# **Мировые Информационные ресурсы**

# Преподаватель: Выдрин Алексей Лонгинович

**Информационные ресурсы**.

**Сведения**  - это набор сигналов физических процессов воспринимаемых субъектом через органы его чувств. ( Субъектом может быть человек или машина, которая предназначена для восприятия сигналов.)

## **Данные** - это сведения, полученные путём измерения, наблюдения, логических или арифметических операций представленные в форме, пригодной для хранения, передачи и обработки. ( Пример данных – автомат, собирающий космические измерения, записывающий на носитель и эти данные в последствии будут использованы учёными.)

##

**Передача данных -** обмен данными любого характера между различными устройствами по каналам связи. ( Пример – работа факсов.)

**Обработка данных**  - последовательность операций, производимых над данными.

 **Данные различаются по:**

**1)*Формату данных*** *–* характеристика данных, способствующая оптимальному их использованию и определяющая структуру и способ их хранения, диапазон возможных значений и допустимые операции, которые можно выполнять над этими данными.

(Графический документ, электронный формат и т.д.)

**2)*Структуре данных*** – это организационная схема, в соответствии с которой данные

упорядочены с тем, чтобы их можно было максимально эффективно интерпретировать

или выполнять над ними различные операции.

**Информация** – это сведения, независимо от формы их представления, усваиваемые субъектом в форме знаний.

**Документ** – это материальный объект с зафиксированной на нем информацией, пред–

назначенный для передачи во времени и пространстве в целях хранения и общественного использования, содержащий реквизит.

**Реквизит документа** – обязательные характеристики, которые должен содержать документ для его однозначной идентификации.

**Документ в электронной форме** – это документ, представленный в форме набора состояний элементов вычислительной техники или иных средств обработки, хранения, передачи информации, допускающий преобразования в форму, пригодную для однозначного восприятия человека.

**Информационный продукт** – это документированная информация, подготовленная в соответствии с потребностями пользователей и представленная в форме товара.

 Основные особенности информационного продукта, отличающие информацию от других товаров:

1. Информация не исчезает при потреблении, а может быть использована многократно. Информационный продукт сохраняет содержащуюся в нем информацию, независимо от того, сколько раз она была использована.
2. Информационный продукт со временем подвергается моральному износу, заключающемся в потере актуальности и появлении более новых копий.
3. Разным потребителям дается возможность различных способов потребления информационного продукта. (Монитор, распечатка и т.д.) Эта особенность называется *адресность* *информации.*
4. Производство информации в отличие от производства материальных товаров требует значительных затрат по сравнению с затратами на тиражирование.

**Информационная система** – это организационно упорядоченная совокупность документов (информационных продуктов) и информационной инфраструктуры. (Примеры информационных систем - библиотеки, архивы, фильмотеки, базы данных, Internet.)

**Информационные ресурсы** – это отдельные документы, массивы документов, которые входят в состав информационных систем.

# **Информационные технологии**

**Информационная инфраструктура** – это информационные технологии в виде вычислительных комплексов, оргтехники, средств и каналов связи, управляемая и поддерживаемая в рабочем состоянии с помощью неких организационных мероприятий.

 Основные части информационной инфраструктуры:

1. Вычислительная техника.
2. Организационная техника (конторское оборудование).
3. Дисплейная техника (мониторы, телевизоры и т.д.)
4. Устройства хранения информации (накопители на жестких дисках, на магнитную ленту).
5. Печатное оборудование (принтер).
6. Средства связи (средства радио и теле вещания).
7. Системы передачи данных (коммутаторы, маршрутизаторы).
8. Каналы связи (оптоволокно).

# **Тенденции развития информационных технологий**

1. Возрастание роли информационного продукта.
2. Стандартизация в сфере информационных технологий.
3. Глобализация информационных технологий.
4. Ликвидация промежуточных звеньев.
5. Интеграция информационных технологий.

**1.Возрастание роли информационного продукта.**

 Потребность в обработке все возрастающих объемов информации, потребность в различных формах восприятия информации, а также потребность в актуальности и точности информации послужило развитию этой тенденции.

**2.Стандартизация в сфере информационных технологий**.

Эта тенденция раскрывает способность к взаимодействию между элементами информационных технологий различных производителей. То есть необходимо, чтобы старые устройства могли взаимодействовать с новыми.

**3.Глобализация информационных технологий.**

 Процесс глобализации обуславливает пять основных причин:

1. Различный уровень знаний в области информационных технологий.
2. Соотношение стоимости разработки отдельных элементов информационных технологий и эффективности их применения.
3. Правительственная поддержка.
4. Стандартизация.
5. Сравнительное достоинство сосуществующих и взаимозаменяемых технологий.

(В основном глобализации способствует первая причина).

**4.Ликвидация промежуточных звеньев.**

Эта тенденция способствует непосредственному взаимодействию источника и потребителя информации.

**5.Интеграция информационных технологий.**

Интеграция – это глобализация мирового масштаба. Это обеспечение бизнеса с помощью сторонних компаний и предприятий. (Например, одной компании дорого строить магазин, а две собрались вместе и построили.)

# **Компьютерные сети**

**Компьютерная сеть** – это группа компьютеров, объединенных с помощью каналов связи и обеспечивающих с совместное использование информационных и вычислительных ресурсов отдельных компьютеров.

**Вычислительные ресурсы** – это жесткие диски, оперативная память, процессорная мощность, принтеры, модемы, сканеры и другие устройства, установленные на отдельных компьютерах сети.

**Локальная сеть** – это компьютерная сеть в пределах одного здания, офиса или группы зданий.

**Сервер** – это компьютер, подсоединенный к компьютерной сети, работающий непрерывно во все время функционирования сети, имеющий мощные информационные и вычислительные ресурсы и обеспечивающий многочисленные обращения к этим ресурсам пользователей.

**Клиентский компьютер (клиент)** – это компьютер, подключенный к сети, основное назначение которого – обеспечить доступ пользователя к ресурсам сети.

**Глобальная сеть** – это компьютерная сеть мирового масштаба, включающая в свой состав локальные сети, используемые для совместного доступа к мировым информационным ресурсам.

**Узел глобальной сети** – это программно - аппаратный комплекс, обслуживаемый соответствующим персоналом и исполняющий роль регионального центра управления глобальной сетью.

 Функции узла глобальной сети:

1. Поддержка функционирования регионального сегмента сети.
2. Работы по изменению структуры сегмента сети (его увеличение или уменьшение).
3. Поддержка функционирования региональных ресурсов (то есть больших баз данных).

В состав комплекса узла могут входить, как серверы, так и клиентские компьютеры.

клиент

#

узел

клиент

узел

клиент

 - коммутируемые каналы связи.

 - выделенные каналы связи.

***Коммутируемый канал*** предоставляет временную связь к компьютерам сети посредством

телефонной сети общего назначения (через коммутатор АТС).

***Выделенный канал*** предоставляет постоянную связь к компьютерам сети посредством следующих видов связи:

* Выделенная медная пара (телефонная линия).
* Коаксиальный кабель.
* Витая пара.
* Оптоволокно.
* Радиоканал.
* Спутниковый канал.
* Воздушный оптический канал (инфракрасный канал).

**Сервис** – это программа, работающая на серверах, посредством которой обеспечивается доступ к информационным и вычислительным ресурсам.

# **Основные сервисы глобальной сети Internet**

### 1.Web сервис

**Web сервис** (www) – это технология доступа к информационному ресурсу посредством Web сайта. Это самый используемый сервис сети.

**Web сайт** – это набор связанных друг с другом документов. Для связи таких документов используются гиперссылки.

* Personal Web Server – это программа, с помощью которой мы можем регулировать работу Web сервиса. (В рамках локального компьютера).
* Internet Information Service – более полная версия предыдущей программы.

### 2.FTP сервис

Этот сервис обеспечивает удаленный доступ к файловой системе сервера.

Существует также понятие **FTP сайт** – это группа файлов, папок, находящихся в файловой системе сервера к которым обеспечивается удаленный доступ.

В составе FTP сервиса имеются средства для передачи файлов в больших объемах и возможности повторной передачи данных с того места, на котором прервалась первая передача.

### 3.Сервис электронной почты (E-mail)

Предназначен для пересылки информации конкретному пользователю глобальной сети (Internet). Каждый пользователь имеет **ящик электронной почты** – это папка на сервере,

где хранятся сообщения пользователя.

### 4.Сервис телеконференций

Данный сервис также называют сервер групп новостей (NEWS). Он позволяет участвовать всем пользователям сети в групповых дискуссиях, в которых обсуждаются различного рода вопросы.

**Группа новостей** – это место, где происходит дискуссия по какой-то определенной тематике. (Это папка на жестком диске сервера, в которой хранятся сообщения участников дискуссии).

# **Адресация в сети Internet**

Передача информации в глобальных сетях осуществляется (в основном) в пакетном режиме. Информация или файл разбиваются на пакеты и передаются независимо друг от друга, а затем снова собираются в целую информацию (файл).

 Оптоволоконный канал Спутниковый канал

пакеты узел маршрутизации Радиоканал

Каждый переданный пакет должен получить подтверждение по достижении им получателя. (То есть отправитель должен получить подтверждение о том, что пакет достиг получателя).

**Протокол** – это набор правил и команд, определяющий передачу информации в сетях (любых).

 ***Сетевой протокол*** определяет передачу на уровне пакетов (низкий уровень), а ***прикладной протокол*** определяет передачу информации на уровне сервисов (то есть информация более высокого уровня).

**Адресация глобальных сетей** – это система уникальных имен (адресов), позволяющая передавать информацию в определенном направлении.

 ***Сетевая адресация*** позволяет нам установить местонахождение компьютера в глобальной сети.

 ***Прикладная адресация*** – устанавливает местоположение единиц информации. (Примерами единиц информации являются Web сайт, FTP сайт, файл и т. д.).

 ***На сетевом уровне различают следующие виды адресов***:

**1.IP адрес** – уникальное имя компьютера в глобальной сети. (Например, 192.168.0.1 – четырехразрядный адрес. В каждом разряде может быть от одной до трех цифр (числа от 0 до 255).

Именно IP адрес используется для отправки и получения пакетов.

**2.Доменный адрес** – так же предназначен для определения местоположения компьютеров в сети.

Пример:

[www.kirov.ru](http://www.kirov.ru) - (состоит из букв, некоторые из которых складываются в слова). Указывает на компьютер с именем www, находящийся в России в Кирове.

**ru** – домен первого уровня (это группа компьютеров, входящих в состав российского сегмента сети Internet).

**kirov** – домен второго уровня (в данном случае указывает на компьютеры, входящие в состав кировского сегмента сети).

Последняя составляющая **www** – это имя компьютера в домене.

**Порт компьютера** – указывает местоположение конкретного сервиса среди других, работающих на одном компьютере.

Порт выглядит как обычный номер от 1 до 10000. № порта обязательно указывается при передаче и получении пакетов.

Сервер, содержащий информационный ресурс

Клиентский компьютер

 IP адрес отправителя.

 IP адрес сервера получателя.

 № порта (адрес сервиса).

***Прикладной уровень адресации***

***или адресация на уровне сервисов в глобальной сети***

Данная адресация используется для указания информационных ресурсов доступных пользователю посредством конкретного сервиса.

**URL адрес** - используется в WEB и FTP сервисах для указания местоположения WEB документов и файлов.

Пример URL адреса:

<http://www.site.ru:81/folder1/file2.htm>, где

**http** – это протокол, используемый для обращения к ресурсу.

**://** - синтаксис.

[**www.site.ru**](http://www.site.ru) - доменный адрес, указывает местоположение сервера, где находится ресурс.

**81** – номер порта – адрес, по которому находится Web сервис на данном сервере. Этот параметр не обязателен. По умолчанию используется порт 80.

 **/** - символизирует корневую папку сайта.

**folder1/** - путь к ресурсу, начиная от корневой папки.

**file2.htm** – имя файла, в котором хранится документ.

Имя файла, в котором хранится ресурс не обязательный параметр. По умолчанию используется имя, которое было указано в настройках Web сервера.

В самом упрощенном виде URL адрес выглядит так:

[www.site.ru](http://www.site.ru)

Автоматически подставляется http:// и :80/

**Сайт** – это группа документов, которые связаны между собой тематикой и гиперссылками.

Обращение к сайту выглядит так:

<http://www.site.ru/>

На одном сервере может находиться несколько сайтов.

|  |  |
| --- | --- |
| www.site.ru | f1 |
| www.site2.ru | f2 |
|  |  |

 http://www.site.ru/

 194.84.201.1

Папка f1

Папка f2

При обработке запроса, поступившего к Web сервису, анализируется доменный адрес и сопоставляется с таблицей размещения сайтов на сервере, после чего предоставляется доступ к запрошенному сайту. Таким образом, на сервере может находиться несколько сайтов, обслуживаемых одним Web сервисом.

Для того чтобы разные доменные адреса указывали на один сервер, они должны транслироваться в один и тот же IP адрес. Трансляция доменного адреса в IP адрес производится с помощью DNS сервиса.

FTP сервис обрабатывает URLадреса точно так же.

# **Адрес электронной почты**

Он указывает местоположение папки, где хранятся сообщения, пришедшие к конкретному пользователю (почтовый ящик).

Например, user@server.ru, где

user – имя папки (почтового ящика).

@- синтаксическая единица (коммерческая at, в переводе на; или просто собачка).

server.ru - адрес сервера, где хранится почтовый ящик.

IP 2

IP 1

Компьютер пользователя

 25

 user@server.ru

 110

 25

Отправитель

 Сообщение

Папка user

 Ответ

Номер порта, по которому принимает запросы почтовый сервер **25**.

SMTP сервис – почтовый сервис.

POP3 сервис – служит для обращения к почтовому ящику (обслуживает сообщения по **110** порту).

# **Адрес группы новостей**

Он указывает расположение папки, где хранятся сообщения по обсуждаемой в рамках этой группы теме.

Группа новостей рассредоточена по нескольким серверам глобальной сети. Отдельные папки, располагаемые на этих серверах и являющиеся основной частью группы новостей, постоянно синхронизируются, то есть поддерживается их одинаковое содержимое.

Сервер новостей News

Пользователь

alt.soft.windows

**alt** – корневая группа новостей. Используется для обозначения начала иерархии.

**soft** – основное обсуждение по теме.

**windows** – конкретно обсуждается эта проблема. (Обсуждение программного обеспечения, которое работает под управлением Windows).

# **Работа с Web сервисом.**

# **Использование программы Internet Explorer**

Программа Internet Explorer, как и любой другой браузер, предназначена для работы с Web сайтами. В качестве дополнительной функции обеспечивается работа с FTP сайтами.

 Основные функции программы:

1. Открытие HTML документов.
2. Сохранение HTML документов на локальных дисках.
3. Хранение избранных URL адресов и учет URL адресов, посещенных пользователем страниц (открытых HTML документов).
4. Обеспечение эффективности работы с Web сайтом.
5. Загрузка документов, отличных от HTML документов (то есть имеющих иной формат).
	1. **Открытие HTML документов**.

Основная функция Internet Explorer – открытие HTML документов.

 Способы открытия:

1. *Адресная строка*. В ней указывается адрес того документа, который необходимо открыть.
2. *Использование ссылок.*

**Гипрессылка** *–* это URL адрес того документа, который привязан к одному из элементов Web страницы (текстовому или графическому).

Ссылки бывают *внутренние* и *внешние,* по отношению к открытому Web документу.

 ***Внутренние*** – хранятся внутри открытого Web документа.

 ***Внешние*** – хранятся в папках Windows (избранное и в других).

1. Команда открытьв Internet Explorer.
2. Команда открыть в Windows.
3. Диалог загрузка файла.

**2.Сохранение документа.**

Команда Internet Explorer сохранить как.

 Существует 3 способа сохранения:

* *Web страница полностью* (когда сохраняется и текст и мультимедийные элементы).
* *Web страница только HTML* (сохраняется только текстовое содержимое без мультимедийных элементов).
* *Текстовый файл* (сохраняется текст, но теряется форматирование, то есть внешний вид текста).

**3.Учет URL адресов. Сохранение внешних ссылок**.

**1)***Использование папки избранное*.

Папка избранное является обычной папкой Windows, хранящейся на локальном диске, в которой ссылки хранятся в виде файлов (папки хранятся в виде ярлыков).Internet Explorer предоставляет доступ к этой папке через панель => избранное и команд добавить => избранное и упорядочить => избранное.

 **2)***Использование панели ссылок*.

Панель ссылок – это панель инструментов Internet Explorer, на которой отображается содержимое папки ссылки, которая в свою очередь, входит в состав папки избранное.

**3)***Назначение домашней страницы.*

Домашняя страница – это Web документ, открывающийся при запуске программы Internet Explorer, а также при щелчке по кнопке HOME на основной панели инструментов Internet Explorer. Назначение домашней страницы осуществляется или через свойства обозревателя, где явно указывается URL адрес домашней страницы, или с помощью перетаскивания ссылки на домашнюю страницу поверх кнопки домой на панели инструментов.

**4)***Журнал.*

Журнал – это автоматически заполненный Internet Explorer список ссылок на Web документы, когда-либо открывавшиеся пользователем.

Настройка журнала осуществляется командой сервис => обозреватель.

В журнале имеется возможность сортировки вида ссылок:

* По узлу (то есть по принадлежности к тому или иному сайту).
* По дате.
* По порядку посещения.
* По посещаемости, то есть по частоте посещения сайта (наиболее часто посещаемые сайты отображаются на первой странице журнала).

Папки хранятся в виде ярлыков.

**5)***Создание ярлыка Internet.*

Ярлык Internet – это файл, в котором хранится URL адрес. Этот файл располагается в какой-либо папке на жестком диске. Этот файл имеет расширение **.url**

**4.Обеспечение эффективности работы с Web сайтом.**

 **1)***Авто заполнение* – это автоматическое завершение набора текста, начатого пользователем в любом текстовом поле.

 **2)***Открытие нового окна Internet Explorer*.

Открытие производится с помощью клавишCNTRL+N или открыть New => окно. Это позволяет нам работать сразу с несколькими документами.

 **3)***Отключение мультимедийных элементов* (картинок, видео, аудио анимации).

Осуществляется посредством диалога со свойством обозревателя в пункте меню Сервис. (Сервис => свойства обозревателя => дополнительные).

 **4)***Использование КЕШ памяти.*

КЕШ память – это область оперативной памяти компьютера, или папка на локальном диске, или папка на диске удаленного компьютера, предназначенная для промежуточного хранения Web документов и последующего более быстрого доступа к ним.

 КЕШ память.

КЕШ оперативной памяти используется в Internet Explorer посредством команд вперед (стрелка вправо) и назад (стрелка влево).

КЕШ на локальном диске организуется Internet Explorer в папке временных файлов.

КЕШ на диске удаленного компьютера подключается посредством диалога со свойством обозревателя (настройка прокси-сервера).

 **5)***Работа в автономном режиме.*

Работа автономно осуществляется с Web документами, копии которых хранятся в папке временных файлов. Работа осуществляется без подключения к Internet.

**5.Исправление ошибок.**

 *Изменение вида кодировки:*

Web документы хранящиеся на сервере с платформой отличной от Windows (например Unix) могут отображаться неправильно по причине использования других принципов хранения файлов.

Эту ошибку мы можем исправить с помощью изменения кодировки (пункт меню Вид).

2 основные кодировки:

* Кириллица Windows-1251 (Windows).
* Кириллица KOI8-R (Unix).

 *Индикация ошибок Internet Explorer:*

* Недоступность сервера – “Невозможно отобразить страницу”
* Отсутствие страницы на сайте – “Невозможно найти страницу”

# **Работа с сервисом электронной почты.**

# **Программа Outlook Express.**

**Существует два основных этапа работы с электронной почтой**:

**1)*Открытие почтового ящика.***

Открытие почтового ящика это автоматическая (посредством Web интерфейса почтовой службы) или ручная (с помощью провайдера) регистрация на почтовом сервере, связанная с созданием папки на диске почтового сервера (то есть создание почтового ящика) и присвоением имени и пароля для доступа к этому почтовому ящику.

**2)*Отправка и получение корреспонденции с помощью почтовой программы***(мы рассмотрим Outlook Express).

Основные функции программы Outlook Express:

1. Хранение сообщений.
2. Создание учетных записей.
3. Получение сообщений из почтового ящика.
4. Ведение адресной книги.
5. Создание и отправка сообщений.
	1. **Хранение сообщений.**

Для хранения сообщений используется служебная папка в программе Outlook Express:

* *Папка входящие* – здесь хранятся вновь поступающие сообщения. Сообщения загружаются в эту папку из почтового ящика по команде доставить.
* *Папка исходящие* – предназначена для временного хранения отправленных сообщений до тех пор, пока они не будут переданы SMTP сервису.
* *Папка отправленные* – здесь хранятся копии отправленных вами сообщений.
* *Папка черновики* – здесь хранятся незаконченные вновь создаваемые сообщения.
* *Папка удаленные* – здесь хранятся сообщения, предназначенные для удаления (почти как корзина).
* *Пользовательские папки* – предназначены для хранения различного рода сообщений по усмотрению пользователя. Эти папки создаются пользователем.

Содержимое папок отображается в виде списка сообщений.

* 1. **Создание учетной записи (и ее хранение).**

*Учетная запись* – это набор настроек указывающих программе Outlook Express, где находится почтовый ящик, сервер и так далее.

Основные поля учетной записи:

* Имя учетной записи.
* Имя пользователя.
* Поле организация (если сообщение посылается организации).
* Электронный адрес (в большинстве случаев сервис SMTP при получении от пользователя сообщения не проверяет достоверность электронного адреса, то есть адрес может быть фиктивным).
* Тип сервиса входящей почты (чаще сервис POP3, следующий по популярности сервис IMAP4).
* Адрес сервера входящей почты. Здесь указывается IP или доменный адрес сервера, где хранится почтовый ящик.
* Адрес сервера исходящей почты. Здесь указывается IP или доменный адрес сервера, который используется для отправки сообщений.
* Имя для входа. Указывается имя пользователя, зарегистрированное в сервисе входящей почты и используемое для доступа к почтовому ящику.
* Пароль. Используется для авторизации доступа к почтовому ящику.
* Порт SMTP сервиса. Здесь указывается номер порта, к которому обращается программа Outlook Express для отправки сообщений (стандартный 25 порт).
* Порт сервиса POP3. Здесь указывается номер удаленного сервера (сервера входящей почты).
	1. **Получение сообщений.**

Оно производится с помощью команды доставка. Сообщение попадает в папку входящие и отображается в виде списка сообщений. Просмотр вновь полученных сообщений возможен с помощью открытия отдельного окошка или с помощью команды открыть.

Сообщение сохраняется на жестком диске в формате **.eml**

При получении сообщений может использоваться их автоматическая сортировка. Для этого создаются правила для сообщений почты:

1. Определение условия (критерия) по возникновению которого в отношении сообщения нужно произвести определенное действие.
2. Определение действия.

Настройка правил помогает не только сэкономить время при сортировке сообщений, но и сэкономить денежные средства при загрузке сообщений (так как можно удалять сообщения из сервера без загрузки, то есть без переноса в папку входящие).

**4.Адресная книга.**

Адресная книга предназначена для создания и хранения контактов.

**Контакты** – это электронный адрес пользователя и персональная информация пользователя.

Контакты могут создаваться в адресной книге:

* *Автоматически* – с помощью специальной настройки программы Outlook Express:

”Помещать в адресную книгу адреса отправителей сообщений, на которые вы отвечаете”

* *Ручной* – создать контакт.
* *Полуавтоматический* – добавить контакт в адресную книгу.

**5.Создание сообщения.**

Оно может происходить тремя способами:

**1)***Создание оригинального сообщения с помощью команды “создать сообщение”.*

Поля нового сообщения:

* Поле. О*т* или *from*: (адрес отправителя).
* Поле *Кому* или *to*: (адрес получателя сообщения). Здесь указывают электронные адреса получателей сообщений (через запятую или точку с запятой, если получателей несколько). Адрес должен быть полным.
* Поле *Копия* или *cc*: (функции как у предыдущего поля).
* Поле *Скрытая копия* или *fcc*: Используется для скрытной рассылки сообщения сразу в несколько адресов.
* Поле *Тема* или *Subject*. Используется для краткого обозначения сути письма (содержимое, кому отправляется, от кого).
* Поле *Тело письма.* Используется для помещения текста сообщения. Это поле не обязательно для заполнения.
* Поле *Прикрепленный файл*. Используется для отображения названий и размеров файлов, присоединяемых к сообщениям.

**2)***Создание ответа.*

Команда “ответить” используется для создания ответа в отношении полученного вами письма. Создается новое сообщение с указанием адреса отправителя оригинального письма в поле “кому”, с указанием оригинальной темы оригинального письма в поле “тема” (добавляется аббревиатура Re) и копируется текст оригинального сообщения в поле “тело сообщения”.

**3)***Пересылка или перенаправление письма.*

Также производится в отношении полученного вами письмаи выполняется команда переслать. Появляется окно “создание нового сообщения”, в которое копируется тема оригинального письма с добавлением аббревиатуры Fw и копируется текст оригинального письма.

Вложение или присоединение файлов к сообщению производится с помощью: Вставка => вложить файл.

Незаконченное новое сообщение сохраняется в папку черновики командой “сохранить” или при попытке закрыть окно создания нового сообщения.

# **Сервис новостей (NEWS).**

# **Работа с телеконференциями.**

Для работы с сервисом новостей используются или специальные клиентские программы чтения новостей или используется работа с Web интерфейсами.

**NNTP** – этот протокол подразумевает определенный интерфейс для работы с сервисом новостей.

Одной из программ для чтения новостей является программа Outlook Express. Эта программа может выполнять следующие функции:

1. Создание учетной записи новостей.
2. Управление группами новостей.
3. Синхронизация групп новостей, чтение сообщений в группах новостей.
4. Отправка сообщений группы новостей.

Процесс подключения отдельных групп для работы с группой новостей называется **подписка**.

Работа с группами разбивается на 2 этапа:

Загрузка сообщений (синхронизация):

* Только заголовков.
* Полностью.

Пользователь может осуществлять дополнительную синхронизацию (загрузку) специально помеченных им сообщений (при загрузке сначала только заголовков). Эти этапы проходят быстро (загрузка заголовков и затем загрузка выделенных заголовков).

 Синхронизация

 Киров Москва

# **Поиск информации.**

Internet

Поисковая система

Например, принтер

ПК пользователя

Например, хранение данных

Поисковый сервис выдает список URL адресов, которые указывают на документы, соответствующие запросу пользователя. Общение с поисковым сервисом осуществляется с помощью Web интерфейса.

Поисковый сервис делится на 3 части:

1. *Web интерфейс*.
2. *Поисковый робот*.
3. *СУБД*. В поисковой базе СУБД хранятся ссылки (URL адреса, хранящиеся в Internet). Помимо списка URL адресов в поисковой СУБД также хранится ключевое содержимое соответствующих документов.

**Поисковый робот** – это программа, осуществляющая автоматическое сканирование Web ресурсов (индексирование) на предмет появления новых, модификаций существующих и удаление старых Web ресурсов.

В результате сканирования обновляется поисковая база.

Существует 2 способа работы поискового робота:

1. Сканирование роботом Web ресурсов; заполнение базы данных.
2. Обращение пользователя к поисковому сервису через Web интерфейс (используется специальная форма для указания запроса).

 Форму (пользователь передает сообщение (запрос))

База адресов

База ключей

 Список всех

 Список URL слов русского

 адресов языка

Например:

|  |  |
| --- | --- |
| арбуз | 1 |
| арбалет | 2,3,5,10 |
| арка | 1 |

|  |  |
| --- | --- |
| 1 | www.mail.ru |
| 2 | www.rui.ru |
| 3 |  |

Каждая отметка сопровождается весом, например 2:50; 3:1 (вес показывает сколько раз слово встречается в документе).

Например, в поисковую систему пользователь вводит “арбалет”.

1.Ранжирование (вес), определяющее ценность документа рассчитывается на основании количества нахождений слова в документе.

2.Нахождение слова в составе заголовка документа.

3.Нахождение слова в составе ссылки документа.

Эффективность поисковой системы сводится к объему поисковых баз, а также определяется эффективностью алгоритма ранжирования документов, наличием языка запросов.

**Язык запросов** – это набор команд, позволяющий пользователю уточнить поисковый запрос и получить более точные результаты на свой запрос.

Например:

Если нужно найти в предложении Российская федерация, пишем, федерация пробел Российская. Будет найден пятый документ.

|  |  |
| --- | --- |
| Российская | 1,5,10,17 |
| … … … … |  |
| Федерация | 2,5,11,12 |

# **Средства разработки Web сайтов**

Средства разработки состоят из среды разработки и языка разработки.

* 1. **Язык разметки HTML.**

(Это основной инструмент разработки Web сайтов).

**Язык разметки HTML** – это система команд, основная задача которой – представление текстовых и мультимедийных элементов Web страницы в наиболее благоприятной для восприятия форме.

(Первую часть определения можно отнести к любому языку разметки). В дополнение к командам разметки в язык HTML были добавлены команды для работы с гиперссылками.

Язык HTML позволяет:

1. Разместить элементы Web страницы в удобной для восприятия форме:
	* Установить тип текстовых фрагментов, их стиль.
	* Использовать в составе Web станицы элементы сложного форматирования (таблицы, списки и так далее).

1. Определять предназначение Web страницы в целом, отдельные ее свойства (кодировка, сроки использования); параметры взаимоотношения данной Web страницы с другими Web документами (наследование).
2. Поддерживать интерактивное взаимодействие с Web сайтом (это реализуется с помощью форм).
3. Переходить к другим Web документам посредством гиперссылок.

Документ HTML состоит из нескольких частей:

1. Основная составляющая – *это дескриптор (или ТЭГ) HTML*.

**Дескриптор (ТЭГ)** – это команды языка HTML, представляющие собой слово или аббревиатуру, заключенные в треугольные скобки(<HTML>).

Дескрипторы бывают двух видов: ***одиночные*** и ***парные***.

* ***Одиночный дескриптор*** – определяет реакцию браузера в том месте, в котором он находится на Web странице.

Пример.

Дескриптор <BR> переводит в то место, где встретится на Web странице.

* ***Парный дескриптор*** – определяет действие в отношении фрагмента Web страницы, в которой находится между открывающей и закрывающей частью парного дескриптора.

 Пример.

<I> … … … </I>

 **2)**  *Атрибут дескриптора.*

В состав начального дескриптора могут применяться дополнительные параметры, например:

Дескриптор абзаца (параграфа).

<P align=”center”>

----------------------

----------------------

<P>

Текст будет отображаться выровненным по центру.

Здесь Align является параметром, а center – значением данного параметра. (Такой параметр называется в языке HTML *атрибут*).

**Атрибут дескриптора** – определяет дополнительное свойство дескриптора, уточняет его действие. В случае парного дескриптора используется внутри открывающего дескриптора.

У дескриптора может быть один атрибут, несколько атрибутов или может вообще их не быть.

Пример с несколькими атрибутами.

<Font color=”red” size=3>

 ---------------------------------

</Font>

 Два атрибута: color - атрибут цвета и size - атрибут размера.

**2.Сценарий (SKRIPT).**

**Сценарий (SKRIPT)**  - это программный код, обеспечивающий работу интерактивных и динамических элементов Web сайта.

Сценарии бывают двух видов:

**1)*Серверный*** – это программа, выполняющаяся на Web сервере.

**2)*Клиентский*** – это программа, выполняющаяся с помощью интерпретатора браузера (то есть на компьютере пользователя).

Наиболее частая функция сценария – взаимодействие с базами данных и создание их на основе Web страниц.

*Клиентский* сценарий в отличие от языка разметки HTML предоставляет гораздо больше возможностей:

* Для взаимодействия с операционной системой и браузером;
* По управлению элементами интерфейса операционной системы и браузером.

*Серверный* сценарий чаще всего используется для отображения в виде Web страниц содержимого баз данных, которые хранятся на сервере.

Преимущества *серверного* сценария:

Более широкий спектр возможностей по обработке взаимодействия пользователя и Web сайта.

Преимущества *клиентского* сценария:

Большая скорость обработки и экономия ресурсов (за счет уменьшения передачи данных между сервером и компьютером пользователя).

**3.Среда программирования.**

**Среда программирования** (среда разработки) – это программное обеспечение, позволяющее эффективно создавать элементы Web станиц, разрабатывать сценарии и осуществлять их отладку.

Основные из них:

1. MS Font Page.
2. Macromedia Dream Weaver.
3. Блокнот.

# **Основные дескрипторы HTML**

**1.*Заголовок HTML документа*** – определяет назначение HTML документа и его основные свойства.

Формируется при помощи парного дескриптора <HEAD>:

<HEAD>

-----------

</HEAD>

Содержимое заголовка обычно не отражается на экране пользователя при просмотре Web документа.

**2.*Тело Web документа*** – содержит содержательную часть Web документа (информация, графика и так далее).

Формируется при помощи парного дескриптора <BODY>:

<BODY>

------------

</BODY>

**3.*Дескриптор, определяющий начало и конец документа*.**

Формируется при помощи парного дескриптора <HTML>. Начальный находится в самом начале HTML документа, конечный – в самом конце Web документа.

 <HTML>

Заголовок

 Тело

 </HTML>

# **Заголовок HTML документа и его основные свойства**

Формируется при помощи МЕТА дескрипторов (<META>).

**Дескриптор <META>** используется для описания свойств HTML документа. Он является одиночным, однако содержит следующие атрибуты:

* ***http-equiv***
* ***name***
* ***content***

Первые два используются для обозначения свойств (то есть дают свойству имя), последний используется для придания свойству значения.

Например:

<META http-equiv=”expires” content=”Friday, 21-Feb-03 00:00:00GMT”>

Используется для обозначения годности документа.

<META http-equiv=”Content-Type” content=”txt/html; charset=Windows1251”>

Указывает на тип документа, его отношение к текстовому документу и его кодировка.

<META name=”DECRIPTION” content=”описание”>

Дает описание содержимого Web страницы.

<META name=”KEYWORDS” content=”…, …, …”

Ключевые слова указываются через запятую (можно пробел). Определяет ключевые слова данного Web документа.

* Свойства HTML документа, обозначенные ***http-equiv*** передаются на компьютер пользователя в составе HTTP заголовка. Браузер обязательно интерпретирует http заголовок. Инструкции, содержащиеся в http заголовке особенно важны для правильного отображения HTML документа.
* Свойства HTML документа, определенные с помощью атрибута ***name*** носят рекомендательный характер и в большинстве своем игнорируются браузером.

Подобные МЕТА свойства (или свойства HTML документа) обрабатываются поисковыми системами.

Пример.

<BODY BGCOLOR=”red”/”gray”/”# FFCC22”

BACKGROUND=1.jpg>

# **Структура дескриптора BODY**

**1.***Атрибут* ***bgcolor*** – для отображения фона HTML документа.

Пример.

bgcolor=”white” (black, blue и другие цвета)

Данному атрибуту может присваиваться свойство RGB (RGB – red, green, blue).

Пример.

bgcolor=”#FFCC22”, где

FF – уровень красной составляющей, CC - уровень зеленой составляющей, 22 – уровень синей составляющей.

В шестнадцатеричной системе исчисления FFFFFF – белый цвет, 000000 – черный цвет.

Данная кодировка (RGB) может использоваться во всем HTML документе.

**2.** *Атрибут* ***background*** – определяет фон с помощью графического формата.

Пример.

background=”2.jpg”

В значении атрибута указывается название документа и путь к нему. (Если указано только название, то данный файл должен находиться в той же папке, что и HTML документ).

Если путь указан так:

background=”/fold1/2.jpg”, то папка будет искаться начиная от корневого каталога.

**3.** *Атрибут* ***TEXT.***

Значения данного атрибута могут быть такими же**,** как у атрибута bgcolor.

Эти же правила верны и для следующих атрибутов:

**4.** *Атрибут* ***LINK=***  определяет цвет ссылки.

**5.** *Атрибут* ***VLINK=*** определяет цвет уже посещенной ранее ссылки***.***

**6.***Атрибут* ***ALINK=*** определяет цвет активной ссылки (то есть ссылки, указывающей на текущий документ).

**7.** *Атрибут* ***TOPMARGIN***=(верхнее поле).

**8.** *Атрибут* ***LEFTMARGIN***=(левое поле).

Пример.

TOPMARGIN=”20” (в пунктах).

LEFTMARGIN=”10” (в пунктах).

Все содержимое HTML документа будет смещено относительно левого и верхнего края HTML документа.

 20

10

Пример парного дескриптора:

<BODY bgcolor=”…” LINK=”…”>

-----------------------

</BODY>

# **Форматирование текста**

В HTML документе можно помещать комментарии (они игнорируются браузером и используются для служебных целей).

**Комментарии.**

**1)** *Одиночный дескриптор*.

<!--

Все, что находится между первой и последней треугольной скобкой считается комментарием.

-->

 **2)** *Парный дескриптор* <**COMMENT**> .

<COMMENT>

------------------

</COMMENT>

Браузеры игнорируют “обычные” средства форматирования текста, такие как последовательность пробелов, знаки табуляции, переводы строки и прочие. Чтобы отформатировать документ используются средства ***физического*** *и* ***логического***форматирования.

К ***физическим*** относятся дескрипторы, которые определяют напрямую внешний вид текста.

К ***логическим*** относятся дескрипторы, которые определяют логический статус (значение) текста.

**Физические средства форматирования.**

**1.***Одиночный дескриптор* ***<*br*>*** *-* перевод строки (разрыв строки)

.

**2.***Парный дескриптор <***nobr>…</nobr***> -* запрещает разрыв текста, находящегося в контейнере. Данный дескриптор еще называют дескриптор жесткого переноса.

По причине различных условий, в которых отражается HTML документ (разные браузеры, видеорежимы, разрешение экрана) HTML документ может отображаться с нарушением взаимного расположения элементов.

**3.** *Одиночный дескриптор <***wbr***>* - определяет место мягкого переноса. Данный дескриптор используется совместно с дескриптором <nobr**>** и размещается внутри его контейнера.

Браузер осуществляет перенос в соответствии с этим дескриптором только в том случае, если строчка не помещается в ширину окна и обычный перенос запрещен с помощью дескриптора <nobr**>**.

**4.** *Парный дескриптор<***p>…</p***>* - параграф или абзац. Текст, находящийся внутри дескриптора оформляется в виде параграфа или абзаца.

Атрибут ***ALIGN=*** выравнивание текста.

Возможные значения:

* =LEFT – по левому краю.
* =CENTER – по центру.
* =RIGHT – по правому краю.

**5.** *Парные дескрипторы заголовка.*

Бывают шести уровней:

<**H1**>**…**</**H1**>

 **… … …**

<**H6**>**…**</**H6**>

Текст заголовка выравнивается, до и после текста интервал.

 Допустимо использование атрибута ***ALIGN.***

Данные дескрипторы имеют большое значение для повышения информационной ценности документа в ”глазах” поисковой системы. (Дескрипторы заголовка – один из основных инструментов для повышения ценности).

**6.** *Дескриптор* ***<*HR*>*** *– одиночный горизонтальный дескриптор.*

Там где находится данный дескриптор, отображается горизонтальная черта.

 Атрибуты:

* ***ALIGN***
* ***COLOR*** –цвет черты (значения как у*background*).
* ***WIDTH*** – ширина линии (в пунктах).
* ***SIZE*** – толщина линии (в пунктах).
* ***NOSHADE*** – без тени.

**7.** *Парный дескриптор <***FONT***>***…**</ **FONT** >*.*

Предназначен для определения шрифта (начертания) отображаемого текста (курсив и так далее).

Атрибуты:

* ***FACE*** – название шрифта.

Пример:

FASE=”Times New Roman”

Текст, который помещен внутри контейнера будет отображен данным шрифтом. В связи с тем, что сложно угадать есть ли у пользователя такой же шрифт, допускается перечисление нескольких шрифтов через запятую в дескрипторе.

* ***COLOR***
* ***SIZE=3*** *-**по умолчанию*. (Размер символов может быть от 1 до 7).

**8.***Одиночный дескриптор* ***<*BASEFONT*>***

Данный дескриптор устанавливает базовое значение текстовых значений Web страниц, находящихся после *<*BASEFONT*>.* Атрибуты как у дескриптора *<*FONT*>*.

**9.** *Парный дескриптор <***PRE***>***…**</ **PRE** > *- дескриптор пре форматирования.*

Используется для форматирования текста стандартными способами (табуляцией, последовательностью пробелов и так далее).

Пример:

<PRE>

\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_
|\_ \_ \_ \_ \_ \_ \_|\_ \_ \_ \_ \_ \_ \_|

|\_ \_ \_ \_ \_ \_ \_|\_ \_ \_ \_ \_ \_ \_|

</PRE>

**10.** *Парный дескриптор <***DIV***>***…**</ **DIV** >

Используется для выравнивания группы элементов таблицы (группа – несколько текстовых или графических элементов). Все, что внутри таблицы выравнивается.

Пример:

*<*DIV*>*

*Align*

*</*DIV*>*

**10.** *Парный дескриптор <***Center***>***…**</ **Center** >

Все содержимое контейнера данного дескриптора выравнивается по центру.

# **Советы по форматированию текстовых фрагментов**

**1.**Использовать мягкий контраст между фоном и текстом (например, черный и белый).

**2.**Использовать как можно меньшее число шрифтов (2-3). Принято для отображения заголовков использовать шрифты типа Arial (то есть без засечек), а шрифты с засечками для самого текста (например, ТАЙМС).

**3.**Реже используйте выделения, подчеркивания и жирные шрифты.

Существуют так же дескрипторы:

1. **<b>…</b>** - выделяет текст в контейнере с помощью жирного шрифта.
2. **<i>…</i> -** выделяет текст в контейнере с помощью наклонного текста (курсив).

**Пример логического форматирования.**

**1)** *Парный дескриптор <***CITE***>***…**</ **CITE** >

Используется для цитат. Обычно браузером отображается в виде курсива (наклонного текста).

**2)** *Парный дескриптор* <**BLOCKQUOTE**>**…**<**/BLOCKQUOTE**>

Используется для выделения больших цитат. (Браузер обычно делает отступ от левого края).

В большинстве случаев браузер не допускает пересечение дескрипторов.

Пример пересечения:

*<*PRE*>*

*<*FONT*>*

</PRE >

</FONT >

(В лучшем случае просто не запустится).

Но есть и исключения, например **<**b>…**<i>…</b>**…</i**>**

В том месте, где пересеклись дескрипторы <i> и **<**b>, текст будет выделен жирным курсивом.

**3)** *Парный дескриптор* <**U>…</U**> - подчеркивание.

# **Специальные символы**

Используются для отображения символов, которые являются в HTML управляющими и не могут быть отображены обычными способами.

Например, (>) - больше или (<) – меньше. Браузер ее воспринимает как начало контрольной последовательности.

Для того чтобы все-таки отобразить эту скобку необходимо набрать последовательность:

1. **&lt** - отобразится левая скобка.
2. **&gt** - отобразится правая скобка.
3. **&copy** - отобразится значок авторского права.
4. **&deg** - отобразится значок температуры.
5. **&nbsp**; отобразится символ пробела (жесткого пробела). Если пробел указан с помощью данного специального символа, то разрыв строки не допускается, так как браузер будет видеть слова и пробелы как одно слово.

# **Сложное форматирование**

В отличие от простого форматирования символов и абзацев в HTML сложное форматирование производится с помощью многоуровневых конструкций, основу которых составляет дескриптор, задающий способ форматирования (список, таблица) и элементы более низкого уровня, используемые непосредственно для отображения текста.

Пример:

<OL>

<LI>abc</LI>

<LI>def</LI>

</OL>

Данная конструкция формирует нумерованный список, при этом дескриптор <OL> является дескриптором, который определяет способ форматирования. В браузере отобразится:

1.abc или a. abc

2.def b. def

1. *Парный дескриптор* <**LI**>***…</*LI*>*** (от английского слова List - список) используется для отображения элементов списка. Атрибутов не имеет.

2. *Парный дескриптор* <**OL**>**…</OL>** - формирует нумерованный список (или упорядоченный).

* Изменение способа нумерации производится с помощью *атрибута* ***TYPE.***

 Например:

 <OLTYPE=[1/a/A/I] , где

1 - арабские цифры.

а – прописные буквы.

А – заглавные буквы.

I – римские цифры.

* *Атрибут* ***START*** позволяет указать номер, с которого начинается нумерация.

 Например, если START=3, то нумерация начнется с трех или с буквы “c”.

2. *Парный дескриптор* <**UL**>**…</UL>**

Формирует не упорядоченный список (маркерный).

* abc
* def

Например:

<ULTYPE=disc, square, circle>

<LI>abc</LI>

**</**UL**>**

Где disc – жирная точка; square - квадрат; circle – не закрашенная окружность.

При отображении элементов списка браузер делает отступ слева.

**Многоуровневые списки.**

Например:

<ULTYPE=disc, square, circle>

<LI>abc</LI>

 <UL>

 <LI>def</LI>

<LI>ghi</LI>

**</**UL**>**

**</**UL**>**

Отобразиться:

* abc
	+ def
	+ ghi

Браузер будет делать очередной отступ слева.

**Таблицы.**

**1**. *Парный дескриптор* <**TABLE**>***…</*TABLE*>***

В его контейнере должны быть сформированы элементы таблицы (ячейки) в которых и будет информация.

**2**. *Парный дескриптор* <**TR**>***…</*TR*>*** - формирует строку таблицы.

**3**. *Парный дескриптор* <**TD**>***…</*TD*>*** - формирует ячейку внутри строки.

Пример:

<TABLE>

|  |  |
| --- | --- |
| <TD>1</TD> | <TD>2</TD> |
| <TD>3</TD> | <TD>4</TD> |

<TR> <TR>

<TR> <TR>
 </TABLE>

**Атрибуты дескриптора <TABLE>**:

* *Атрибут* ***WIDTH*** – определяет ширину таблицы (в процентах или пикселях).

Пример:

<TABLE WIDTH=100% или WIDTH=200>

* *Атрибут* ***HEIGHT*** – высота (принимает такие же значения, как и атрибут WIDTH).
* *Атрибут* ***ALIGN*** – выравнивание таблицы по ширине относительно краев окна.
* *Атрибут* ***BORDER*** – определяет ширину рамки (если значение данного атрибута 0, то рамки отсутствуют).
* *Атрибут* ***bgcolor*** – для формирования фона таблицы с помощью цвета.
* *Атрибут* ***background*** - для формирования фона таблицы с помощью картинки.
* *Атрибут* ***cellpadding*** – для указания отступов от краев ячейки (в пикселях).
* *Атрибут* ***cellspacing*** – расстояние между ячейками.

Если значение атрибута больше нуля,

то таблица выглядит так:

Обычная таблица

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

* *Атрибут* ***bordercolor*** – цвет границ ячеек (рамки).

**Атрибуты дескриптора <TR>**:

1. *Атрибут* ***ALIGN*** – выравнивание текста в ячейках.
2. *Атрибут* ***VALIGN*** – определяет вертикальное выравнивание текста в ячейках.

Возможные значенияатрибутаVALIGN:

* ***=***TOP – по верху.
* ***=***MIDDLE – по центру.
* ***=***BOTTOM – по нижнему краю ячейки.

***3)****Атрибут* ***bgcolor*** – для указания фона строки.

**Атрибуты дескриптора <TD>**:

* *Атрибут* ***ALIGN****.*
* *Атрибут* ***VALIGN****.*
* *Атрибут* ***bgcolor****.*

Вышеперечисленные атрибуты действуют так же, как для дескриптора <TR>.

 Дополнительно атрибуты:

* *Атрибут* ***COLSPAN*** – объединение ячеек, находящихся в одной строке. Значение атрибута равно количеству близлежащих ячеек находящихся в одной строке, которые будут объединены.

Пример:

<TABLE>

|  |
| --- |
| <TD COLSPAN=2>1</TD> |
| <TD>3</TD> | <TD>4</TD> |

<TR> </TR>

<TR> </TR>

</TABLE>

* *Атрибут* ***ROWSPAN*** – объединение ячеек, находящихся в одном столбце. Значение атрибута указывает количество близлежащих ячеек находящихся в одном столбце, которые будут объединены.

Пример:

<TABLE>

|  |  |
| --- | --- |
| <TD ROWSPAN=2>1</TD> | <TD>2</TD> |
| <TD>4</TD> |

<TR> </TR>

<TR> </TR>

</TABLE>

# **Форматирование текстовых фрагментов HTML документа**

Кроме ширины и высоты таблицы в целом можно регулировать ширину и высоту ячеек (то есть использовать такие же атрибуты в дескрипторе <TD>, в пикселях и в процентах).

Если они не указаны, то браузер выравнивает размер ячейки по ширине текста.

Таблица является наиболее часто употребляемым инструментом в Web дизайне, позволяющим точно разместить на таблице отдельные мультимедийные элементы. Подобно спискам, таблицы могут быть вложены в другие таблицы.

Пример:

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

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

 |  |
|  |  |

Чтобы добавить такую таблицу в контейнере дескриптора ячейки пишут следующее:

<TD>

<TABLE>

<TR><TD></TD><TD></TD></TR>

<TR><TD></TD><TD></TD></TR>

</TABLE>

 <TD>

Фон отдельных ячеек могут отображать не все браузеры.

# **Размещение графических изображений в HTML документе**

Для размещения изображений используются в основном два графических формата:

* JPG
* GIF

**1.Формат JPG.**

*Плюсы* - Графическое изображение, сохраненное в данном формате, имеют наименьший объем.

*Минусы* – при сохранении происходит довольно сильное сжатие картинки, а при сжатии изображение теряет качество и наблюдается эффект пикселизации.

Пример:

Картинка сжалась в девять раз.

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

|  |
| --- |
|  |

Девять пикселей заменяются одним пикселем среднего цвета.

Изображение сохраняется в формате **.jpg** в том случае, если необходимо сохранить максимальное количество оттенков.

**2.Формат GIF.**

При сохранении картинок в этом формате также происходит сжатие, но пикселизации не происходит.

*Минусы* – изображение сохраняется с помощью ограниченного количества цветов (обычно 256 цветов) => изображение будет не очень красочным.

С помощью формата **.gif** сохраняются навигационные кнопки, навигационные графические элементы и фоновые изображения.

Для внедрения изображения используется *одиночный*  *дескриптор* <**IMG**>.

* *Атрибут* ***SRC*** – основной атрибут. Он указывает путь и имя файла, в котором содержится графический элемент.

Пример:

<IMG SRC=1.gif>

* *Атрибут* ***WIDTH*** – ширина внедряемого изображения (в пикселях).
* *Атрибут* ***HEIGHT*** - высота внедряемого изображения (в пикселях).
* *Атрибуты* ***VSPACE,*** ***HSPACE*** – устанавливают отступ текста от внедряемого в HTML документ изображения.
* *Атрибут* ***BORDER*** – задает толщину рамки вокруг изображения.
* *Атрибут* ***ALT*** – задает альтернативный текст, который появляется в том месте, где должны быть картинки.

Пример:

ALT=”Здесь была картинка”.

Обычно атрибут используют при размещении навигационных изображений (ссылок), а так же при размещении изображений, о которых надо дать знать поисковой системе.

* *Атрибут* ***ALIGN*** – выравнивание.

Возможные значения:

=LEFT, CENTER, RIGHT.

Добавляются значения: =TOP, MIDDLE, BOTTOM.

* Если не используются *атрибуты* ***WIDTH, HEIGHT***, то изображение внедряется оригинального размера.

Если используется только один атрибут, например *HEIGHT*, то браузер уменьшит или увеличит ширину пропорционально изображению.

# **Использование гипрессылок**

**Гипрессылка** *–* это URL адрес того документа, который привязан к одному из элементов Web страницы (текстовому или графическому).

Для внедрения ссылок в HTML документ применяется парный дескриптор <**A**>**…</A>**.

Пример:

<A HREF=”http://www.kirov.ru/price.htm”>

Все, что находится внутри контейнера, называется элемент привязки (текстовый фрагмент или графический элемент).

</A>

В результате на экране отображается подчеркнутый выделенный текст.

Пример привязки к изображению:

<A>

<IMG SRC=”1.gif”>

</A>

Изображение, к которому привязывается ссылка, называется навигационной картинкой (изображением).

В случае привязки к картинке ссылки вокруг картинки формируется рамка (чаще всего синего цвета). Чтобы ее убрать, необходимо указать:

<IMG border=”0”>

Значением *атрибута* ***HREF*** может быть абсолютный адрес (URL адрес) или относительный адрес (имя файла, в котором хранится HTML документ, указываемый с помощью ссылки). Например, HREF=”1.htm”. Браузер будет искать данный документ в текущей папке, то есть в той, в которой находится активный документ.

Исключение составляет *дескриптор* <**BASE>**

<BASE URL=”http://www.kirow.ru”**>**

В случае использования данного дескриптора в заголовке HTML документа (то есть контейнера HEAD)относительные адреса (указанные в виде значений HREF) будут восприниматься браузером не относительно текущей папки, а относительно URL адреса, указанного в значении атрибута URL.

###### Использование дескриптора <A>…</A>

###### для указания отдельных элементов HTML документов.

**1.** *Атрибут* ***NAME*** используется для указания места, до которого необходимо пролистать

HTML документ.

 Пример:

<A NAME=”ZDES”>

**</**A**>**

В этом случае внутри дескриптора <A> находится якорь – это элемент HTML документа, который может быть адресован с помощью гиперссылки. Якорь не выделяется браузером.

Для того чтобы с помощью ссылки адресоваться к якорю надо после URL адреса добавить # и указать имя якоря.

Пример:

<A HREF=”http://www.kirov.ru/price.htm#1>

Название якоря может формироваться без указания URL адреса HTML документа.

Пример:

<A HREF=”#1”>

Данный якорь будет искаться в текущей папке.

При использовании абсолютного адреса необходимо использовать аббревиатуру http (например, <http://www.kirov.ru>).

# **HTML форма**

**Форма** (в документе HTML) – это средство, позволяющее организовать диалог пользователя и Web сайта.

Основные виды HTML форм:

1. Кнопки.
2. Списки.
3. Текстовые поля.
4. Переключатели (флажки).
5. Радиокнопки.

Элементы управления HTML форм позволяют пользователю вводить информацию с помощью браузера и отправлять её на обработку Web сайту.

Для обработки форм, заполненных пользователем, на Web сайте используется программа-обработчик. В ответ на данную форму эта программа формирует HTML код.

**Группа дескрипторов, формирующих HTML форму**.

**1.** *Парный дескриптор* **<FORM>** - задает свойство формы и содержит в своем контейнере элементы этой формы.

Атрибуты:

* *Атрибут* ***NAME*** – для задания имя формы.

Например,NAME=”anketa”.

! Каждый элемент HTML формы, как и сама форма, должны быть обязательно поименованы.

* *Атрибут* ***METHOD*** – определяет, каким способом HTML форма будет отправляться к программе-обработчику.

Есть два варианта:

1)=POST – используется чаще всего.

 2)=GET – используется для отправки HTML формы с помощью URL адреса.

(По умолчанию задается метод GET).

* *Атрибут* ***ACTION*** – задает местоположение программы-обработчика. Значением атрибута служит адрес (абсолютный или относительный).

Например,

ACTION=”executive.php” – файл, в котором хранится программа-обработчик.

Или ACTION=“mailto:a@kirov.ru” – пользовательская форма будет отсылаться по указанному электронному адресу.

* *Атрибут* ***ENCTYPE*** – тип кодировки.

Варианты:

=”text/htm/”

=”text/plain” – обычный текст без элементов форматирования.

# **Элементы HTML форм**

**1.Текстовые поля.**

 Бывают двух видов:

1. Многострочное. *Парный дескриптор* **<TEXTAREA>…</TEXTAREA>**

Атрибуты:

* + *Атрибут* ***NAME.***
	+ *Атрибут* ***ROWS*** – высота поля в строчках.
	+ *Атрибут* ***COLLS*** – ширина поля в символах.

Размещенный в контейнере дескриптора **<**TEXTAREA**>** текст будет отображаться внутри этого текстового поля (по умолчанию).

 

1. Однострочное. Формируется с помощью *одиночного дескриптора* **<INPUT>**.

(Данный дескриптор также используется для переключателей, **радиокнопок**).

 Атрибуты:

* *Атрибут* ***NAME***.
* *Атрибут* ***TYPE*** – задает тип создаваемого элемента. Для создания однострочного поля значение атрибута TYPE=”Text”.
* *Атрибут* ***SIZE*** – ширина поля в символах.
* *Атрибут* ***MAXLENGTH*** – максимальная длина вводимого текста.
* *Атрибут* ***VALUE*** – используется для хранения значения элемента формы.

Пример:

 Мы можем задать значение поля по умолчанию VALUE=”Например, пошел дождь”.



**2.Переключатель или флажок.**



Формируется с помощью *дескриптора* **<INPUT>,**  *атрибут* ***TYPE*** которого принимает значение TYPE=”CHECKBOX”.

<INPUT TYPE=”CHECKBOX”>

 У данного дескриптора имеются также атрибуты:

* *Атрибут* ***NAME***.
* *Атрибут* ***CHESKED*** – если он указан, то флажок будет установлен.

**3.Радиокнопка.**

(Выглядит как жирная точка).

Достигается с помощью двух дескрипторов **<INPUT>**, имеющих одинаковое имя. Например, NAME=”1”.

<INPUT TYPE=”RADIO” NAME=”1” VALUE=”GENAT”>

<INPUT TYPE=”RADIO” NAME=”1” VALUE=”XOLOST”>



 

**4.Текстовое поле типа пароль.**

<INPUT TYPE=”PASSWORD”>

Данный дескриптор формирует текстовое поле типа пароль. Работает также как и однострочное текстовое поле, но значение отображается в виде звездочек.



**5.Кнопка.**

* <INPUT TYPE=”BUTTON”>

Используется для вызова клиентских скриптов (сценариев).

 Выглядит так:

 

* Кнопка созданная с помощью:<INPUT TYPE=”SUBMIT” VALYE=”Пуск”> предназначена для отправки формы на обработку в Web сайт.

Выглядит так:

 

* *Кнопка сброса формы*.

Предназначена для установления первоначального значения формы.

<INPUT TYPE=”RESET” VALUE=”Сброс”>

Выглядит так:

 

**6.Списки выбора** (меню) – это многоуровневая конструкция.

*Парный дескриптор* **< SELECT >…</ SELECT >** (дескриптор первого уровня)формирует свойства меню.

Атрибуты:

* *Атрибут* ***NAME***.
* *Атрибут* ***SIZE***.
	+ Если данный атрибут равен единице, меньше единицы или не указан, то формируется выпадающее меню.

 Выглядит так:



* + ЕслиSIZE больше единицы, то формируется прокручиваемый список.

Выглядит так:

 

Внутри контейнера дескриптора<SELECT> находятся элементы меню, которые формирует одиночный дескриптор <**OPTION**> (дескриптор второго уровня).

Пример:

<SELECT NAME=”Menu”>

<OPTION>Файл

<OPTION>Печать

</SELECT>

Ширина меню определяется максимальным значением элемента меню.

* Атрибут **MULTIPLE** – для выбора нескольких разных опций.

**Советы преподавателя**:

1. Размещать каждый дескриптор с новой строки.
2. Перед большими конструкциями делать интервал.

HTML форма отсылается на обработку Web сайту в виде записей, которые выглядят следующим образом:

Имя элемента формы(Name) = значение(Value).

Для списков, переключателей, кнопок, радиокнопокобязательно задавать значение атрибута Value. В остальных случаях значение задает пользователь (во время ввода). В случае если при формировании элементов формы типа переключатель (флажок) атрибут Value не используется, в качестве значения будет формироваться предлогOn**.**

 Для списков таковым значением по умолчанию будет назначаться текст, расположенный сразу после дескриптора <OPTION>.

Если вы используете элемент Web формы для передачи информации Web сайту, при передаче формы на обработку должна появиться запись:

Имя элемента формы = его значение.

# **История создания Internet**

Толчком к созданию сети Internet послужила холодная война.

Первые разработки по созданию глобальных компьютерных сетей были осуществлены американским агентством по созданию военных разработок DARPA (агентство было создано в 1957 году).

В **1965** году состоялся первый сеанс удаленной связи между компьютерами. Был выделен специальный проект по созданию сети ARPANET (прообраз сети Internet).

 В **1969** году были созданы первые маршрутизаторы, благодаря этому в этом же году было создано четыре узла маршрутизации, которые позволили в полной мере реализовать ту модель ГВС, которая была изначально принята за основу при создании сети ARPANET.

В **1972** году был запущен стандарт, регламентирующий основу транспортного протокола TCP.

В этом же году появились первые сервисы:

* TELNET (1972).
* FTP (1973).

С помощью этих сервисов пользователь получает удаленный доступ к командной строке сервера. Для передачи сообщений использовали командную строку. Для передачи электронной почты использовался протокол FTP.

В **1977** году появился стандарт на сервис электронной почты SMTP.

В **1983** году произошло официальное внедрение протокола TCP/IP.

В результате расширения сети было принято решение о передаче координации проекта ARPANET из обороны науке. И проект был переименован в NSFNET.

В **середине 80-х** к NSFNET подключились британская сеть, затем канадская, финская и другие.

В **1993** году к Internet присоединилась Россия, и был образован домен .ru

В **1991** году был создан сервис Gopher – прообраз Web сервиса.

# **Основные организационные структуры,**

#  **координирующие работу Internet**

После выхода Internet за национальные рамки основная роль по управлению сетью была передана Всемирной общественной организации по назначению имен и чисел – **ICANN**.

Инженерная организация по развитию Internet – **IETF**.

Основная задача: Развитие старых и создание новых сетевых стандартов.

Развитием Web стандартов занимается **W3C**.

**IOSC** – интернет сообщество. Основная задача – развитие содержательной части, правовые вопросы.

**РСИЦ** – региональный сетевой информационный центр (в России). Этот центр занимается разработками, опираясь на стандарты четырех вышеперечисленных организаций.

В каждой стране есть такой центр.

# **Листы каскадных стилей или CSS**

**CSS** – это набор правил для оформления и форматирования различных элементов HTML документов.

Правила подключения стиля к конкретному HTML документу:

 1) С помощью *одиночного* *дескриптора* **<LINK>**.

<LINK REL=”Stylesheet” TYPE=”text/css” [MEDIA=”SKREEN”] HREF=”URL”>

MEDIA – среда, в которой будут работать стили.

При задании значения атрибуту MEDIA правила будут действовать только при выводе страниц на:

MEDIA=”PRINT” – печать.

MEDIA=”SCREEN” – экран.

 2) При помощи *парного дескриптора* **<STYLE>…<STYLE>**.

<STYLE TYPE=”text/css” [MEDIA=”…”]>…</STYLE>.

В контейнере дескриптора <STYLE> располагаются правила, определяющие свойства отдельных элементов страницы.

 3) Определение стиля непосредственно внутри элемента страницы.

<P STYLE=”…”>…</P>.

 4) Использование пользовательских стилей (то есть с помощью настроек браузера).

# **Правила описания стилей**

Правила описания стилей - это команды, определяющие свойства HTML документов (цвет, выравнивание). Набор свойств, доступный через CSS гораздо больше, чем стандартный HTML набор.

Синтаксис:

P {text-align:right; color:red}, где

P – класс HTML документов, которые будут подвергаться форматированию (то есть к которым будет применяться стиль).

Набор таких правил у нас может храниться в отдельном файле, могут находиться в контейнере дескриптора <STYLE>.

Например,

<P STYLE =”text-align:right”>

P.krass{color:red} – все параграфы класса krass будут красного цвета.

<P class=krass>…</P>

P:first-letter {color:red; font-size:300%; float:left}, где first-letter – имя псевдонима. (Здесь первая буква).

P.krass:first-letter

 <P class=krass>

A:visited {color:red; font-size:300%; float:left}

A:active

 first-line

**Псевдокласс** – это множество HTML элементов, описываемых непосредственно спецификацией CSS.

#111{color:red} ,где #111 –идентификатор элемента.

<P ID=111>

Атрибут ID служит для присваивания идентификатора элементу.

Пример:

<IMG ID=111 SRC=”…”>

Так можно накладывать элементы друг на друга:

#111{top:10; left:100; width:100; height:200; width:100; height:200; z-index:2}

Координата z определяет место (порядок) элемента.

**Каскадный приоритет** – это порядок, в котором правила, определяющие конкретные стили будут применяться.

 1) На первом месте правила, определенные атрибутом STYLE.

 2) Правила, определяемые с помощью стилей.

Например, P.1{…}.

 3) Правила, определяющие стили для дескрипторов определенного типа.

Например, P {text-align:right; color:red}

Подключение стилей с помощью дескриптора <STYLE> является более важным, чем подключение стилей с помощью дескриптора LINE.

 Чтобы придать акцент (приоритет) определенному правилу используется important.

Пример:

P {text-align:right !important}- искусственное изменение приоритета.

 4) На последнем месте каскадного приоритета (то есть наименьшие по важности) правила определения стилей указываемых пользователем с помощью браузера.

P, H1, DIV {…} – эта конструкция используется для одновременного присвоения свойств сразу нескольким классам.

# **XML - расширяемый язык разметки**

Расширяемый язык разметки XML предназначен для описания внутренней логической структуры документа.

<ТЕМА>информация</ТЕМА>

В качестве команд языка HML используются элементы. Элемент начинается так же, как и дескриптор:

<название элемента>содержимое элемента</название элемента>

!Элемент всегда заканчивается закрывающимся тэгом. Одиночных элементов (как в HTML) не бывает.

При указании элементов могут использоваться атрибуты (свойства) элемента. В отличие от атрибута HTML, значения атрибутов обязательно должны заключаться в кавычки.

**Иерархическая структура документов.**

Все элементы XML документа должны располагаться в строгой иерархической последовательности. Обязательно должен быть один корневой элемент, от которого иерархическое дерево растет и изветвляется.

Пример:

<?XML version=”1.0” encoding=”windows-1251”?>

<PREDMET\_LIST>

 <PREDMET>

<NAME>МИР</NAME>

<PREPOD>Выдрин А.Л.</ PREPOD >

<TEMY\_LIST>

<TEMA>Информация</TEMA>

< TEMA >Структура Интернета</ TEMA >

… … …

</ TEMY\_LIST >

 </ PREDMET >

</ PREDMET\_LIST >

Основное свойство HML – его расширяемость. Создатель документа может использовать для обозначения элементов любые названия, определяющие смысл элемента одинаково для него и пользователя.

Для определения смысла отдельных элементов HML документа используются специальные конструкции **DTD** – определители типа документов, которые определяют не только логический смысл элемента (дают комментарии по поводу описания использования элемента), но и жестко определяют структуру элемента.

Синтаксис данных определителей типа документов достаточно сложен и напоминает отдельный язык разметки (или язык программирования).

! Синтаксис достаточно строг. Нельзя делать ошибки.

DTD используются так же, как эталоны для проверки правильности XML документов.

Для задания внешнего представления XML документа используются стили. Основная нагрузка XML документа – содержание.

<?xml-stylesheet type=”text/css” ref=”1.css”?>

# **Классификация Web ресурсов**

|  |
| --- |
| Навигационные сайты |

|  |
| --- |
| Конечные (или функциональные) сайты |

|  |
| --- |
| Информационные(тематические)сайты |

|  |
| --- |
| Корпоративныесайты |

|  |
| --- |
| Каталоги |

|  |
| --- |
| Поисковыесистемы |

|  |
| --- |
| Сайтыэлектроннойкоммерции |

|  |
| --- |
| Сайты, являющиесяWeb интерфейсамисетей Internet |

|  |
| --- |
| Порталы |

**Навигационные сайты** – перенаправляют пользователей к конечным сайтам (указующая и направляющая роль). Например, порталы Яндекс, Кирилл и Мифодий и прочие.

**Портал** – это Web сайт, сочетающий в себе функции навигационного сайта и информационного ресурса по различным темам.

**Конечный (функциональный) сайт** – это сайты или ресурсы, которые содержат информацию или документы, ради которых пользователи приходят в глобальные сети (Internet).

**Информационные сайты** - обеспечивают доступ пользователей к документам определенной тематики.

**Корпоративные сайты**, а так же **сайты электронной коммерции** – дают доступ к коммерческой информации (информации о товарах, услугах, производителях), а также возможность удаленного заказа, оплаты и приобретения товаров и услуг.

Web интерфейс – это сайты, которые через стандартные HTML страницы (Web страницы) предоставляют доступ к сервисам Internet (электронной почты, телеконференций и другим).Роль последней группы сайтов очень велика.

Семинарские занятия

1) Знакомство с Internet (чтение учебника). Подключится к серверу.

2) Создание учетных записей. Создание оригинального сообщения.

3) Получить ваше сообщение. Создать правила для сообщений почты в отношении названных вами сообщений.

4) Получить ваше сообщение, создать ответ и отправить сообщение. Перенаправить ваше сообщение вам и еще кому-нибудь, присоединив к нему файл.

5) Подписаться на группу новостей.

6) Создание запросов. Найти в документах данные вами слова и фразы.

7) Создать HTML документ. В заголовке использовать дескрипторы <title>, <meta> (с атрибутами http-equiv, name, content). В теле опробовать действие атрибутов bgcolor, background, text, link, vlink, alink, topmargin, leftmargin.

8) Форматирование текста. Создать HTML документ. Скопировать текст. Попробовать действие дескрипторов <!>,<comment>, <br>, <nobr>, <wbr>, <p> (с атрибутом align), заголовка от <h1> до <h6> (с тем же атрибутом), <hr> (с атрибутамиalign, color, width, size, noshade),< font> (атрибут face, color, size),< basefont> (атрибуты как у font), <pre>(в его контейнере создать таблицу стандартными средствами форматирования), <div>, <center>.

9) Форматирование текста. Создать HTML документ. Скопировать текст. Попробовать действие дескрипторов <b>, <i>, *<*cite**>**, <blockquote>, <u>. Отобразить символы символов&lt (левая скобка), &gt (правая скобка), &copy (значок авторского права), &deg (значок температуры), &nbsp (жесткий пробел).

10) Сложное форматирование. Создать HTML документ. Создать нумерованный и маркерный списки. То есть попробовать действие дескрипторов <ol> (атрибуты type, start), <li> ,<ul> (атрибут type). Создать многоуровневый список.

11) Сложное форматирование. Создать нарисованную на доске таблицу. Поменять ее размеры, ширину рамки, фон, расстояние между ячейками, цвет ячеек и их границ, выравнивание в ячейках. Попробовать создать таблицу в ячейке данной таблицы.

12) Внедрение графического изображения. Создать HTML документ. Скопировать текст. В тексте поместить картинки, как показано на доске. Использовать дескриптор <img> (атрибуты src, width, height, hspace, vspace, border, align, alt).

13) Создание ссылок. Создать три HTML документа. Связать их между собой с помощью ссылок, предусмотреть возможность перемещения обратно. Используем дескриптор <A> атрибуты href, name.

14) Скачать с компьютера преподавателя картинки и расположить их так, чтобы получился лабиринт (смотри на доску).

15) HTML форма. Проверить действие дескрипторов <form> (атрибуты name, method, action, enctype), **<**textarea> (атрибуты name, rows, colls), <input>(атрибут name, и всевозможные значения атрибута type с соответствующими каждому элементу формы возможными атрибутами), <select>(атрибуты name, size, option)

16) HTML форма. Создать изображенную на доске форму заказа, заполнить ее и отправить ее на обработку в Web сайт. Проверить соответствие HTML кода и данных формы.

17) Листы каскадных стилей. Создать HTML документ. В него скопировать текст и поместить два рисунка (один на другой). Подключить стили всеми возможными способами. Поменять приоритет стилей.

18) Листы каскадных стилей. Создать HTML документ. В него скопировать текст создать правила, указанные на доске. Подключить стили. Поменять приоритет стилей.

19)XML. Создать XML документ, описывающий структуру HTML документа.