появившихся в HTML 5:

- 1) menu.
- 2) command.
- 3) datagrid.
- 4) details.

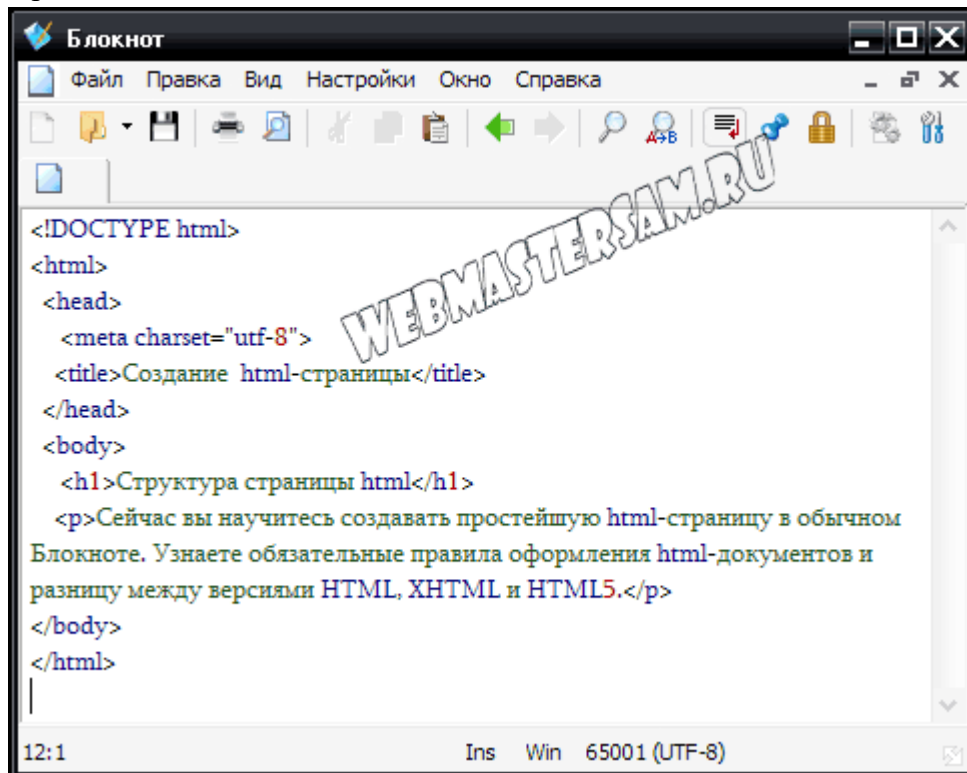
Элемент menu в HTML 5 является достаточно любопытным, в том плане, что он был ещё в более старых версиях [языка HTML](#), но потом был признан устаревшим, и вот в HTML 5 он неожиданно вернулся вновь.

Элемент menu содержит элементы command (подобно списку и его элементам), каждый из которых способен запускать определённое действие.

```
<html>
<head>
</head>
<body>
<menu>
  <command onclick="alert('Первая команда')" label="Первая команда"/>
  <command onclick="alert('Вторая команда')" label="Вторая команда"/>
</menu>
</body>
</html>
```

ТЕМА 4. Оформление html-страницы

Сейчас вы научитесь создавать простенькую html-страницу в обычном текстовом редакторе Блокнот. Узнаете о структуре и обязательных правилах оформления html-документов, о разнице между версиями HTML, XHTML и HTML5. Наберите в Блокноте следующие несколько строк:



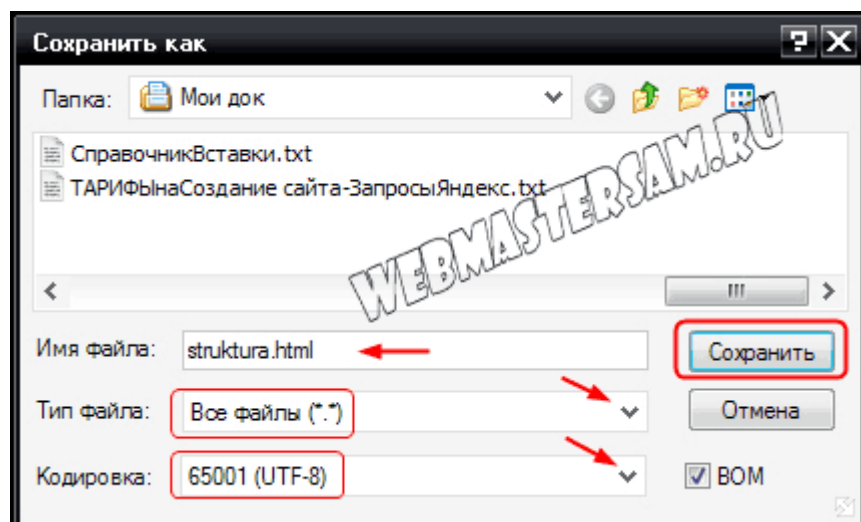
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Создание html-страницы</title>
  </head>
  <body>
    <h1>Структура страницы html</h1>
    <p>Сейчас вы научитесь создавать простейшую html-страницу в обычном
Блокноте. Узнаете обязательные правила оформления html-документов и
разницу между версиями HTML, XHTML и HTML5.</p>
  </body>
</html>
```

Скриншот наглядно показывает **структуру страницы html**, которая должна соответствовать следующим требованиям: в первой строке - указан тип документа. Далее следует тег `<html>`, в который вложены теги `<head>` и `<body>`. В теге `<head>` располагаются служебные теги: *link*, *meta*, *base*, *script* - они не отображаются на странице и тег [title](#) - название веб-страницы, которое видно в заголовке вкладки браузера. В теге `<body>` расположены видимые элементы разметки (у нас - заголовок и абзац).

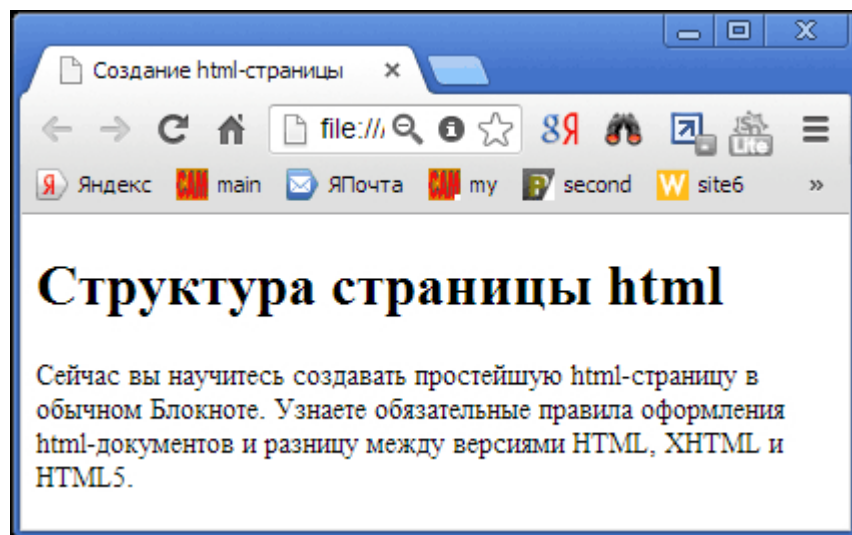
Все вложенные элементы следует закрывать в порядке, обратном их открытию. Некоторые **теги** не имеют *закрывающего тега*, некоторые содержат **атрибуты** с определенными **значениями**, пример для обоих случаев: тег *meta* в строке 4, у него атрибут *charset*, а его значение *utf-8* и нет закрывающего тега.

Для того чтобы создать html-страницу, просто сохраним наш документ в формате .html, задав ему имя, у меня - *struktura*. Обратите внимание,

что нужно изменить *Тип файла* и выбрать *Кодировка*, у меня выбрана - универсальная UTF-8:

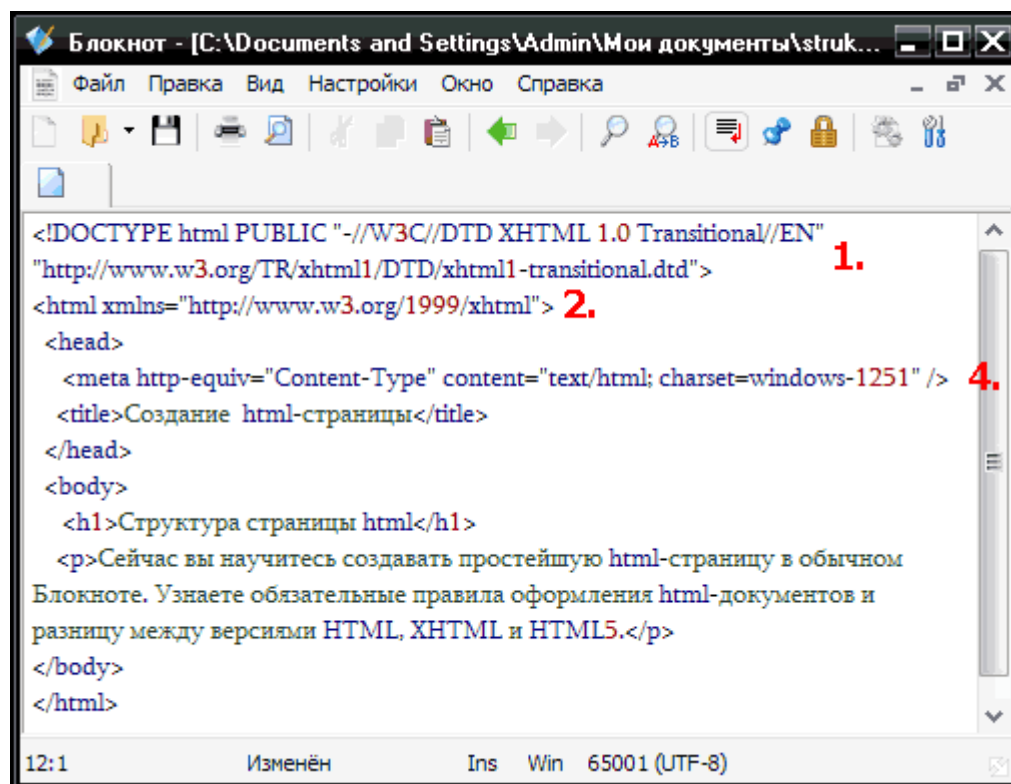


Чтобы просмотреть нашу html-страницу - откроем ее в браузере:



Наш html-документ соответствует версии html5. Эта версия, видимо, будет одобрена в 2014, однако большинство элементов html5 читаются основными браузерами, в т.ч. Internet Explorer, с версии 9. Вот тот же документ, только в русской кодировке windows-1251. Для русскоязычных сайтов используется кодировка windows-1251 или универсальная utf-8. Документ соответствует распространенной версии XHTML 1.0

Transitional:



```
Блокнот - [C:\Documents and Settings\Admin\Мои документы\struk...
Файл Правка Вид Настройки Окно Справка
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"> 1.
<html xmlns="http://www.w3.org/1999/xhtml"> 2.
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1251" /> 4.
  <title>Создание html-страницы</title>
</head>
<body>
  <h1>Структура страницы html</h1>
  <p>Сейчас вы научитесь создавать простейшую html-страницу в обычном
Блокноте. Узнаете обязательные правила оформления html-документов и
разницу между версиями HTML, XHTML и HTML5.</p>
</body>
</html>
```

Обратите внимание, что в коде страницы изменилась первая строка, содержащая тег `<!DOCTYPE>`, объявляющий тип документа. Смотрите о правильном написании [тега !DOCTYPE](#) для всех версий в справочнике. К тегу `<html>` в строке 2 добавлен обязательный для xhtml-документов атрибут *xmlns*. Изменение в строке 4: это [тег мета](#), задающий кодировку документа.

Тема 5. Работа со шрифтами.

Рассмотрим несколько наиболее популярных видов шрифтов.

Жирный шрифт (bold)

Для управления плотностью шрифта применяются теги:

`текст`

`текст`

Отличие тэга `` от тэга `` в том, что тэг `` указывает браузеру выводить текст жирным шрифтом, а тэг `` указывает, что текст надо выделить. Как браузер будет выделять текст, жирным шрифтом, курсивом или жирным курсивом - зависит от конкретного браузера.

Курсив (italic)

Курсивный шрифт выводится с помощью тэгов:

`<i>текст</i>`

`текст`

Пример: Курсив

Подчеркнутый шрифт (underline)

Выводится с помощью тэга:

`<u>текст</u>`

Пример: Подчеркнутый шрифт

Надстрочный индекс (Superscript)

Выводится с помощью тэга:

`^{текст}`

Пример: основной текст ^{надстрочный индекс}

$2^3=8$

Подстрочный индекс (Subscript)

Выводится с помощью тэга:

_{текст}

Пример: основной текст _{подстрочный индекс}

C₂H₅OH

Имитация стиля печатной машинки (Teletype)

<tt>текст</tt>

Пример: Teletype

Заголовки

Выводятся с помощью тэгов:

<h1>Самый большой заголовок</h1>

...

<h6>Самый маленький заголовок</h6>

Пример:

H1

H2

H3

H4

H5

H6

Блок с отступом

Выводится с помощью тега:

<BLOCKQUOTE>текст</BLOCKQUOTE>

Пример:

основной текст

блок текста с отступом

Тема 6. Добавление изображений на веб-страницу

Если на веб-странице не будет ничего, кроме текстовой информации, она, возможно, и покажется кому-то интересной, благодаря представленным на ней сведениям, однако вряд ли у кого то повернется язык назвать ее привлекательной. Добавлять изображения на веб-страницу очень легко. Для добавления же графики, которая сможет придать странице профессиональный вид, требуются некоторые знания. Чему, собственно и посвящен сегодняшний урок.

Двумя наиболее часто применяемыми в Интернете графическими форматами являются GIF и JPEG. Формат, разработанный группой JPEG (Joint Photographic Experts Group — Объединенная группа экспертов по машинной обработке фотографических изображений) и получивший ее название, используется, как правило, для сохранения изображений с плавными цветовыми переходами, таких, например, как фотоснимки.

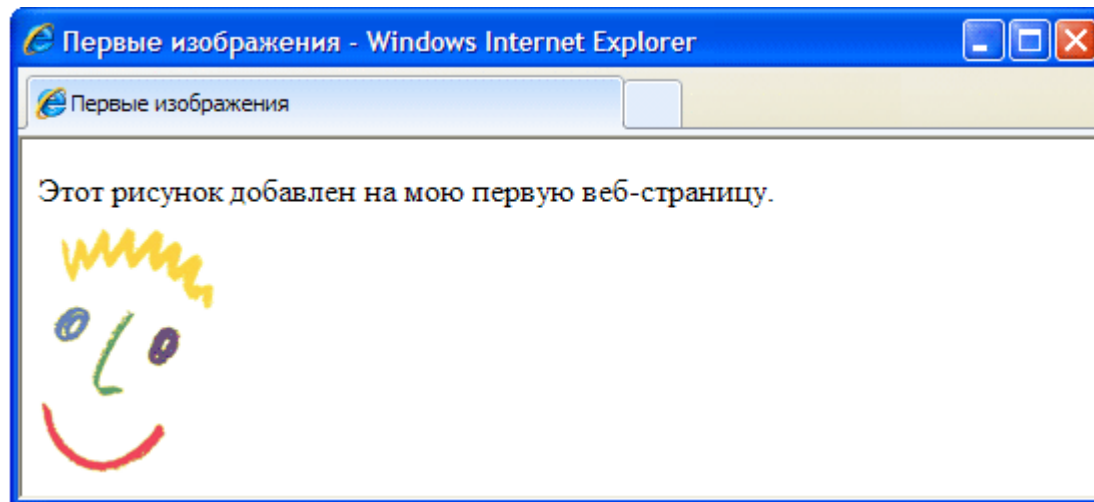
Практически все другие графические элементы сохраняются в формате GIF (Graphics Interchange Format) — формат обмена графическими данными. В настоящее время появился еще один новый графический формат, приобретающий все большую популярность: PNG (Portable Network Graphics — переносимая сетевая графика). Ожидается, что со временем он заменит собой формат GIF. Однако не спешите пересохранять все свои графические файлы в этом формате, пока еще он поддерживается далеко не всеми браузерами.

Все изображения добавляются на веб-страницу с помощью одного и того же тега, называемого тегом источника изображения ``. Наверное, сейчас уже вы сами можете определить, что данная запись состоит из собственно тега ``, его атрибута (src) и значения данного атрибута (место расположения). Однако, поскольку применение атрибута src является обязательным, удобнее говорить об этой записи как об одном общем теге. Вероятно, вы обратили также внимание, что для тега источника изображения не предусмотрен соответствующий закрывающий тег. Это отдельный самодостаточный тег, и потому не забывайте добавлять в его конце закрывающую косую черту: ``.

`<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"`

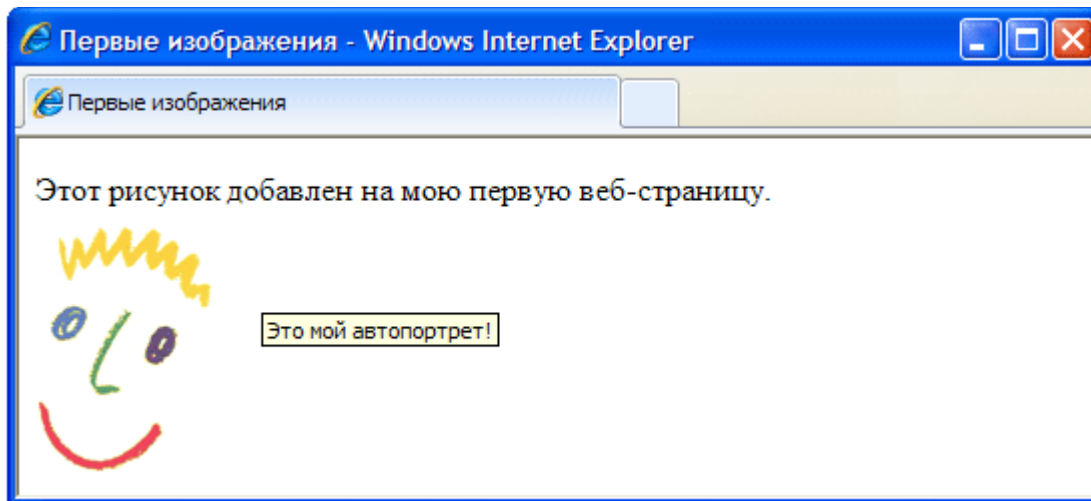

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ru" lang="ru">
<head>
<title>Первые изображения</title>
<style type="text/css">
</style>
</head>
<body>
<p>Этот рисунок добавлен на мою первую веб-страницу.</p>

</body>
</html>
```



Добавление альтернативных надписей

Путешествуя по Интернету, вы, наверное, не раз замечали различные надписи, отображаемые при наведении указателя мыши на какой-нибудь графический элемент веб-страницы. Эти надписи обычно сообщают какую-то дополнительную информацию о самом изображении или о той области страницы, которую оно занимает.



На примере представленного далее HTML кода показано, как атрибут alt добавляется внутрь тега ``. Подобно атрибуту `src`, атрибут `alt` сообщает веб-браузеру некоторую дополнительную информацию об изображении, и его также всегда можно использовать вместе с тегом ``.

```

```

Атрибут `alt` определяет для добавленного на веб-страницу графического элемента альтернативный текст. Этот текст называется альтернативным, поскольку может отображаться на экране как альтернатива самому изображению. Атрибут `alt` имеет еще одно очень важное назначение. Многие посетители интернета, использующие каналы доступа с низкой скоростью передачи данных, могут давать указание своим браузерам не загружать и не отображать графическую информацию. Это позволяет им ускорить загрузку веб-страниц на свои компьютеры.

Кроме того, помните, что не все браузеры способны отображать в своих окнах графику. Например, браузер Lynx вообще не поддерживает такой возможности. Таким образом, атрибут `alt` позволяет веб-дизайнеру быть уверенным в том, что, если посетитель его веб-страницы не увидит на своем экране изображения, он по крайней мере сможет ознакомиться с добавленной к этому изображению текстовой информацией.

Хотя атрибут alt можно определять для каждого тега , следите за тем, чтобы не назначить некоторым графическим элементам несоответствующие им текстовые сообщения. Например, не имеет смысла добавлять альтернативные текстовые надписи к различным элементам внешнего оформления веб-страницы. Чтобы избежать подобных ошибок, можете присваивать атрибуту alt таких элементов пустое значение (alt=» »). Если не задавать значения никаких других атрибутов, браузер воспроизведет изображение в его исходном размере, выровняв верхний край изображения с верхним краем расположенной рядом текстовой строки. Вы можете изменить оба эти параметра, воспользовавшись тегами стилевых таблиц.

Атрибуты изображения

Для тега предусмотрены атрибуты, позволяющие изменять размер изображения. Некоторые из этих атрибутов перечислены ниже.

height — Указывается в пикселях или процентах — Определяет высоту изображения

width — Указывается в пикселях или процентах — Определяет ширину изображения.

Тема 7. Обработка изображений с помощью специальных программ.

Для простой обработки качественных изображений бывает достаточно программ для просмотра изображений, часто распространяемых в комплекте с цифровыми фотоаппаратами и сканерами.

Если необходима правка, изображения в цифровом виде редактируются в специальных программах – графических редакторах.

Разные графические редакторы имеют свои особенности и предназначаются для разных целей, правильное их использование дает возможность выполнить всевозможные задачи, появляющиеся при редактировании изображений. Цифровая обработка помогает в значительной мере исправить технические недочеты, возникшие в процессе фотосъемки.

Оригинал редактируемого изображения обязательно сохраняется. Обработываемый файл сохраняется под новым именем. Во время редактирования желательно сохранять копии после каждого значительного изменения.

Инструменты для технического и структурного редактирования изображений

Редактирование изображений осуществляется с помощью разных графических редакторов, большинство которых имеет основные функции и инструменты для самых распространенных операций редактирования. Как правило, графические программы предоставляют следующие возможности.

- Выделение фрагмента изображения для работы с ним. Выделяется область изображения, которую необходимо отредактировать, после чего коррекция происходит только в этой области, не затрагивая остальное изображение. Во многих программах выделить фрагмент изображения можно с помощью контура, также для выделения применяется *маскирование*^[1].

- Как правило, в графических редакторах выделение может быть временным, в более сложных программах часть изображения может представлять из себя другой слой или объект, что дает возможность строить изображение из редактируемых по отдельности фрагментов, которые накладываются друг на друга.

Графические программы предоставляют выбор функций обработки, которые применяются к одному или нескольким изображениям, выделенной области или объекту. Функции делятся на две основные группы – изменение формы и изменение цвета.

Выделенную область изображения в большинстве программ можно двигать, поворачивать, масштабировать, деформировать, дорисовывать, обрезать.

Также к целому изображению или выделенной области применяется цветовая коррекция – изменение цвета, тона, насыщенности, яркости, контраста и т.п.

- Большинство графических редакторов имеют набор инструментов для рисования, создания фигур, заполнения областей цветом, градиентом^[2] или готовым узором.

применяются инструменты, аналогичные текстовым редакторам.

- При ретушировании используются специальные инструменты, копирующие содержимое одной области в другую.

- Многие программы содержат функции подавления цифрового шума, а также другие специальные эффекты и фильтры, применяемые с разными целями.

- Для редактирования размера изображения используются функции изменения размера, изменения размера холста^[3] и кадрирования (обрезки).

Отдельно надо отметить используемые для настройки инструментов и функций меню параметров.

- [1] Маскирование – наложение маски на участки изображения, которые не должны подвергаться изменениям. Свободные от маски области изображения при этом изменяются.
- [2] Градиент в компьютерной графике – вид заливки, создающей плавный переход из одного цвета в другой. Для получения градиентной заливки задается цвет и прозрачность начальной и конечной точек, цвет и прозрачность остальных точек рассчитываются относительно них по определенным математическим алгоритмам. Градиенты используются как в растровой, так и в векторной графике. Основные виды градиентов: линейный – ключевые точки располагаются на одной прямой; круговой – относительно первой ключевой точки строятся концентрические окружности, цвет которых зависит от расположения остальных ключевых точек; угловой (конусный) – ключевые точки располагаются по кругу; отраженный – похож на линейный, состоит из двух частей (линейного градиента и его воображаемого отражения в зеркале); ромбовидный (алмазный) – напоминает собой алмаз.

Оптимизация метатегов

1. **Title** – заголовок страницы должен обязательно содержать главное ключевое слово. Именно этот фактор учитывается при определении релевантности сайта поисковому запросу. И если заголовок не является ответом на вопрос пользователя, маловероятно, что именно этот ресурс будет выбран для посещения. Важно соблюдать размер заголовка. Если название страницы будет длиннее примерно 65 символов, поисковик просто обрезает предложение и тогда пользователи увидят бессмысленный текст. Заголовки и подзаголовки могут включать дополнительные ключевые слова для расширения возможностей поиска.
2. **Metater description** (описание) используется Google в качестве сниппета (кстати в недавнем прошлом google запустил структурированные сниппеты) – анонса сайта на SERP. Яндекс выбирает сниппет из любого места текста по своим принципам. Если сниппет окажется неудовлетворительным, можно просто отредактировать абзац на месте, где он находится.
3. **Metater keywords** (ключевые слова) не играют особой роли. Но не мешает их все же тщательно прописать. В тексте количество ключевых слов сегодня не требует определенной численности. Гораздо важнее богатство словаря и использование специальной терминологии при развитии темы.
4. Текст форматируется при помощи каскадных таблиц стилей и сами CSS убираются со страницы в отдельный каталог. Это тоже приводит к снижению нагрузки на браузер и улучшению работы сайта. Основной текст нужно помещать как можно выше на странице, а все служебные теги убирать вниз.
5. Кроме всего вышеперечисленного, нужно использовать веб-совместимые шрифты. Иначе могут возникать сложности с отображением текста в браузерах.

Очистка кода от мусора (Оптимизация html кода)

Наличие излишних, ошибочных тегов, даже двойных пробелов могут сильно повысить вес страницы и замедлить работу сайта.

Валидация и стандартизация

Чтобы браузеры адекватно могли работать с сайтом, код должен соответствовать современному стандарту, что указывается в верхней строке HTML. Далее идет валидность кода. Этот параметр можно бесплатно проверить на сайте консорциума разработчиков WWW.

Оптимизация графики

Графика в хорошем качестве очень много весит и сильно перегружает ресурс. Все картинки хранятся не на страницах, а в отдельной директории и в сжатом состоянии. В теги Alt и Description (всплывающее описание картинки), в название файла полезно добавить ключевые слова. Это повышает видимость сайта в поиске по картинкам и облегчает поисковым роботам задачу индексации.

Оптимизация скриптов

По современным стандартам кодирования скрипты не должны располагаться на самой веб-странице, а, так же как и графические элементы, сжимаются и убираются в отдельный каталог. Поскольку эти полезные маленькие программы имеют большой вес.

Удаление битых ссылок

Поисковики отрицательно относятся к мертвым ссылкам, которые никуда не ведут и по которым нет живого трафика. Все нерабочие ссылки нужно отыскать и безжалостно удалить со страниц. Полезно заодно проверить и качество доноров. Ссылки с линкопомоек и незаконных проектов сильно вредят трасту сайта.

Тема 9.

Графика, анимация, видео и звук

- ☐ Размещение графических изображений
- ☐ Как выбрать формат графического файла
- ☐ Прозрачные изображения
- ☐ Подготовка чересстрочных изображений
- ☐ Анимация
- ☐ Видеофрагменты в документах HTML
- ☐ Звук в документах HTML

В этой главе мы расскажем о том, как оживить страницы вашего сервера WWW, разместив на них графические изображения, видеофрагменты, анимацию, а также звук, - словом мультимедийную информацию.

Для подготовки мультимедийной информации вам могут потребоваться специальные приложения, такие как графические редакторы, средства ввода и обработки звука. В качестве примера мы можем назвать такие графические редакторы как Corel Draw и Adobe Photoshop.

Не исключено, что вам придется приобрести видеокамеру, адаптер для ввода видеосигнала от видеомэгнитофона или видеокамеры, а также редактор файлов с видеофрагментами. В 15 томе “Библиотеки системного программиста”, который называется “Мультимедиа для Windows”, мы рассмотрели основные средства и методики, пользуясь которыми вы сможете самостоятельно создавать небольшие видеофрагменты в формате файлов AVI.

Если вы не обладаете художественными талантами, необходимыми для создания графических изображений или видеофрагментов, можно их заказать у художника либо приобрести отдельно. В любом случае вам нужно позаботиться о соблюдении авторских прав на всю информацию, которую вы собираетесь опубликовать на вашем сервере WWW, а также на все приложения, которые вы установите как расширение сервера (программы CGI или расширения типа ISAPI).

Размещение графических изображений

Для размещения графических изображений внутри текста или таблиц вы можете воспользоваться оператором . Для этого оператора можно указывать ряд параметров, определяющих адрес файла с изображением, выравнивание текста, расположенного возле изображения и так далее. Ниже мы привели список параметров, допустимых для оператора в стандарте языка HTML версии 3.2.

Параметр	Описание
SRC	Адрес URL файла с графическим изображением
ALT	Текстовая строка, которая отображается вместо графического изображения в тех случаях, когда навигатор не может показывать графические изображения или когда такая возможность отключена
ALIGN	Выравнивание текста относительно графического изображения LEFT по левой границе; RIGHT по правой границе; TOP по верхней границе; MIDDLE по центру изображения; BOTTOM по нижней границе; TEXTTOP выравнивание по верхней границе относительно самых высоких символов в текстовой строке; ABSMIDDLE выравнивание середины текстовой строки относительно середины изображения; BASELINE выравнивание нижней рамки изображения относительно базовой линии текстовой строки; ABSBOTTOM выравнивание нижней границы изображения относительно нижней границы текущей строки
HEIGHT	Высота изображения в пикселах
WIDTH	Ширина изображения в пикселах
BORDER	Ширина рамки вокруг изображения в пикселах

	(используется только навигатором Netscape Navigator)
HSPACE	Ширина свободного пространства в пикселах, которое должно отделять изображение от текста по горизонтали
VSPACE	Ширина свободного пространства в пикселах, которое должно отделять изображение от текста по вертикали
USEMAP	Адрес URL файла, содержащего так называемую карту изображения, которая используется для сегментированной графики. Сегментированная графика будет рассмотрена нами позже в этой главе
ISMAP	Этот параметр указывает, что данное изображение является сегментированным

С помощью параметров HEIGHT и WIDTH вы можете выполнять масштабирование графических изображений. Значение этих параметров можно указывать в процентах от ширины окна просмотра.

Масштабирование позволяет подготовить графический файл относительно небольшого размера, который быстро передается через Internet, и занимает при этом значительную площадь в окне навигатора. Вы, однако, не можете масштабировать сегментированные графические изображения и фоновые изображения.

В листинге 4.1 мы привели пример документа HTML, в который вставлено два растровых графических изображения.

Листинг 4.1. Файл chap4\inline.htm

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<HTML>
  <HEAD>
    <TITLE>Размещение графических изображений</TITLE>
  </HEAD>
  <BODY BGCOLOR=#FFFFFF>
    <H2>Размещение графики в документах HTML</H2>
    <IMG SRC="cpanel.jpg">Это содержимое папки
Control Panel на моем компьютере
    <P>Вы можете включать графические
изображения <IMG SRC="clock.jpg">непосредственно в текст
```


</BODY>

</HTML>