

**Департамент образования Вологодской области  
бюджетное профессиональное образовательное учреждение  
Вологодской области  
«ВОЛОГОДСКИЙ СТРОИТЕЛЬНЫЙ КОЛЛЕДЖ»**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**  
к практическим работам  
по МДК 02.04 Web-программирование  
ПМ.02. Участие в разработке информационных систем

---

09.02.04 Информационные системы (по отраслям)

2017

Рассмотрено на заседании предметно-цикловой комиссии общепрофессиональных, специальных дисциплин и дипломного проектирования по специальностям 08.02.01. Строительство и эксплуатация зданий и сооружений, 08.02.07. Монтаж и эксплуатация внутренних сантехнических устройств, кондиционирования воздуха и вентиляции, 43.02.08. Сервис домашнего и коммунального хозяйства

Данные методические указания предназначены для студентов специальности Информационные системы (по отраслям) БПОУ ВО «Вологодский строительный колледж» при выполнении практических работ по МДК 02.04. Web-программирование

Объем практических работ по МДК 02.04. Web-программирование составляет 50 часов.

Автор: Норинова С.В., преподаватель БПОУ ВО «Вологодский строительный колледж»

## Оглавление

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА .....	4
<b>ПЕРЕЧЕНЬ ПРАКТИЧЕСКИХ ЗАНЯТИЙ</b> .....	6
ПОРЯДОК ВЫПОЛНЕНИЯ ПРАКТИЧЕСКИХ РАБОТ .....	6
Тема 1. Язык гипертекстовой разметки HTML. ....	<b>Ошибка! Закладка не определена.</b>
Практическая работа №1 .....	6
Практическая работа №2 .....	9
Практическая работа № 3 «Создание таблиц» .....	11
Практическая работа №4 «Вставка в HTML-документ рисунков. Создание закладок и гиперссылок» .....	13
Практическая работа № 5 «Создание форм в HTML-документе».....	14
Практическая работа № 6 «Фреймы» .....	19
Практическая работа №7 «Алгоритм разработки сайта» .....	21
Тема 2. Каскадные таблицы стилей CSS. ....	<b>Ошибка! Закладка не определена.</b>
Практическая работа №8 «Создание навигационной карты» .....	25
Практическая работа №9 «Способы подключений CSS».....	26
Практическая работа №10 «Редактирование html-страницы с применением CSS».....	27
Практическая работа №11 «Верстка макетов с помощью CSS».....	31
Тема 3. Язык сценариев JavaScript.....	<b>Ошибка! Закладка не определена.</b>
Практическая работа №12 «Динамичные эффекты посредством CSS» .....	33
Практическая работа №13 «Размещение кода JavaScript. Основы JavaScript» .....	36
Практическая работа №14 «События и Функции в JavaScript».....	39
Практическая работа №15 «Создание сценариев с использованием встроенного объекта Date».....	42
Практическая работа №16 «Массивы».....	43
Практическая работа №17 «Встроенный объект String».....	45
Практическая работа №18 «Регулярные выражения» .....	46
Тема 4. Основы работы в PHP .....	<b>Ошибка! Закладка не определена.</b>
Практическая работа № 19 «Проверка заполнения форм» .....	48
Практическая работа № 20 «Работа с файлами в PHP».....	50
Практическая работа №21 «Сохранение данных из формы в базе данных MySQL».....	52
Практическая работа №22 «Знакомство с интерфейсом редактора Flash CS3. Рисование»	52
Практическая работа №23 «Требования к иллюстрациям в Internet и подготовка графики для Веб в Photoshop» .....	55
Практическая работа №24. Выбор доменного имени. Хостинг. Выбор хостинга .....	59
Практическая работа №25. Перенос сайта с помощью FTP-клиента на хостинг .....	63

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

МДК 02.04 Web-программирование является вариативной частью учебных циклов в соответствии с ФГОС и преподается студентам специальности 09.02.04 Информационные системы (по отраслям).

В результате освоения ПМ.02. Участие в разработке информационных систем обучающийся должен **иметь практический опыт:**

- использования инструментальных средств обработки информации;
- участия в разработке технического задания;
- формирования отчетной документации по результатам работ;
- использования стандартов при оформлении программной документации;
- программирования в соответствии с требованиями технического задания;
- использования критериев оценки качества и надежности функционирования информационных систем;
- применения методики тестирования разрабатываемых приложений;
- управления процессом разработки приложений с использованием инструментальных средств;

### **уметь:**

- осуществлять математическую и информационную постановку задач по обработке информации, использовать алгоритмы обработки информации для различных приложений;
- уметь решать прикладные вопросы интеллектуальных систем с использованием, статических экспертных систем, экспертных систем реального времени;
- использовать языки структурного, объектно-ориентированного программирования и языка сценариев для создания независимых программ, разрабатывать графический интерфейс приложения;
- создавать проект по разработке приложения и формулировать его задачи, выполнять управление проектом с использованием инструментальных средств;

### **знать:**

- основные виды и процедуры обработки информации, модели и методы решения задач обработки информации (генерация отчетов, поддержка принятия решений, анализ данных, искусственный интеллект, обработка изображений);
- сервисно - ориентированные архитектуры, CRM-системы, ERP-системы;
- объектно-ориентированное программирование;
- спецификации языка, создание графического пользовательского интерфейса (GUI), файловый ввод-вывод, создание сетевого сервера и сетевого клиента;
- платформы для создания, исполнения и управления информационной системой;
- основные процессы управления проектом разработки.

В соответствии с учебным планом на изучение МДК 02.04 Web-программирование отводится 108 часов, в том числе 50 часов – практические занятия.

Целью практических занятий является формирование практических умений, необходимых в последующей учебной и профессиональной деятельности.

Содержание практических занятий по МДК 02.04 Web-программирование направлено на реализацию требований Федерального государственного образовательного стандарта по специальности СПО 09.02.04 Информационные системы (по отраслям).

Практическое занятие включает следующие структурные элементы:

- 1) инструктаж, проводимый преподавателем,
- 2) самостоятельная деятельность обучающихся,

3) анализ и оценка выполненных работ.

Контроль и оценка результатов выполнения обучающимися работ, заданий на практических занятиях направлены на проверку освоения умений, практического опыта, развития общих и формирование профессиональных компетенций, определённых программой учебной дисциплины.

Оценки за выполнение заданий на практических занятиях выставляются по пятибалльной системе и учитываются как показатели текущей успеваемости обучающихся.

**ПЕРЕЧЕНЬ ПРАКТИЧЕСКИХ ЗАНЯТИЙ  
МДК 02.04. WEB-ПРОГРАММИРОВАНИЕ**

<b>№ п/п</b>	<b>Тема программы</b>	<b>Тема работы</b>	<b>Количество часов</b>
1.	<i>Тема 1. Интернет-технологии. Инструменты создания web - сайтов и приложений.</i>	Проектирование сайта (концептуальное, логическое и физическое).	2
2.	<i>Тема 2. Язык гипертекстовой разметки HTML</i>	Основы работы в HTML. Форматирования шрифта и абзаца	2
3.		Создание таблиц	2
4.		Вставка в HTML-документ рисунков. Создание закладок и гиперссылок	2
5.		Создание форм в HTML-документе	2
6.		Фреймы	2
7.		Алгоритм разработки сайта	2
8.		<i>Тема3. Каскадные таблицы стилей CSS</i>	Создание навигационной карты
9.	Способы подключений CSS		2
10.	Редактирование html-страницы с применением css		2
11.	Верстка макетов с помощью CSS		2
12.	<i>Тема4. Язык сценариев JavaScript</i>	Динамические эффекты посредством CSS	2
13.		Размещение кода JavaScript. Основы JavaScript	2
14.		События и Функции в JavaScript	2
15.		Создание сценариев с использованием встроенного объекта Date	2
16.		Массивы	2
17.		Встроенный объект String	2
18.		Регулярные выражения	2
19.	<i>Тема 5. Основы работы в PHP</i>	Проверка заполнения форм	2
20.		Работа с файлами в PHP	2
21.		Сохранение данных из формы в базе данных MySQL	2
22.	<i>Тема 6. Подготовка графики для web</i>	Знакомство с интерфейсом редактора Flash CS3. Рисование	2
23.		Требования к иллюстрациям в Internet и подготовка графики для Веб в Photoshop	2
24.	<i>Тема 7. Публикация сайта в сети Интернет</i>	Выбор доменного имени. Хостинг. Выбор хостинга.	2
25.		Перенос сайта с помощью FTP-клиента на хостинг.	2

**ПОРЯДОК ВЫПОЛНЕНИЯ ПРАКТИЧЕСКИХ РАБОТ**

**Практическая работа №1**

**Проектирование сайта (концептуальное, логическое и физическое).**

Существует три уровня моделирования данных:

- Концептуальный -► концептуальная модель
- Логический -► логическая модель
- Физический -► физическая модель

### **Концептуальное проектирование**

Порой бывает сложно оценить эффективность сайта. Действительно, какими критериями воспользоваться, чтобы понять, что сайт, над которым столько корпели и тратили на него свое время, действительно хорош. Разумеется, можно использовать количество посетителей как некоторое мерило. Но этот показатель относителен и зависит от множества внешних параметров. То же самое обстоит и с другими способами оценки сайта.

Есть однако универсальный критерий, который довольно точно характеризует эффективность сайта. Это достижение разработчиками сайта поставленных перед ними целей. В этом случае сайт превращается в действенный инструмент, который выполняет возложенные на него функции.

Концептуальное проектирование служит для указания целей, задач сайта и определения аудитории, на которую он рассчитан.

#### **На этом этапе следует описать следующее:**

1. Основные и второстепенные цели.
2. Действия, которые необходимо предпринять для достижения поставленных целей.
3. Состав пользователей.
4. Интересы групп пользователей.
5. Разделы сайта.
6. Критерии достижения цели.

С учетом поставленных целей, а также интересов пользователей, в итоге получаем список сервисов и разделов, которые будут располагаться на сайте.

### **Логическое проектирование**

Разделы сайта, продуманные на предыдущем этапе, пока не упорядочены и не структурированы, поэтому их нужно привести к удобному и понятному виду. Логическое проектирование включает организацию информации на сайте, построение его структуры и навигации по разделам.

На данном этапе следует задаться вопросом, каким образом будет упорядочена информация. Варианты могут быть самыми разными и зависеть от типа данных и предпочтений создателей сайта: по времени, разделам, в алфавитном порядке, определенным группам или другим критериям. Так, для сайта музыкальной группы, поиск определенной песни можно сделать в виде алфавитного указателя, по названию альбома, первым строчкам песни, году выпуска и по ключевым словам. Одновременное использование различных способов охватывает большую аудиторию и позволяет быстрее найти нужную информацию на сайте.

#### **На этом этапе следует описать следующее:**

1. Тип структуры сайта (линейная, иерархическая, контекстная, другая).
2. Названия разделов.
3. Что будет содержать в себе каждый раздел.
4. Организация и связь разделов между собой.
5. Какая информация будет размещена на определенных страницах сайта.

Конечный результат логического проектирования оформляется в виде блок-схем, структурных диаграмм или другими способами, показывающими взаимосвязь различных частей сайта.

### **Физическое проектирование**

Этап поиска проблем, а не их решений, связанных, по большей части, с технической реализацией сайта.

#### **На этом этапе следует описать следующее:**

1. Технологии, которые будут применяться на сайте.
2. Используемое программное обеспечение.
3. Возможные проблемы и способы их устранения.
4. Как будет обновляться информация.

После завершения данного этапа следует вернуться к концептуальному проектированию и проверить, не нужно ли внести изменения, в связи с переосмыслением проекта на других стадиях.

Так, если на сайте планируется использовать базы данных и доступ к данным с помощью CGI, следует подумать о хостинге, который поддерживает выбранные технологии. Возвращаясь к концептуальному проектированию, мы либо ставим себе цель разместить сайт на платном хостинге, затратив на это определенную сумму денег, при этом получая взамен дополнительные возможности. Либо зажав наши кровные окунуться в мир бесплатного хостинга, в просторечье называемым бомжатником. При этом часть возможностей просто теряется.

**Задание.** На примере разработки сайта школы спроектировать концептуальную, логическую и физическую модель сайта.

#### **Пример концептуальной модели:**

##### **Цели**

1. Качественная подготовка выпускника;
2. Сделать школу открытой;
3. Повышение имиджа школы.

##### **Пользователи:**

Родитель

Ученик

Учитель

##### **Сценарии пользователей**

здесь описаны разделы сайта, интересные данной группе пользователей

Родитель:

- Информация об успеваемости учеников
- Информация о платных/бесплатных услугах
- Рейтинг среди других школ
- Резюме учителей и руководителей
- Информацию о страховании
- Телефоны, адреса, реквизиты школы
- Информация о системе непрерывного образования

Ученик:

- С какими вузами взаимодействует школа
- Расписание занятий
- Расписание мероприятий
- Информация об выпускниках
- Система дополнительного образования



Учитель:

- Расписание занятий
- План мероприятий
- График работы аттестационной комиссии
- Методические разработки
- Перечень методической литературы
- Информация о повышении квалификации
- Информация о юбилеях, поздравления
- Информация о грантах и финансовых потоках

## Практическая работа №2

### «Основы работы в HTML. Форматирования шрифта и абзаца»

**Цель работы:** Научиться создавать простейший гипертекстовый документ средствами текстового редактора Notepad++. Научиться использовать теги форматирования шрифта и абзаца.

**Задание 1.** Создать HTML-документ.

*Указания к выполнению*

1. Создайте папку (в качестве имени папки выберите свою фамилию). Запустите редактор Notepad++. Ввести приведенные ниже стандартные теги разметки страницы теги

```
<HTML><HEAD><TITLE>Мой первый HTML-документ</TITLE></HEAD>  
<BODY>ПРИВЕТ МИР!!!</BODY></HTML>
```

2. Сохранить документ под именем 1.html в созданной вами папке. Откройте папку, кликните правой кнопкой мыши по файлу 1.html и выберите команду «Открыть с помощью» и выберите удобный для просмотра браузер. В окне браузера вы можете посмотреть как будет выглядеть ваш файл.

3. Отредактируйте документ 1.html, кликните правой кнопкой мыши и выберите команду «Открыть с помощью» и выберите программу Блокнот. После слов ПРИВЕТ МИР!!! поставьте тег <BR> (тег перехода на новую строку), и наберите текст «Программирование сайтов и Web-дизайн». Не забывайте сохранять документ и обновлять браузер.

4. Втег<BODY> вставить атрибуты TEXT(цвет текста на всейhtml-странице) и BGCOLOR(цвет заливки на всейhtml-странице), и установите любое значения цветов (безопасные коды цветов можно найти в интернете). Поставьте горизонтальную линию в документе с помощью соответствующего тега <HR SIZE=1>, где атрибут SIZE определяет толщину линии в пикселях.

**Задание 2.** Создайте заголовки по образцу

Анкета  
Петров Владимир Владимирович  
17 лет  
козерог  
студент  
люблю рок-музыку

*Указания к выполнению*

Откройте в Notepad++ файл 1.html и создайте анкету с помощью заголовочных тегов <H1-6 ALIGN="LEFT| CENTER| RIGHT"> текст </H1-6>, с помощью атрибута ALIGN выравните текст по центру.

**Задание 3.** Создайте текст на HTML-странице по образцу

ПАРУС

Белеет парус одинокой

В тумане моря голубом!..

Что ищет он в стране далекой?

Что кинул он в краю родном?..

Играют волны — ветер свищет,

И мачта гнется и скрипит...

Увы! он счастья не ищет,

И не от счастья бежит!

Под ним струя светлей лазури,

Над ним луч солнца золотой...

А он, мятежный, просит бури,

Как будто в бурях есть покой!

*Указания к выполнению*

В файле 1.html после анкеты наберите текст стихотворения «Парус» и отформатируйте шрифт текста по образцу. Название стихотворения заголовком 3-го уровня (<H3></H3>), первая строчка стихотворения выделена жирным шрифтом с помощью тегов <B></B> или <STRONG></STRONG>; вторая строчка выделена курсивом с помощью тега <I></I>; третья строчка подчеркнута с помощью тега <U></U>; четвертая строчка перечеркнута с помощью тега <S></S> или <DEL></DEL>; пятая, шестая, седьмая и восьмая строчки выделены с помощью тегов с размерами шрифтов <FONTSIZE=1|3|5|7></FONT> соответственно; девятая строчка выделена с помощью тега <SMALL></SMALL>; десятая строчка создана при помощи тега <BIG></BIG>; одиннадцатая и двенадцатая строчки раскрашены с помощью тегов <FONTCOLOR=YELLOW|GREEN></FONT> соответственно. В конце стихотворения добавьте фамилию и инициалы автора стихотворения моноширинным шрифтом(шрифт имитирующий печатную машинку), с помощью контейнеров тегов <TT></TT>,<KBD></KBD>или<SAMP></SAMP>.

**Задание 4.** Создание формулу по образцу  $S_{\text{квадрата}}=a^2$

*Указания к выполнению*

Для создания нижнего индекса используйте теги <SUB></SUB> в контейнер заключите индекс, для возведения в степень используйте теги верхнего индекса <SUP></SUP>. Увеличьте шрифт не индексного текста до 5.

**Задание 5.** Создание абзацев.

*Указания к выполнению*

Скопируйте любой текст и вставьте в файл 1.html после формулы, отформатируйте абзацы с помощью тега <P></P> для выравнивания абзаца используйте атрибут ALIGN=LEFT|CENTER|RIGHT|JUSTIFY.

**Задание 6.** Создание списков.

*Указания к выполнению*

В файле 1.html после отредактированного текста создайте списки.

Для создания нумерованного списка используют теги <OL><LI>.

<OL><LI>Элемент списка 1 </LI>

<LI>Элемент списка 2</LI>

<LI>Элемент списка 3</LI></OL>

Список по умолчанию цифровой, но его можно сделать и буквенным с помощью атрибута TYPE .

<OLTYPE=a><LI>Элемент списка 1 </LI>

<LI>Элемент списка 2</LI>

<LI>Элемент списка 3</LI></OL>

Для создания маркированного списка используют теги <UL><LI>.

```
<UL><LI>Элемент списка 1 </LI>
```

```
<LI>Элемент списка 2</LI>
```

```
<LI>Элемент списка 3</LI></UL>
```

Маркеры в списке можно сделать разнообразными с помощью атрибута

TYPE=CIRCLE|DISC|SQUARE .

```
<UL><LI>Элемент списка 1 </LI>
```

```
<LI>Элемент списка 2</LI>
```

```
<LI>Элемент списка 3</LI></UL>
```

Для создания списков определений используют три тэга:

<DL>- начало/конец списка.

<DT>- начало/конец конкретного термина.

<DD>- начало/конец поясняющей статьи термина.

```
<DL><DT>ГРИБЫ</DT>
```

```
<DD>ОПЯТА</DD>
```

```
<DD>ГРУЗДИ</DD></DL>
```

### Самостоятельная работа.

а) Создайте формулу соляной кислоты и общий вид квадратного уравнения.

б) Создайте смешанный список по образцу

Я знаю как оформлять:

1. Шрифты
  - Размер
  - Цвет
  - Гарнитуру
  - Индексы
2. Заголовки
  - От 1-го до 6-го уровня
3. Абзацы
  - Выравнивание
  - Разрыв строк внутри абзаца
  - С использованием переформатирования.

Составить отчет о выполнении

### Задание 7. Создание аббревиатуры

Указания к выполнению

Для создания аббревиатуры можно использовать либо контейнер тегов <ABBR></ABBR>, либо <ACRONYM></ACRONYM>, разница между ними лишь в том что <ACRONYM></ACRONYM> это элементы только спецификации HTML4.01, А ТЕГИ <ABBR></ABBR> спецификаций HTML4.01/5.0 и расшифровать аббревиатуры можно с помощью атрибута TITLE, например:

```
<ABBR TITLE="Hyper Text Markup Language">HTML</ABBR><BR>  
<ACRONYM>HTML</ACRONYM>.
```

## Практическая работа № 3

### «Создание таблиц»

**Цель работы:** Изучение тегов <TABLE>, <TH>, <TD> и их атрибутов приобретение умения задания таблицы, ее размера, шрифта записей в ячейках таблицы, расположения и вида таблицы.

**Задание 1.** Создание таблицы.

### Указания к выполнению

Откройте Notepad++ и наберите следующий текст

```
<HTML>
<TITLE>таблицав HTML</TITLE><HEAD></HEAD>
<BODY ><H3 ALIGN=CENTER>ТАБЛИЦА № 1</H3>
```

1. Для создания таблицы используется тег <TABLE>, атрибуты данного тега определяют:

ALIGN- выравнивание таблицы

BACKGROUND- Задаёт фоновый рисунок в таблице

BGCOLOR - Цвет фона таблицы

BORDER - Толщина рамки в пикселях

BORDERCOLOR - Цвет рамки

CELLPADDING - Отступ от рамки до содержимого ячейки

CELLSPACING - Расстояние между ячейками

COLS - Число колонок в таблице

FRAME - Сообщает браузеру, как отображать границы вокруг таблицы

HEIGHT - Высота таблицы

RULES - Сообщает браузеру, где отображать границы между ячейками

SUMMARY - Краткое описание таблицы

WIDTH - Ширина таблицы

```
<TABLE ALIGN=CENTER BORDER="5" BGCOLOR =CCCC00 CELLPADDING =1
CELLSPACING=1
```

```
BORDERCOLOR=GOLD><CAPTION>ТАБЛИЦА ИСТИННОСТИ</CAPTION>
```

2. Теги <CAPTION> определяет название таблицы.

Тег <THEAD></THEAD> определяет верхнюю часть таблицы. Теги <TR></TR> определяет табличный ряд. Теги <TH></TH> определяют заголовок таблицы.

Теги <TD></TD> определяют ячейки таблицы. Теги <TBODY></TBODY> определяют тело таблицы

```
<THEAD><TR><TH><B>A</B></TH><TH><B>B</B></TH><TH><B>A and
```

```
B</B></TH><TH><B>A or B</B></TH><TH><B>not A</B></TH><TH><B>not B
```

```
</B></TH></TR></THEAD>
```

```
<TBODY><TD>0</TD><TD>0</TD><TD>0</TD><TD>0</TD><TD>1</TD><TD>1</TD>
</TR>
```

```
<TR><TD>0</TD><TD>1</TD><TD>0</TD><TD>1</TD><TD>1</TD><TD>0</TD></TR>
>
```

```
<TR><TD>1</TD><TD>0</TD><TD>0</TD><TD>1</TD><TD>0</TD><TD>1</TD></TR>
>
```

```
<TR><TD>1</TD><TD>1</TD><TD>1</TD><TD>1</TD><TD>0</TD><TD>0</TD></TR>
>
```

```
</TBODY></TABLE></BODY></HTML>
```

3. Сохраните полученный файл как 2. html в вашей папке, откройте его с помощью браузера, в результате вы должны получить следующую таблицу. Количество тегов <TD></TD> должно соответствовать количеству ячеек с данными в таблице, а количество строк равно количеству тегов <TR></TR>. Содержимое ячеек заключено в тегах <TD></TD>.

ТАБЛИЦА № 1

ТАБЛИЦА ИСТИННОСТИ					
A	B	A and B	A or B	not A	not B
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	1
1	1	1	1	0	0

**Задание 2.** Создать таблицу по образцу

ТАБЛИЦА № 2

Густонаселенные города			
Город	Население	Страна	Координаты
Буэнос-Айрес	12 955 300	Аргентина	34° 37' ю.ш. 58° 22' з.д.
Бомбей	12 147 100	Индия	18° 58' с.ш. 72° 49' в.д.
Дели	10 009 200		28° 40' с.ш. 77° 13' в.д.
Сеул	11 153 200	Корея Южная	37° 34' с.ш. 126° 59' в.д.
Карачи	10 272 500	Пакистан	24° 52' с.ш. 67° 1' в.д.
Манила	10 133 200	Филиппины	14° 37' с.ш. 120° 58' в.д.
Сан-Паулу	10 057 700	Бразилия	23° 32' ю.ш. 46° 38' з.д.
Рио-де-Жанейро	6 029 300		22° 55' ю.ш. 43° 12' з.д.

*Указания к выполнению*

1. Откройте файл 2. html в Notepad++ , создайте таблицу № 2 из девяти строчек и четырех столбцов. Обратите внимание, что в 3-м столбце третья и четвертая строчки объединены в одну ячейку. Объединение ячеек производится с помощью атрибутов ROWSPAN и COLSPAN тегов <TD></TD>. В конкретном случае объединяются две строки, поэтому мы используем атрибут <TD ROWSPAN=2>, если нужно объединить три столбца, то можно использовать следующий атрибут тега <TD COLSPAN=3>. Ячейки, участвующие в объединении на последующих строчках нужно удалить.

2. Чтобы поменять цвет и шрифт текста в ячейках нужно теги <FONT FACE="ТАНОМА" SIZE="5" COLOR="CYAN"></FONT> вставить в контейнер тегов <TD></TD>.

**Самостоятельная работа.** Создайте таблицу вашего расписания на неделю. Составить отчет о выполнении.

#### Практическая работа №4

##### «Вставка в HTML-документ рисунков. Создание закладок и гиперссылок»

**Цель работы:** Научиться выполнять вставку рисунков в HTML-документ. Научиться создавать закладки и гиперссылки

**Задание 1.** Вставить рисунок в HTML-документ

*Указания к выполнению*

1. Скопировать из какой-либо папки или скачать с Интернета в личную папку три графических файла. Откройте Блокнот и наберите стандартные теги <HTML><TITLE>рисунки и </HTML></TITLE><HEAD></HEAD><BODY>сохраните файл как 3.html.

2. Для вставки использовать тег `IMG`, с параметрами `WIDTH` и `HEIGHT` для установки размеров рисунка 500 пикселей по горизонтали и по вертикали и основной атрибут `SRC`, определяющий какой рисунок будет изображаться в HTML-документе. С помощью параметра `ALT` создать всплывающую подсказку Рисунок 1, появляющуюся при наведении курсора мыши на рисунок на пример `<IMG SRC="ШАР.JPG" WIDTH=500 HEIGHT=500 ALT="рисунок 1">`. Посмотрите результат в браузере (Внимание! Не во всех браузерах вид HTML-документа будет выглядеть одинаково).

**Задание 2.** Создание гиперссылки.

Вставка ссылок		
Ссылки на другую страницу	<code>&lt;A HREF=" страница"&gt; текст &lt;/A&gt;</code>	<code>&lt;A HREF="str2.htm"&gt; Ссылка1 &lt;/A&gt;</code>
Ссылка на закладку в другом документе	<code>&lt;A HREF=" страница # имя закладки"&gt; текст &lt;/A&gt;</code>	<code>&lt;A HREF=" index.htm #met1"&gt; На главную страницу &lt;/A&gt;</code>
Ссылка на закладку в том же документе	<code>&lt;A HREF="#имя закладки"&gt; текст &lt;/A&gt;</code>	<code>&lt;A HREF=" #metka1"&gt; Ссылка2 &lt;/A&gt;</code>
Определить закладку	<code>&lt;A NAME="имя закладки"&gt; текст &lt;/A&gt;</code>	<code>&lt;A NAME="metka1"&gt; &lt;/A&gt;</code>

1. Создадим в файле 3.html ссылку на файл 2.html для этого используем тег `<A></A>`, т.е. `<A HREF="3.html">Таблицы </A>`, где атрибут `HREF` указывает какой HTML-документ будет открываться при нажатии на ссылку. Посмотрите результат в браузере.

2. Для создания ссылки на закладку, откройте в блокноте файл 1.html, в начале документа установите закладку на слово «АНКЕТА», т.е. `<a name = "metka1" >АНКЕТА </a>` а затем конце документа создайте ссылку на эту закладку `<A href="# metka1">Перейти на Анкету </A>`. Посмотрите результат в браузере.

3. Для создания ссылки на закладку в др. файле, откройте в блокноте файл 2.html, установите закладку на слово «ТАБЛИЦА № 2», т.е. `<a name = " metka2" > ТАБЛИЦА № 2 </a>`, а затем в конце документа 3.html создайте ссылку на эту закладку `<A href="2.html# metka2">Перейти на Таблицу №2 </A>`.

4. Создадим ссылку на рисунок. Переименуйте любой из скаченных графических файл в вашей папке в картинка.jpg, а затем в конце файла 3.html создайте ссылку `<A HREF="картинка.jpg ">картинка </A>`. Посмотрите, как это выглядит в браузере.

5. Создание картинки в качестве гиперссылки. Переименуйте последний оставшийся графический файл в папке в ссылку.jpg и откройте в блокноте файл 3.html. Наша картинка будет ссылкой на официальный сайт Рамблер `<A HREF="http://www.rambler.ru/"></A>`. Обратите внимание, что тег `<IMG>` заключена в контейнере тегов `<A></A>`, таким образом, в качестве ссылки может быть и видеофайл и анимация, если вы заключите их в контейнер тегов `<A></A>`.

Самостоятельная работа. Скачайте с Интернета картинку парусника, вставьте картинку в HTML-документ, и сделайте из него ссылку на слово ПАРУС в файле 1.html.

### Практическая работа № 5

#### «Создание форм в HTML-документе»

**Цель работы:** формирование умений создание форм с помощью HTML тегов.

**Задание 1.** Создание простой формы.

Для создания html-формы используется тег `<FORM>`. У него могут быть следующие атрибуты:

`NAME` - определяет имя формы, обычно не указывается. Применяется для идентификации формы, если в документе присутствует несколько форм.

ACTION - обязательный параметр, он задает путь к скрипту, который будет запущен веб-сервером.

METHOD - определяет способ отправки параметров формы. Принимает значение GET или POST.

TARGET - определяет окно, в которое возвращается результат обработки отправленной формы.

Последние три атрибута понадобятся, когда мы будем проходить серверный скрипт PHP.

Следующим основным компонентом формы является тег <INPUT>, атрибут TYPE которого, определяет поля формы. Помимо этого атрибута у тега <INPUT>, существуют следующие:

SIZE="25" – длина текстового поля в символах.

MAXLENGTH="30" – максимально допустимое количество вводимых символов.

VALUE="" – определяет значение, которое будет отправлено на обработку

*Указания к выполнению*

1. Откройте Блокнот и наберите стандартный набор тегов

<HTML><TITLE>формаHTML</TITLE><HEAD></HEAD><BODY></BODY>сохраните файл как 4.html. между тегами <BODY></BODY>наберите следующий текст:

```
<FORM NAME=FORM1 METHOD=POST ACTION=SHOW.PHP>
```

```
<INPUT TYPE=TEXT NAME=TEXT VALUE="YOUR NAME" SIZE=15>
```

```
<INPUT TYPE=SUBMIT VALUE=ОПУБЛИКОВАТЬ>
```

```
</FORM>
```

В браузере у вас должно появиться текстовое поле и кнопка, в текстовом поле будет значение по умолчанию YOURNAME, которое можно удалить и подставить другое значение.

2. Установим переключательные поля, которые бывают двух видов: флажки и радиокнопки.

Чтобы подключить флажок, используем тег <INPUT>со значением атрибутаTYPE="CHECKBOX", поставьте тег <BR>после текстового поля (<INPUTTYPE=TEXTNAME=TEXTVALUE="YOURNAME" SIZE=15>) и введите следующий текст:

```
ПОЛ:<INPUT TYPE=CHECKBOX NAME=MEN VALUE=M CHECKED>муж<INPUT  
TYPE=CHECKBOX NAME=WOMAN VALUE=W>жен<BR>
```

посмотрите результат в браузере. ЗначениеCHECKED устанавливает флажок по умолчанию. Минус этого вида переключателей, в том что вы можете установить флажок сразу на муж.и жен. пол одновременно, чтобы этого избежать используем другой вид переключателей – радиокнопка. После полей с флажками вставьте следующий текст:

```
ПОЛ: <INPUTTYPE=RADIONAME=POLVALUE=MCHECKED> муж  
<INPUTTYPE=RADIONAME=POLVALUE=W>жен<BR>
```

Обратите внимание, что значение атрибута NAME в обоих полях одинаковое. В браузере вы не сможете установить переключатели сразу на два значения.

3. Для ввода многострочного текста, нужно использовать теги <TEXTAREA></TEXTAREA>. После полей с переключателями введите следующий текст:

```
<TEXTAREA NAME=COMMENT COLS=25 ROWS=8  
MAXLENGTH=100></TEXTAREA><BR>
```

атрибуты COLS и ROWS устанавливают ширину в символах и высоту в строках поля соответственно, MAXLENGTH устанавливает максимальную длину строки ввода. Если нужно заблокировать поле или установить режим поля для чтения воспользуйтесь атрибутами DISABLED и READONLY соответственно. Посмотрите, как это выглядит в браузере.

4. Введем поле для пароля (после многострочного текста), с помощью тега <INPUT>с атрибутом TYPE=PASSWORD

```
<INPUT TYPE=PASSWORDNAME=PASS SIZE=20 MAXLENGHT=12><BR>
```

где атрибуты SIZE и MAXLENGHT – ширина поля и максимальное число символов, которое может ввести пользователь соответственно.

5. Создание кнопки. Кнопку на веб-странице можно создать двумя способами — с помощью тега <INPUT TYPE=BUTTON> и тегов <BUTTON></BUTTON>.

После поля с паролем введите следующий текст

```
<INPUTTYPE=BUTTONNAME=КНОПКАVALUE="ЖМИ"><BR>
```

- создание обычной кнопки. В кнопку, создаваемую с помощью тегов <BUTTON></BUTTON>, можно разместить любой элемент HTML, например картинку, лишь заключив элемент в теги:

```
<BUTTON>  
<IMG SRC=lara.gif WIDTH=20 HEIGHT=20 ALIGN=CENTER >  
ЖМИ  
</BUTTON><BR>
```

Для отчистки и отправки формы можно использовать и тег <INPUT> и теги <BUTTON></BUTTON>с атрибутамиTYPE=RESET иTYPE=SUBMIT соответственно.

6. Создание списка. Список создается с помощью тегов <SELECT></SELECT> и <OPTION></OPTION>, где последние определяют элемент списка.

После кнопки с рисунком создадим список множественного выбора:

```
<P> ВАШЕ ЛЮБИМОЕ ВРЕМЯ ГОДА</P>  
<SELECT NAME="SPISOK" MULTIPLE>  
<OPTION VALUE="Spring">весна</OPTION>  
<OPTION VALUE="Summer" SELECTED >лето</OPTION>  
<OPTION VALUE="Autumn">осень</OPTION>  
<OPTION VALUE="Winter">зима</OPTION>  
</SELECT>
```

Атрибут MULTIPLE определяет список с множеством значений, атрибут SELECTED определяет значение по умолчанию, а атрибут VALUE определяет значение списка, которое будет отправлено на сервер.

Раскрывающийся список

```
<P> ВАШЕ ЛЮБИМОЕ ВРЕМЯ ГОДА</P>  
<SELECT NAME="SPISOK1" SIZE=1>  
<OPTION VALUE="Spring">весна</OPTION>  
<OPTION VALUE="Summer" SELECTED>лето</OPTION>  
<OPTION VALUE="Autumn">осень</OPTION>  
<OPTION VALUE="Winter">зима</OPTION>  
</SELECT>
```

Обратите внимание на значение атрибута SIZE=1, если взять больше 1 то список уже не будет раскрывающимся.

Группирование списка. Объединения значений списка на подгруппы создают теги <OPTGROUP></OPTGROUP>с атрибутом LABEL.

```
<P>ЛЮБИМЫЙМЕСЯЦ</P>  
    <SELECT NAME="SPISOK2"SIZE=1>  
<OPTGROUP LABEL ="ВЕСНА">  
<OPTION VALUE="3">МАРТ</OPTION>  
<OPTION VALUE="4">АПРЕЛЬ</OPTION>  
<OPTION VALUE="5">МАЙ</OPTION>
```



```

</OPTGROUP>
<OPTGROUP LABEL="ЛЕТО">
<OPTION VALUE="6">ИЮНЬ</OPTION>
<OPTION VALUE="7">ИЮЛЬ</OPTION>
<OPTION VALUE="8">АВГУСТ</OPTION>
</OPTGROUP>
  <OPTGROUP LABEL="ОСЕНЬ">
<OPTION VALUE="9">СЕНТЯБРЬ</OPTION>
<OPTION VALUE="10">ОКТЯБРЬ</OPTION>
<OPTION VALUE="11">НОЯБРЬ</OPTION>
</OPTGROUP>
  <OPTGROUP LABEL="ЗИМА">
<OPTION VALUE="12">ДЕКАБРЬ</OPTION>
<OPTION VALUE="1">ЯНВАРЬ</OPTION>
<OPTION VALUE="2">ФЕВРАЛЬ</OPTION>
</OPTGROUP>
</SELECT>

```

7. Скрытое поле используется для передачи технической информации на сервер и не отображается на странице.

```

<INPUT TYPE=HIDDEN NAME=SECRET VALUE=учебнаяформа>

```

В результате вы должны получить следующую форму:

YOUR NAME

ПОЛ:  муж  жен

ПОЛ:  муж  жен

.....

ВАШЕ ЛЮБИМОЕ ВРЕМЯ ГОДА

весна ▲

лето

осень ▼

зима ▼

ВАШЕ ЛЮБИМОЕ ВРЕМЯ ГОДА

лето ▼

ЛЮБИМЫЙ МЕСЯЦ

МАРТ ▼

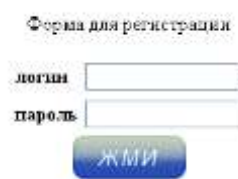
## Задание 2. Редактирование форм.

### Указания к выполнению

Для того чтобы форма имела четкую структуру, поля этой формы нужно заключить в ячейки таблицы. К примеру, форма для регистрации в таблице может выглядеть следующим образом:

```
<FORM NAME=FORM2>
<TABLE ALIGN=CENTER>
<TR><TD><B>логин</B></TD><TD><INPUT TYPE=TEXT NAME=TEXT
SIZE=15></TD></TR>
<TR><TD><B>пароль</B></TD><TD><INPUT TYPE="PASSWORD" NAME=PASS
SIZE=15 MAXLENGHT=12></TD></TR>
<TR><TD COLSPAN=2 ><INPUT TYPE=IMAGE NAME=RIC SRC=кнопка.gif WIDTH=90
HEIGHT=35 ALIGN=CENTER ></TD></TR>
</TABLE></FORM>
```

В этой таблице 2 столбца и 3 строчки, причем в последней, ячейки объединены в одну в которой находится поле, определяющее кнопку в виде рисунка и по действию аналогичную кнопке TYPE=SUBMIT для отправки формы.



Такую кнопку можно сделать в Photoshop, с прозрачным фоном и сохранить с расширением GIF.

## Задание 3. Создание допустимого списка.

### Указания к выполнению

Список допустимых значений определяют комбинация тегов <INPUT> И <DATALIST>. После формы регистрации создадим новую форму для голосования, в которой будет два поля: поле допустимых значений и кнопка «Голосовать». Поле допустимых значений определяется стандартным тегом для полей <INPUT> с атрибутом LIST, значение которого должно совпадать с атрибутом ID тега <DATALIST>. В контейнер тегов <DATALIST></DATALIST> заключаем элементы допустимого списка, которые определяют теги <OPTION></OPTION>, т.е. наша форма должна быть следующей:

```
<FORM NAME=FORM3>
<P>Любимый участник группы LINKIN PARK</P>
<INPUT LIST="LP">
<DATALIST ID="LP">
<OPTION VALUE="Майк"></OPTION>
<OPTION VALUE="Честер"></OPTION>
<OPTION VALUE="Джозеф"></OPTION>
<OPTION VALUE="Феникс"></OPTION>
<OPTION VALUE="Бред"></OPTION>
<OPTION VALUE="Роб"></OPTION>
</DATALIST >
<INPUT TYPE=SUBMIT VALUE="Голосовать">
</FORM>
```

## Задание 4. Сгруппировать элементы формы

Чтобы сгруппировать элементы формы необходимо все поля заключить в контейнер тегов <FIELDSET></FIELDSET>, а тег <LEGEND>используют в качестве заголовка группы элементов формы.

```
<FORM NAME=FORM4>
<FIELDSET>
<LEGEND>Самоеглубокоеозеромира</LEGEND>
<INPUT TYPE="RADIO">Байкал<BR>
<INPUT TYPE="RADIO">Эри<BR>
<INPUT TYPE="RADIO">Ньяса
<P><INPUT TYPE="SUBMIT"></P>
</FIELDSET>
</FORM>
```

**Самостоятельная работа.** Создать тест из десяти вопросов и трех вариантов ответов к каждому в отдельном файле под названием 5.html. Поля формы для теста заключите в ячейки таблицы, состоящей из 30 строк и двух столбцов. В первом столбце объедините три строчки в одну ячейку, для ввода вопроса, и далее объединяйте последующие три строчки в ячейку, в результате вы должны получить 10 ячеек. Во втором столбце, в ячейках, должны находиться поля с вариантами ответов, это могут быть поля-переключатели или поля-списки. Создайте кнопку внизу таблицы таблицей.

Составить отчет о выполнении.

## Практическая работа № 6 «Фреймы»

**Цель работы:** Сформировать умение оформления фреймов на web-странице.

Для создания фрейма используется тег <FRAMESET>, который заменяет тег <BODY> в документе и применяется для разделения экрана на области. Внутри данного тега находятся теги <FRAME>, который определяет область загрузки другого html-документа.

При использовании фреймов в документе подключите в первой строке документа соответствующую спецификацию.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
```

**Задание 1.** Создать простейшую фреймовую структуру.

*Указания к выполнению*

Поделим web-страницу на два фрейма, для этого создадим три документа base.html, left.html и right.html в своей папке. В документе base.html будут располагаться фреймы, поэтому откройте файл в Блокнот и введите следующий код

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<HTML>
<HEAD>
<TITLE>ФРЕЙМЫ</TITLE>
</HEAD>
<FRAMESET COLS="200,*" >
<FRAME SRC="LEFT.HTML" NAME=LEFT><!--!левыйфрейм-->
<FRAME SRC="RIGHT.HTML" NAME=RIGHT><!--!правыйфрейм-->
</FRAMESET>
</HTML>
```

Откройте файл base.html в браузере, в результате ваш документ разделится на две колонки (фреймы) количество и ширину колонок определяет атрибут cols="200,\*", в данном случае у вас два фрейма, первый с толщиной 200 и второй, занимающий оставшееся пространство. Значение атрибут SRC тега <FRAME> определяет html-документ, который будет загружен во фрейм. Теперь определим файлы, загружающиеся во фреймы. Создайте файл left.html в вашей папке и откройте его в Блокнот наберите следующий код

```
<HTML>
<HEAD>
<TITLE>ЛЕВЫЙФРЕЙМ</TITLE>
</HEAD>
<BODY BGCOLOR=SKYBLUE>
<H2 ALIGN=CENTER>левыйфрейм</H2>
</BODY>
</HTML>
```

В созданный файл right.html наберите код

```
<HTML>
<HEAD>
<TITLE>ПРАВЫЙФРЕЙМ</TITLE>
</HEAD>
<BODY BGCOLOR=CYAN>
<DIV ALIGN=CENTER><FONT SIZE=20>правыйфрейм</FONT></DIV>
</BODY>
</HTML>
```

Обновите браузер с документом base.html и посмотрите на результат.

**Задание 2.** Создание структуры из четырех фреймов

*Указания к выполнению*

Для создания структуры из четырех фреймов используем вложенные теги <FRAMESET></FRAMESET>. Сначала страницу разделим на два «строковых» фрейма, а затем их в свою очередь разделим на две колонки. Для этого создадим три 5 файлов: frame.html, frame1.html, frame2.html, frame3.html, frame4.html. Структура фреймов будет находиться в документе frame.html. с соответствующим кодом

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<HTML>
<HEAD>
<TITLE>ФРЕЙМЫ</TITLE>
</HEAD>
<FRAMESET ROWS="50%,50%">
<FRAMESET COLS="50%,50%" >
<FRAME SRC="FRAME1.HTML" NAME=LEFT><!--!первыйфрейм-->
<FRAME SRC="FRAME2.HTML" NAME=RIGHT><!--!второйфрейм-->
</FRAMESET>
<FRAMESET COLS="50%,50%" >
<FRAME SRC="FRAME3.HTML" NAME=LEFT><!--!третийфрейм-->
<FRAME SRC="FRAME4.HTML" NAME=RIGHT><!--!четвертыйфрейм-->
</FRAMESET></FRAMESET>
</HTML>
```

В данном случае размеры фреймов одинаковые, размеры можно выразить как в пикселях так и в процентах. Создайте html коды файлов frame1.html, frame2.html, frame3.html, frame4.html по ниже приведенному примеру.



У тега <FRAMESET> есть еще и атрибуты FRAMEBORDER, BORDERCOLOR и BORDER, которые определяют наличие, цвет и толщину рамки фреймовой структуры соответственно, но они не являются валидными и не признаются спецификацией HTML. **Самостоятельная работа.** Создайте структуру фреймов по образцу

подчеркнутый   *курсив*   **жирный**   ~~зачеркнутый~~   моноширинный

---

<ul style="list-style-type: none"> <li>• желтый</li> <li>• зеленый</li> <li>• красный</li> <li>• синий</li> <li>• оранжевый</li> </ul>	<p>звезда (СВЕТИСЬ, СВЕТИСЬ, ДАЛЕКАЯ ЗВЕЗДА...)</p> <p>Светись, светись, далекая звезда,          Чтоб я в ночи встречал тебя всегда;          Твой слабый луч, сражаясь с темнотой,          Несет мечты душе моей большой;          Она к тебе летает высоко;          И груди сей свободно и легко...</p> <p>Я видел взгляд, исполненный огня          (Уж он давно закрылся для меня),          Но, как к тебе, к нему еще лечу,          И хоть нельзя - смотреть его хочу...</p> 
--	---

Составить отчет о выполнении.

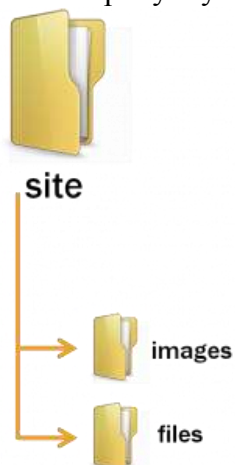
### Практическая работа №7 «Алгоритм разработки сайта»

Цель работы: рассмотреть алгоритм разработки сайта с нуля самостоятельно, т.е. без использования конструкторов сайта, систем управления сайтом и других вспомогательных подобным средств

#### Разработка структуры

Сегодня появилось много средств для разработки сайтов, это и специализированный порталы, предназначенные для конструирования сайта на основе предложенных шаблонов, и системы управления сайтами и другие подобные средства. Данные системы, естественно, облегчают процесс создания ресурса. Но в учебных целях необходимо научиться создавать сайт с нуля самостоятельно. Для этого выберем простую схему, при которой для организации интерфейса страниц сайта используется *фреймовая структура*.

Для начала необходимо подготовить каталог для работы и основные необходимые файлы. Для работы с файлами рекомендуется использовать ПО *Notepad++*. Создайте каталог для работы (с названием *site*, например) и расположите в нем две папки согласно рисунку:



В основном каталоге создайте и сохраните html-файлы со следующими названиями:

- index.html
- header.html
- menu.html
- titul.html
- footer.html
- glava1.html

Откройте файл главный запускаемый файл — *index.html*(файл-раскладка) и добавьте код для организации фреймовой структуры:

```
<html>
<head>
    <title> Фреймы </title>
</head>
<frameset rows="10%,80%,*">
    <frame src="header.html" noresize>
    <frameset cols="20%,80%">
        <frame src="menu.html">
        <frame src="titul.html" name="main">
    </frameset>
    <frame src="footer.html" >
</frameset>
</html>
```

Лого / Название	
Меню	Описание
копирайт	

Затем необходимо добавить код в предназначенные для этого веб-страницы:

header.html

```
<html>
<head>
</head>
<body>
<h1>Лого/Заголовок</h1>
</body>
</html>
```

menu.html

```
<html>
<head>
</head>
<body>
<h1>Лого/Заголовок</h1>
</body>
</html>
```

titul.html

```
<html>
<head>
</head>
<body>
<h1>Описание</h1>
</body>
</html>
```

footer.html

```
<html>
<head>
</head>
<body>
<h1>Копирайт</h1>
</body>
</html>
```

glava1.html

```
<html>
<head>
```

```
</head>
<body>
<h1>Глава 1</h1>
</body>
</html>
```

### Организация работы гиперссылок в меню

Для правильной работы гиперссылок в файле *menu.html*, а именно, чтобы запускаемые файлы открывались именно в правом от меню фрейме, необходимо выполнить два пункта:

1.

В файле фреймовой структуры добавить имя для фрейма, в котором мы собираемся открывать веб-страницы в меню:

```
<frame src="titul.html" name="main">
```

Это мы уже сделали при создании фреймовой структуры. Нужный нам фрейм называется у нас *main*.

2.

В файле *menu.html* добавить код гиперссылок:

```
<html>
<head>
</head>
<body>
<h1> Меню</h1>
<table>
<tr><td>
    <a href="glava1.html" target="main">Глава1 </a>
</td></tr>
<tr><td>
    <a href="glava2.html" target="main">Глава2 </a>
</td></tr>
<tr><td>
    <a href="glava3.html" target="main">Глава3 </a>
</td></tr>
</table>
</body>
```

В ссылках мы указали цель вывода страниц — фрейм с названием *main* — `target="main"`.

### Добавление CSS-стилей

Для оформления сайта, а в некоторых случаях и придания динамичности его элементов необходимо использовать каскадные таблицы стилей.

Сначала создайте документ для хранения стилей — файл *style.css*, и расположите его в папке — *files*.

Для подключения созданного стилевого файла ко всем страницам сайта, необходимо расположить ссылку на этот файл в код всех страниц, кроме файла с фреймовой структурой (*index.html*):

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="files/style.css">
</head>
...
```



## Практическая работа №8 «Создание навигационной карты»

**Цель работы:** Сформировать умения построения навигационной карты, добавление аудио и видео файлов и создание анимации.

**Задание 1.** Создать навигационную карту.

*Указания к выполнению*

В любом созданном прежде HTML файле создайте навигационную карту. Для этого возьмите изображение любимой группы ( к примеру LinkinPark) и зафиксируйте размеры с помощью атрибутов width и height тега <img>. Для начала определим координаты ссылок, создайте следующую конструкцию из тегов.

```
<ahref="#">  
<imgismap="ismap" src="getImageCA7SA2B8.jpg" width="800" height="600" />  
</a>
```

С помощью данной конструкции ваше изображение в браузере становится картой с координатами,

которые должны появляться в конце адресной строки. Наведите на любого члена группы, и в адресной строке вы получите координаты цели, которые в дальнейшем используйте в навигации. Данную конструкцию можно в дальнейшем удалить с html кода, после того как вы определите координаты целей.

Далее установим навигацию по изображению. С помощью тегов <map> и <area>.

При помощи атрибута usemap="#имя\_карты" тега <img>, создаем карту из изображения. Далее подписываем координаты на карте: <areashape="circle" coords="100,200,50" value="Джозеф" title="Джозеф" href="#" />. Атрибут cords определяют координаты целей, которые вы определили выше, атрибут title="Честер" определяют подсказку, которая будет выплывать при наведении мыши.

```
<imgsrc="getImageCA7SA2B8.jpg" width="800" height="600" usemap="#linki" border="0" />  
<mapname="linki">  
<areashape="circle" coords="100,200,50" value="Джозеф" title="Джозеф" href="#" />  
<areashape="circle" coords="200,200,50" value="Майк" title="Майк" href="#" />  
<areashape="circle" coords="300,200,50" value="Феникс" title="Феникс" href="#" />  
<areashape="circle" coords="400,200,50" value="Честер" title="Честер" href="#" />  
</map>
```

**Самостоятельная работа.** Скачайте с интернета изображения структуры ПК и создайте навигацию по данному изображению.

**Задание 2.** Добавить звук и видеовhtml файл.

*Указания к выполнению*

а) Вставка в документ аудио видео файла.

```
<audio src="LOST IN ECHO.mp3" controls="controls" ></audio>  
<video src="video_720p.mkv" controls="controls"></video>
```

б) Вставка анимации

```
<objectdata="пушок.swf" width="60%" height="100%" />
```

Важно! Воспроизведение видео и звука зависит от браузера, поэтому просматривайте html файл в разных браузерах (например Google Chrome).

**Самостоятельная работа.** Добавьте анимацию с помощью тега <embed/>

Составить отчет о выполнении.

## Практическая работа №9 «Способы подключений CSS»

**Цель работы:** Изучить способы подключения стилей CSS к html-документу.

**Задание 1.** Подключение внутреннего стиля.

*Указания к выполнению*

Создайте файл 5.html откройте его в Notepad++ (notepad-plus-plus.org) наберите стандартные теги

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<HEAD>
<META HTTP-EQUIV="CONTENT-TYPE" CONTENT="TEXT/HTML; CHARSET=UTF-
8">
<TITLE>Внутренниестили</TITLE>
</HEAD>
<BODY>
</BODY>
</HTML>
```

Между тегами <BODY></BODY> заключите следующие абзацы

```
<P style="font-size:20pt; font-family:ARIAL; color:darkgreen; text-align: center">
<SPAN style="font-style:italic; font-size:25pt;color:green ">Веб-страница</SPAN> —
документииинформационныйресурсВсемирнойпаутины,
доступккоторомуосуществляетсяспомощьювеб-браузера.</P>
```

```
<Pstyle="font:200% MonotypeCorsiva; color:magenta; text-align: right ">Веб-страницы
обычно создаются на языках разметки HTML и могут содержать гиперссылки для
быстрого перехода на другие страницы.</P>
```

Сохраните и посмотрите на результат в браузере. У нас два абзаца выделенные тегами <P></P> непосредственно к ним с помощью атрибута style подключены стилевые свойства, разделенные точкой с запятой и взятые в кавычки. Если нужно в абзаце выделить фрагмент текста и применить к нему определенные стили используется тег <SPAN>. Стилевые свойства font-size, font-family и color:darkgreenопределяют размер, шрифт и цвет текста соответственно. Во втором абзаце свойство font объединяет в себе значения свойствfont-size и font-family, разделенных через пробел.Свойства text-align выравнивает текст (по центру, ширине, левому и правому краям), font-styleопределяет начертание шрифта (обычное, курсивное или наклонное).

**Задание 2.** Подключение глобальных стилей

*Указания к выполнению*

Создайте файл 6.htmlоткройте его в Notepad++ наберите следующий текст

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
<META HTTP-EQUIV="CONTENT-TYPE" CONTENT="TEXT/HTML; CHARSET=UTF-
8">
<TITLE>Глобальниестили</TITLE>
<STYLE type="text/css">
p { font-size:20pt;
font-family:ARIAL;
```

```

color:darkgreen;
text-align: center
}
.class {font:200% Monotype Corsiva;
color:magenta;
text-align: right
}
span {font-style:italic;
font-size:25pt;
color:green
}
</style>
</head>
<body>
<P><SPAN>Веб-страница</SPAN> — документ или информационный ресурс Всемирной
паутины, доступ к которому осуществляется с помощью веб-браузера.</P>
<P class=class>Веб-страницы обычно создаются на языках разметки HTML и могут
содержать гиперссылки для быстрого перехода на другие страницы.</P>
</BODY>
</HTML>

```

Результат в браузере будет аналогичным, разница лишь в том что стилевые свойства находятся в заголовке веб-страницы. В тегах `<styletype="text/css"></style>` заключены селекторы, а за ними в фигурных скобках свойства со значениями. Первый селектор `p` устанавливает стилевые свойства тексту заключенному в абзацы, второй селектор `.class` устанавливает значения свойств лишь ко второму абзацу `<P class=class>`, у которого значение атрибута `class` соответствует этому селектору, селектор `span` устанавливает свойства тексту заключенному в контейнер тегов `<SPAN></SPAN>`.

**Задание 3.** Подключение связанных стилей.

*Указания к выполнению*

В файле `6.html` вырежьте селекторы заключенные в контейнере тегов `<STYLETYPE="TEXT/CSS"></STYLE>` и вставьте в созданном в Notepad++ новом документе и сохраните его как `style.css` (в начале стилевого документа не забудьте поставить: `@charset "UTF-8";`). В файле `6.html` удалите контейнер `<STYLETYPE="TEXT/CSS"></STYLE>`, а в заголовке добавить следующий текст `<LINKREL="STYLESHEET" TYPE="TEXT/CSS" HREF="style.css">`

Значения атрибутов тега `<LINK>` — `rel` и `type` остаются неизменными независимо от кода, а значение `href` задает путь к CSS-файлу `style.css`. Далее измените заголовок документа `<TITLE>Связные стили</TITLE>`. Результат в браузере должен быть аналогичным предыдущему (за исключением заголовка).

## Практическая работа №10

### «Редактирование html-страницы с применением CSS»

**Цель работы:** Научит основным приемам редактирование текста, списков, изображения и таблиц посредством CSS.

**Задание 1.** Оформление текста с применением стилей.

*Указания к выполнению*

1. Для начала оформим html-документ, весь текст заключим в специальный блочный контейнер тегов <DIV></DIV>. Текст будет состоять из заголовка, двух абзацев и маркированного списка.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<HTML>
<HEAD>
<META HTTP-EQUIV="CONTENT-TYPE" CONTENT="TEXT/HTML; CHARSET=UTF-
8">
<LINK REL="STYLESHEET" TYPE="TEXT/CSS" HREF="new3.css">
<TITLE>Редактирование html-страницы с применением css</TITLE>
</HEAD>
<BODY>
<DIV CLASS=TEXT>
<H2>Альберт Эйнштейн</H2>
<P><SPANid=first>Альбёрт Эйнштéйн</SPAN> (14 марта 1879, Ульм, Вюртемберг,
Германия — 18 апреля 1955, Принстон, Нью-Джерси, США) — физик-теоретик, один из
основателей современной теоретической физики, лауреат Нобелевской премии по физике
1921 года, общественный деятель-гуманист. Жил в Германии (1879—1893, 1914—1933),
Швейцарии (1893—1914) и США (1933—1955). Почётный доктор около 20 ведущих
университетов мира, член многих Академий наук, в том числе иностранный почётный
член АН СССР (1926).</P>
<P>Эйнштейн — автор более 300 научных работ по физике, а также около 150 книг и
статей в области истории и философии науки, публицистики и др. Он разработал
несколько значительных физических теорий:</P>
<UL>
<LI><SPAN>Специальная теория относительности (1905).</SPAN></LI>
<LI><SPAN>В её рамках — закон взаимосвязи массы и энергии: </SPAN></LI>
<LI><SPAN>Общая теория относительности (1907—1916).</SPAN></LI>
<LI><SPAN>Квантовая теория фотоэффекта.</SPAN></LI>
<LI><SPAN>Квантовая теория теплоёмкости.</SPAN></LI>
<LI><SPAN>Квантовая статистика Бозе — Эйнштейна.</SPAN></LI>
<LI><SPAN>Статистическая теория броуновского движения, заложившая основы теории
флуктуаций.</SPAN></LI>
<LI><SPAN>Теория индуцированного излучения.</SPAN></LI>
<LI><SPAN>Теория рассеяния света на термодинамических флуктуациях в
среде</SPAN></LI></UL>
</DIV>
</BODY>
</HTML>
```

Сохраним данный документ как 7.html.

2. Теперь оформим стили на отдельном документе для нашего текста, создадим новый документ new3.css, со следующим содержимым:

```
@charset "UTF-8";
div.text
{ position:absolute; /*устанавливает абсолютное положение блока в браузере*/
left:0px; top:0px; right:0px; /*сбрасывает расстояние от окна браузера до блока с текстом
слева, сверху и справа соответственно*/
```

```
font-family:corbel; /* устанавливает шрифт в блоке с текстом*/
font-size:12pt; /* устанавливает размер шрифта в блоке с текстом*/
line-height:130%;/*устанавливает расстояние между строками относительно высоты
шрифта*/
border-style:double; /*устанавливает стиль рамки блока с текстом */
border-width:3px; /* устанавливает ширину рамки блока с текстом*/
border-color:gold; /* устанавливает цвет рамки блока с текстом*/
background-color: #FFFFCC; /*устанавливает цвет заливки блока с текстом*/
padding:10px; /*устанавливает расстояние от внутреннего края рамки до содержимого
блока*/
margin:20px /*устанавливает расстояние от внешнего края рамки блока до внутренней
границы его родительского элемента (окно браузера)*/}
```

h2

```
{ text-align:center; /*выравнивает заголовок по центру*/
color:#999900; /*устанавливает цвет шрифта заголовка*/
text-shadow:2px 2px 4pxgold; /*создает тень заголовка(по вертикали, горизонтали,
рассеивания и цвет тени)*/}
```

p

```
{ text-align:justify; /*выравнивает абзацы по ширине*/
text-indent:30px; /*устанавливает «красную строку» абзацев*/}
```

#first

```
{ font-style:italic; /*устанавливает стиль шрифта - курсив*/
font-size:14pt; /* устанавливает размер шрифта */
text-decoration:underline; /*делаеттекстподчеркнутым*/
letter-spacing:2px /*устанавливает расстояние между буквами*/}
```

ulspan

```
{ color:#999900; /*устанавливает цвет текста списков*/
font-style:oblique /*устанавливает стиль шрифта – косой*/}
```

ul

```
{ list-style: squareoutside; /*устанавливает стиль маркера*/
color:gold /*цвет маркера*/}
```

Посмотрите результат в браузере.

**Задание 2.** Добавить изображение.

*Указания к выполнению*

Скачать любую фотографию Эйнштейна назовите ее как einstein.gif и сохраните в свою папку. Фотографию поставим слева, причем текст будет обтекать картинку. В документе 7.html, в блоке с текстом, перед заголовком добавим фотографию Эйнштейна, т.е. добавим следующий тег <IMG SRC=einstein.gif>. Добавим фотографии стиливые свойства в конце файла new3.css.

img

```
{ margin:20px; /*устанавливает расстояние от фотографии до текста*/
width:250px; /*ширина фотографии*/
height:350px; /*высота фотографии*/
float:right; /*выравнивает фотографию по правому краю и текст обтекает слева*/
border:7px inset gold} /*устанавливает сразу ширину, стиль и цвет рамки фотографии*/
```

Сохраните и п

осмотрите результат в браузере.

**Задание 3.** Создание таблицы с помощью стилей.

Для начала создадим таблице в документе 7.html после блока с текстом.

```
<table>
<caption>Великие физики</caption>
<tr id=head>
<td>Физик</td><td>Годы жизни</td><td>Достижение</td></tr>
<tr><td>Ампер Андре Мари </td><td>(1775—1836)</td><td>Открыл связь между
электрическими и магнитными явлениями</td></tr>
<tr><td>Бор Нильс Хендрик Давид </td><td>(1885—1962)</td><td>Создал первую
квантовую теорию атома</td></tr>
<tr><td> Галилей Галилео</td><td>(1564—1642)</td><td>Основоположник точного
естествознания</td></tr>
<tr><td> Курчатов Игорь Васильевич </td><td>(1905—1960)</td><td>Открыл явление
разветвления ядерных реакций</td></tr>
<tr><td> Лебедев Петр Николаевич</td><td>(1866—1912)</td><td>Открыл и исследовал
давление света на твердые тела (1899) и газы (1907)</td></tr>
<tr><td>Ньютон Исаак</td><td>(1643—1727)</td><td>Создал физическую картину мира
(теорию пространства и времени)</td></tr>
<tr><td> Рентген Вильгельм Конрад </td><td>(1845—1923) </td><td>Предсказал и открыл
(1895) излучение, названное им X-лучами (рентгеновские лучи)</td></tr>
<tr><td>Эйнштейн Альберт</td><td>(1879—1955)</td><td>Дал формулировку
специальной (1905) и общей (1916) теории относительности</td></tr>
</table>
```

Теперь применим стили, в конце файла new3.css добавим селекторы со следующими свойствами:

```
table
{border:6pxoutsetblue;/* устанавливает рамку всей таблицы (толщину, стиль и цвет)*/
background-image:url(star.jpg);/*устанавливает рисунок в качестве заливки таблицы*/
border-collapse:collapse;}/*устанавливает только одну рамку для ячеек*/
td
{border:2pxdotted #6699FF;/* устанавливает рамку только ячейки (толщину, стиль и
цвет)*/
color:white}/*цвет текста*/
#head
{text-align:center;/*выравнивает текст по центру только в первой строке таблицы*/
background-color:blue;/*устанавливает цвет строки только в первой строке таблицы*/
font-weight:bold}/* устанавливает жирный шрифт тексту в первой строке таблицы*/
caption
{font-size:14pt;/*размер шрифта*/
text-transform:uppercase;/*устанавливает шрифт прописным*/
color:blue}/*цвет текста*/
```

**Задание 4.** Создание ссылок с помощью стилей.

*Указания к выполнению*

В документе 7.html в таблице создайте ссылку из имени Ньютона на его биографию, т.е. заключите это имя в ТЕГИ

```
<A HREF="http://ru.wikipedia.org/wiki/%D0%9D%D1%8C%D1%8E%D1%82%D0%BE%D0%BD,_%D0%98%D1%81%D0%B0%D0%B0%D0%BA">Ньютон Исаак</A> и примените к нему стили с использованием псевдоклассов:
```

```
a:link
```

```
{color:white;}/*цветнетронутойссылки*/  
a:visited  
{color:blue;}/*цвет посещенной ссылки*/  
a:active  
{color:#9966FF;}/* цвет ссылки в момент нажатия на нее*/
```

## Практическая работа №11 «Верстка макетов с помощью CSS»

**Цель работы:** Сформировать умения построения блочных макетов посредством CSS.

**Задание 1.** Создание «жесткой» (фиксированной) верстки с помощью блоков

*Указания к выполнению*

Создадим фиксированный макет с двумя колонками, в документе будут 5 блочных контейнеров <DIV></DIV>: главный – включающий все остальные блоки, заголовочный, левый, правый и нижний – все фиксированной ширины. Создайте новый документ и сохраните его как верстка1.html и наберите в нем следующий текст

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">  
<title>Жесткая верстка блоками</title>  
<style type="text/css">  
#mainblock /*основной контейнер блоков - главный*/  
{ width: 1000px; /* ширина основного блока*/  
margin: auto;} /*выравнивание по ширине*/  
#header /*заголовочный блок */  
{width: 990px; /* ширина заголовочного блока*/  
height: 98px; /* высота заголовочного блока*/  
border-bottom: 2px solid white; /* нижняя рамка заголовочного блока(ширина, стиль и  
цвет)*/  
background-color: darkblue; /*заливка заголовочного блока*/  
padding: 5px ;} /*расстояние то внешней границы блока до его содержимого*/  
#leftblock /*левый блок*/  
{width: 190px; /* ширина левого блока*/  
min-height: 800px; /* высота левого блока*/  
max-height:2000px; /* высота максимального расширения левого блока*/  
background-color: blue; /*заливка левого блока*/  
float: left; /* выравнивание по левой части*/  
padding: 5px;} /*расстояние то внешней границы блока до его содержимого*/  
#rightblock /* правый блок*/  
{width: 790px; /* ширина правого блока*/  
min-height: 800px; /* высота правого блока*/  
max-height: 2000px; /* высота максимального расширения правого блока*/  
background-color: cyan; /*заливка правого блока*/  
float: right; /* выравнивание по правой части*/  
padding: 5px ;} /*расстояние то внешней границы блока до его содержимого*/  
#footer /* нижний блок*/  
{width: 990px; /* ширина нижнего блока*/
```

```

height: 98px; /*высота нижнего блока*/
border-top: 2px solid white; /* верхняя граница нижнего блока*/
background-color: lightblue; /*заливка нижнего блока*/
clear: both; /*отменаобтекания*/
padding: 5px;}
.text
{color: white; /* цветтекста*/
font: bold 30pt Courier New; /*параметрышрифта*/
text-align: center; /* выравниваниешрифта*/
}
</style>
</head>
<body>
<div id="mainblock" class="text">
<div id="header">заголовочный блок</div>
<div id="leftblock">левыйблок</div>
<div id="rightblock" >правыйблок</div>
<div id="footer">нижнийблок</div></div>
</body>
</html>

```

Обратите внимание, как тщательно высчитывается ширина макета, ширина левого(200px) и правого(800px) блока в сумме равна ширине главного блока, при этом в сумме учитывается и расстояние то внешней границы блока до его содержимого(190+10+790+10=1000). Посмотрите результат в браузере.

**Задание 2.** Создание «резиновой» верстки с помощью блоков.

*Указания к выполнению*

При «резиновой» верстке относительно фиксированной ширины будет только левый блок, ширина всех остальных будет зависеть от ширины окна браузера. В документе будут 4 блочных контейнеров <DIV></DIV>: заголовочный, левый, правый и нижний. Создайте новый документ и сохраните его как верстка2.html и наберите в нем следующий текст

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Резиновая верстка блоками</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<style type="text/css">
#header /*заголовочныйблок */
{height: 100px; /* высота заголовочного блока*/
border-bottom: 2px solid white; /* нижняя граница заголовочного блока*/
background-color: violet; /*заливка заголовочного блока*/
padding: 5px ;} /*расстояние то внешней границы блока до его содержимого*/
#leftblock /*левый блок*/
{width: 20%; /* ширина левого блока (20% от ширины окна браузера)*/
min-height: 800px; /* минимальная высота левого блока*/
background-color: deeppink; /*заливкалевогоблока*/
float: left; /*Выравнивание по левому краю*/

```



```
padding: 5px;}
#rightblock /* правый блок*/
{
min-height: 800px; /* минимальная высота правого блока*/
background-color: pink; /*заливка правого блока*/
padding: 5px ;}
#footer /* нижний блок*/
{
height: 100px; /*высота нижнего блока*/
border-top: 2px solid white; /* верхняя граница нижнего блока*/
background-color: lightcoral; /*заливка нижнего блока*/
clear: both; /*отменаобтекания*/
padding: 5px;}
div
{color: aquamarine; /* цвет текста*/
font: bold 30ptMistral; /*параметрышрифта*/
text-align: center; /* выравниваниешрифта*/
}
</style>
</head>
<body>
<div id="header">заголовочныйблок</div>
<div id="leftblock">левыйблок</div>
<div id="rightblock" >правыйблок</div>
<div id="footer">нижнийблок</div>
</body>
</html>
```

Ширина блоков не установлена, т.к. зависит от браузера. Посмотрите результат в браузере.  
**Самостоятельная работа.** Отталкиваясь от этих примеров создать HTML-документы с жесткой и резиновой версткой блоками для трех колонок соответственно.  
 Составить отчет о выполнении.

## Практическая работа №12

### «Динамические эффекты посредством CSS»

**Цель работы:** Научить основным приемам создание динамических эффектов посредством CSS. Показать основные принципы создания меню с выпадающими ссылками, изображения с меняющейся прозрачностью и текста, меняющего размеры и цвет

**Задание 1.** Создать в заголовке меню с выпадающими ссылками.

*Указания к выполнению*

1. Меню создать с помощью таблиц, ячейки которых в свою очередь также будут состоять из выпадающих таблиц, в ячейках которых будут заключены ссылки.

В файле верстка1.html в заголовочном блоке удалите текст «заголовочный блок» и добавьте следующую таблицу, состоящую в свою очередь из других таблиц. Основная таблица состоит из одной строки и 5 колонок (ячеек), в каждой из них своя таблица <tableclass=t1>, появляющаяся при наведении мыши. В ячейках таблицы <tableclass=t1> заключены ссылки. Для начала наберите html-код таблицы с подтаблицами в контейнере <divid=header></div>

```
<table width=990px height=98px><!--Основная таблица меню-->
```

```

<colwidth="190" ><colwidth="190"><colwidth="190"
><colwidth="190"><colwidth="190"><!--Ширина каждой из 5 колонок в основном меню-->
<tr><td>Гранж<!--1 колонка-->
<table class=t1>
<tr><td><a href=http://ru.wikipedia.org/wiki/Nirvana>Nirvana</a></td></tr>
<tr><td><a
href=http://ru.wikipedia.org/wiki/Foo_Fighters_(%D0%B3%D1%80%D1%83%D0%BF%D0%
BF%D0%B0)>Foo Fighters</a></td></tr>
<tr><td><a href=http://ru.wikipedia.org/wiki/Pearl_Jam>Pearl Jam</a></td></tr>
</table>
</td>
<td>Инди-рок
<table class=t1><!--2 колонка-->
<tr><td><a href=http://ru.wikipedia.org/wiki/Radiohead>Radiohead</a></td></tr>
<tr><td><a href=http://ru.wikipedia.org/wiki/The_Smashing_Pumpkins>Smashing
Pumpkins</a></td></tr>
<tr><td><a href=http://ru.wikipedia.org/wiki/Muse>Muse</a></td></tr>
<tr><td><a href=http://ru.wikipedia.org/wiki/Coldplay>Coldplay</a></td></tr>
<tr><td><a href=http://ru.wikipedia.org/wiki/Placebo>Placebo</a></td></tr>
</table>
</td>
<td>Поп-панк<!--3 колонка-->
<table class=t1>
<tr><td><a href=http://ru.wikipedia.org/wiki/Green_Day>Green Day</a></td></tr>
<tr><td><a href=http://ru.wikipedia.org/wiki/The_Offspring>Offspring</a></td></tr>
</table>
</td>
<td>Ню-метал<!--5 колонка-->
<table class=t1>
<tr><td><a href=http://ru.wikipedia.org/wiki/Red_Hot_Chili_Peppers>Red Hot Chili
Peppers</a></td></tr>
<tr><td><a href=http://ru.wikipedia.org/wiki/System_of_a_Down>System Of A
Down</a></td></tr><tr><td><a
href=http://ru.wikipedia.org/wiki/Evanescence>Evanescence</a></td></tr>
</table>
</td>
<td>Брит-поп<!--5 колонка-->
<table class=t1>
<tr><td><a href=http://ru.wikipedia.org/wiki/Blur>Blur</a></td></tr>
<tr><td><a href=http://ru.wikipedia.org/wiki/Oasis>Oasis</a></td></tr>
<tr><td><a href=http://ru.wikipedia.org/wiki/The_Verve>The Verve</a></td></tr>
</table>
</td>
</tr></table>

```

2. В конце глобальных стилей файла верстка1.html, перед тегом </style>, добавьте стилевые свойства к меню.

К основной таблице и подтаблицам подключите следующие стилевые свойства table /\*селектор к основной таблице\*/

```
{font: bold 15ptArial; /*параметры шрифта таблиц*/
text-align:center; /*выравнивание текста в таблице по центру*/
}
```

Зафиксируйте положение ячеек в основной таблице (если добавить свойство position:fixed селектору table, то зафиксируется таблица, причем относительно окна браузера, и даже при прокрутки окна будет фиксированной )

```
td table
{ position:fixed;} /* фиксирует положение ячеек таблицы*/
Для того чтобы скрыть подтаблицы table.t1 воспользуйтесь свойством display: none,
подтаблицы должны появляться только при наведении курсора. Залейте фон подтаблиц
любым цветом и установите прозрачность фоновой заливки opacity:0.5.
table.t1 { /*селектор по классу для подтаблиц*/
display: none; /*скрывает подтаблицы*/
background-color:#0099CC; /*заливка фона подтаблиц*/
opacity:0.5; /*прозрачность фона подтаблиц*/
text-align:center;
}
```

Псевдокласс :hover меняет стилевые свойства при наведении курсора. Нужно что бы при наведении курсора появлялось содержимое ячеек основной таблицы (т.е. подтаблицы), поэтому установите псевдокласс td:hovertablec свойством display:block отображающим скрытое.

```
td:hovertable/* появление подтаблиц*/
{ display:block;}
```

Добавьте рамку к ячейкам подтаблиц

```
table.t1td {border:1pxsolidwhite} /*параметры рамки ячеек подтаблиц */
```

Уберите линию подчеркивания ссылок и залейте их любым цветом

```
a {text-decoration: none; /*нет линии подчеркивания*/
color:#00FF99;} /* цвет ссылки*/
```

Посмотрите результат в браузере, отключайте (берите в комментарии) последовательно стилевые свойства и смотрите как их отсутствие сказывается на виде документа в браузере.

**Самостоятельная работа.** Создайте в левом блоке список, состоящий из выпадающих подписков при помощи псевдокласса :hover и стилевого свойства display с значениями none/block.

Составить отчет о выполнении.

**Задание 2.** Добавить динамический эффект на изображение.

*Указания к выполнению*

1. В правом блоке файла верстка1.html добавьте изображение (скачайте с интернета фото любой рок-группы) изображение должно менять прозрачность при наведении курсора. Используйте псевдокласс :hover и стилевое свойство opacity, изображение выровняйте по левому краю.

Заключите изображение в контейнер тегов, определяющий правый блок

```
<div id="rightblock" >

</div>
```

Теперь подключите следующие стилевые свойства (после стилевых свойств меню)

```
img /*положение изображения*/
{margin: 3px; /*расстояние от изображения до внутренней границы правого блока*/
float: left;} /* выравнивание изображения слева*/
```

```
.pic1/*прозрачность изображения*/  
{opacity: 0.6;}/*устанавливает прозрачность*/  
.pic1:hover /* изменение прозрачности при наведении курсора*/  
{opacity: 1;}/* убирает прозрачность*/
```

2. В правом блоке файла верстка1.html добавьте изображение (скачайте с интернета фото любой рок-группы) рамка изображения должна менять размер при наведении курсора. Добавьте новое изображение после изображения `pic1`.

```

```

Со следующими стилевыми свойствами

```
.pic2 /* ширина рамки второго изображения*/  
{border: 1px solid grey;}/* параметры рамки в режиме покоя*/  
.pic2:hover /* ширина рамки изменяется*/  
{ border: 8px outset grey;} /* параметры рамки при наведении курсора*/
```

Посмотрите результат в браузере

Самостоятельная работа. Добавьте изображение, которое при наведении курсора изменяло бы размер, используя `:hover` и свойства `width` и `height`.

**Задание 3.** Добавить динамические эффекты.

*Указания к выполнению*

Текст должен менять цвет и размер при наведении курсора.

В правом блоке после изображений создайте новый класс с `id=citata` в блок поместите произвольный текст (например, информацию о рок-группе)

```
<div id= citata> Произвольный текст</div>
```

И установит следующие стилевые свойства

```
# citata /* параметры цитаты «в покое»*/  
{width:600px;/* ширина блока цитаты*/  
margin: 3px; /*положение за границей блока*/  
padding: 20px; /*расстояние содержимого до границей блока*/  
border: 6px inset #FF4500; /* параметры границы блока цитаты*/  
float: left; /* выравнивание блока слева*/  
color: #FF4500; /* цвет текста*/  
font: 14ptMonotypeCorsiva; /* размер и шрифт текста*/  
text-align: justify;}/*выравнивание текста в блоке по ширине*/  
# citata:hover/* параметры при наведении курсора*/  
{font-size: 20pt;/* изменение размера текста*/  
color: #8B0000; /* изменение цвета текста*/  
text-align: center; /*выравнивание текста в блоке по центру*/  
border: 6px inset #FF4500;} /* параметры границы блока цитаты*/
```

Сохраните и посмотрите результат в браузере.

## Практическая работа №13

### «Размещение кода JavaScript. Основы JavaScript»

**Цель работы:** Сформировать навыки размещения JavaScript сценариев на html-странице. Изучить иерархическую структуру объектов на html-странице. Изучение новых свойств и методов.

**Задание 1.** Подключит сценарии JavaScript

*Указания к выполнению*

1. Создайте в Notepad++ новый файл `js1.html` –и добавьте в документ сценарий JavaScript. Сценарии, написанные на языке javascript, могут располагаться на самой html-странице

между тегами `<script></script>`, которые в свою очередь обычно находятся в тегах `<head></head>` (теги `<script></script>` могут располагаться и в тегах `<body></body>`, но сценарий выполнится лишь тогда, когда его прочтает браузер). У тега `<script>` должен быть указан параметр `language`, который и указывает язык написания скрипта:

```
<html>
<head>
<title>JS</title>
<script language="javascript">document.write("Hello")</script>
</head>
<body>
</body>
</html>
```

`document.write("Hello")` выведет на экран слово «Hello»

2. Создайте в Notepad++ страницу и сохраните ее как `script.js` в вашу папку, на ней разместите следующий код:

вырезанный из контейнера тега `<script></script>` в файле `js1.html` в тегах `<script language="javascript" src="script.js"`

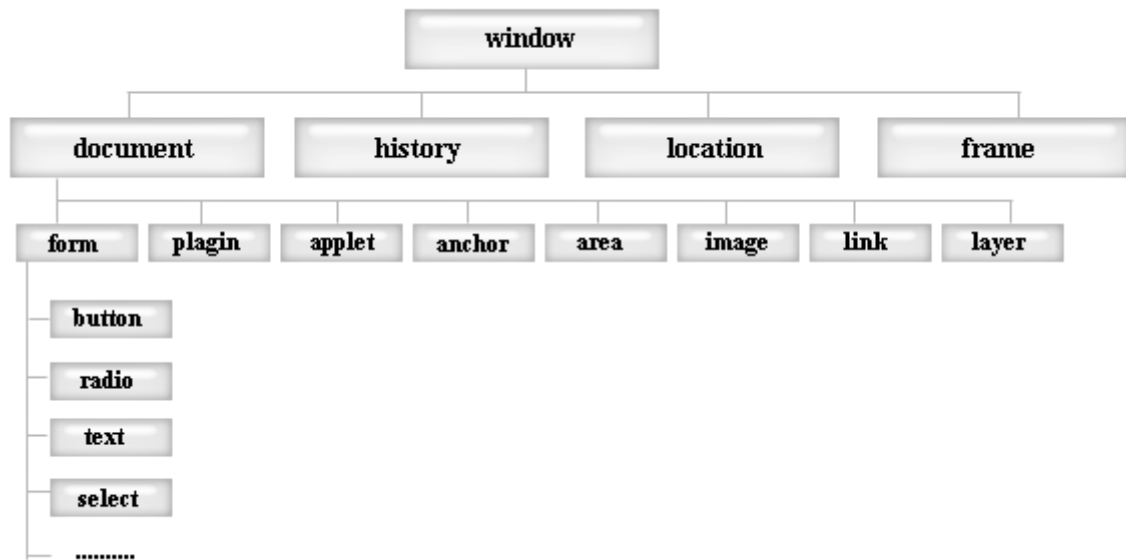
`></script>` в качестве атрибута используйте `language="javascript"` (либо `type="text/javascript"`) и `src="script.js"` (указывает на файл скрипта), т.е. вы должны получить

```
<html>
<head>
<title>JS</title>
<script language="javascript" src="script.js"></script>
</head>
<body>
</body>
</html>
```

**Задание 2.** Изучить иерархическую структуру объектов страницы html.

*Указания к выполнению*

При интерпретации html-страницы браузер создает объекты javascript. Они хранятся в виде иерархической структуры, отражая структуру документа, например:



На самом верхнем уровне находится объект `window`, представляющий окно браузера и являющийся "родителем" всех остальных объектов. Расположенные ниже могут иметь свои подчиненные объекты. Так объект `document` (текущая страница) может иметь дочерний объект `form` (форма) и т.д. Все объекты имеют методы (отделяются от объекта точкой), например: `document.write` позволяет писать текст в текущую страницу, `window.open` открывает новое окно браузера. Также объекты имеют свойства, например: `document.bgColor` содержит значение фонового цвета текущей страницы, `document.title` содержит заголовок страницы. Все инструкции программного кода заканчиваются точкой с запятой.

Установите цвет фона страницы зеленым `<body bgcolor=green>`, в тегах `<script></script>` добавьте условный оператор `if (document.bgColor=="green")`

```

    document.write("Фоновыйцветзеленый")
    else document.write("Фоновый цвет не зеленый");

```

В результате у вас на экране должно появиться сообщение "Фоновый цвет зеленый", измените фоновый цвет и посмотрите результат в браузере.

Теперь добавьте перед условным оператором следующие инструкции `window.alert("Цвет заливки окна");` метод `alert()` на экране браузера создает окно, в котором выводятся данные, которые заданы в качестве аргумента этой функции. После нажатия кнопки "ОК" окно метода `alert()` закрывается. Сохраните и посмотрите результат в браузере.

Создайте новый документ `js1.html` и добавьте следующий код

```

<html>
<head>
<title>JS</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<script language="javascript" >
    if (window.confirm("Хотетеузнатькакойфоновыйцветокна?"))
        document.write("зеленый")
    else document.write("нуиладно");

```

```

</head>
<body bgcolor=green>
</body>
</html>

```

Метод `confirm` выводит на экран диалоговое окно с выбором, если вы нажмете на клавишу «ОК», значение метода `true`, в противном случае `false`. Сохраните и посмотрите результат в браузере.

Удалите содержимое тегов `<script></script>` и добавьте следующий код.

```

var text=window.prompt("Сколько вам лет?"); /*объявление переменной text и
присваивание ей значения, введенного в диалоговое окно*/
switch (true){/* Операторвыбора*/
case (window.parseInt(text)>0&&parseInt(text)<=7):document.write("Выребенок");
break;
case (parseInt(text)>=8)&&(parseInt(text)<=15): document.write("Вышкольник");
break;
case (parseInt(text)>=16)&&(parseInt(text)<=25): document.write("Выстудент")
break;
default : document.write("Вылибоработайтелибопенсионер")}]

```

Метод `window.prompt("Сколько вам лет?")` выводит в окно сообщение "Сколько вам лет?" с полем для ввода информации, которая возвращается этим методом, и в конкретном случае присваивается переменной `text`. Функция `window.parseInt(text)` (или просто `parseInt(text)`) преобразуют строку в целое число, которое возвращает. Введенное вами значение, инициализируется в переменной `text` в виде строки, поэтому мы преобразуем ее в целое число.

`alert()`, `confirm()`, `prompt()` и `parseInt()` это глобальные методы, т.е. методы объекта `window`, поэтому их можно использовать без указания того, что они принадлежат объекту `window`.

**Самостоятельная работа.** Создайте сценарий, который запускал бы, диалоговое окно с предложением представится, в котором в свою очередь можно было бы вбить свое имя, и в результате в окне браузера появлялось приветствие.

Составить отчет о выполнении.

### Практическая работа №14 «События и Функции в JavaScript»

**Цель работы:** Дать навыки использования обработчиков событий совместно с функциями. Создание простейших функций. Формирование элементарных навыков получения доступа к элементам страницы.

**Задание 1.** Обработка событий.

*Указания к выполнению*

Действия, производимые вами на web-странице называются событиями. Для каждого события существует свой обработчик, с помощью которых можно создавать сценарии.

Событие	когда происходит	обработчик события
Blur	потеря объектом фокуса	onBlur
Change	пользователь изменяет значение элемента	onChange

Click	пользователь щелкает мышью по объекту	onClick
DblClick	пользователь делает двойной щелчок мышью по объекту	onDblClick
DragDrop	пользователь перетаскивает мышью объект	onDragDrop
Error	возникновение javascript-ошибки	onError
Focus	окно или элемент формы получает фокус	onFocus
KeyDown	пользователь нажимает клавишу клавиатуры	onKeyDown
KeyPress	пользователь удерживает нажатой клавишу клавиатуры	onKeyPress
KeyUp	пользователь отпускает клавишу клавиатуры	onKeyUp
Load	документ загружается в браузер	onLoad
MouseDown	пользователь нажимает кнопку мыши	onMouseDown
MouseOut	указатель мыши выходит за пределы элемента	onMouseOut
MouseOver	указатель мыши помещается над элементом	onMouseOver
MouseUp	пользователь отпускает кнопку мыши	onMouseUp
Move	пользователь перемещает окно	onMove
Reset	пользователь нажимает кнопку "reset" формы	onReset
Resize	пользователь изменяет размеры окна или элемента	onResize
Select	пользователь выбирает элемент формы	onSelect
Submit	пользователь нажимает кнопку "submit" формы	onSubmit
Unload	пользователь закрывает документ	onUnload

Обработчики событий используются вместе с функциями. Функции в свою очередь объявляются после ключевого слова `function`

```
function имя ()
```

```
{
операторы;
}
```

Создайте сценарий, при котором загрузка страницы вызывала диалоговое окно с приветствием. Создайте либо в заголовке между тегами `<script></script>`, либо на отдельном документе с расширением `.js` следующую функцию

```
function hello()
{alert("
```



Теперь присвоим эту функцию обработчику события, который запустит функцию как только произойдет соответствующее событие, поместим его в тег `<body>` в качестве атрибута

```
<body onLoad="hello()" /*при загрузке страницы появится окно с фразой «Добро пожаловать!»*/
```

### Задание 2. Закрытие окна

*Указания к выполнению*

Создайте новый html-документ `js2.html`, добавьте стандартные теги.

В контейнере тегов `<body></body>` создайте кнопку с надписью «Закрыть окно»

```
<button onClick="shutwindow()">Закрыть окно</button>
```

Объявите функцию `shutwindow()` в заголовке между тегами `<script></script>`

```
functionshutwindow()
```

```
{if (confirm("Вы действительно хотите покинуть страницу?"))
```

```
    window.close();/* глобальный метод объекта window, закрывающее текущее
```

```
окно*/
```

### Задание 3. Вычислить корни квадратного уравнения

*Указания к выполнению*

В документе `js2.html` создайте форму с полями для ввода коэффициентов и кнопкой, запускающей функцию, считающую корни уравнения. К примеру, следующую

```
<form name=sqr>
```

```
<h2>Решение квадратного уравнения  $ax^2+bx+c=0$ </h2>
```

```
<p>a<input id=a type=text size=5></p>
```

```
<p>b<input name=b type=text size=5></p>
```

```
<p>c<input name=c type=text size=5></p>
```

```
<button onClick="korni()">вычислить</button><br>
```

```
</form>
```

Примените стили CSS к созданной вами форме



Создайте функцию вычисляющую корни уравнения которую запускает обработчик событий `onClick="korni()`. Для того чтобы посчитать корни уравнения нужны коэффициенты, т.е. необходимо получить доступ к полям куда будут вводиться коэффициенты. Доступ к полям можно получить разными способами:

1. Через метод объекта `document` `getElementById()`, аргументом которого является значение `id` необходимого элемента. Если необходимо получить введенное значение поля с коэффициентом `a` где `id=a`, тогда `document.getElementById('a').value`, где последнее свойство определяет значение, введенное в поле.

2. Через метод объекта `document` `getElementsByName()`, аргументом которого является значение атрибута `name` необходимого элемента. Для получения введенного значение поля с коэффициентом `c` где `name =c`, необходимо использовать следующую конструкцию `document.getElementsByName('c')[0].value`, где `[0]` указывает что необходимо брать значение у нулевого элемента с атрибутом `name =c`

3. В следующем способе объект form с атрибутом name=sqr используется в качестве свойства объекта document, а элемент input с атрибутом name=b является свойством элемента form. Поэтому используя их имена получить доступ к полю можно посредством атрибута name, каждого элемента document.sqr.b.value.

Тогда функция корни примет вид

```
functionkorni()
    {vara=document.getElementById('a').value; /*объявление и инициализация
переменной*/
    var b=document.sqr.b.value;
    var c=document.getElementsByName('c')[0].value;
    var d=Math.pow(b,2)-4*a*c; /*дискриминант*/
    x1=(-b-Math.sqrt(d))/(2*a); x2=(-b+Math.sqrt(d))/(2*a);
    if (d<0) alert("Корней нет")/*Результат выводится через диалоговое окно*/
    else
    if (d==0) alert("X="+x1)
    else alert("x1="+x1+ " "+"x2="+x2);
    }
```

**Самостоятельная работа.** С помощью JavaScript создать простейший калькулятор. Составить отчет о выполнении.

### Практическая работа №15

#### «Создание сценариев с использованием встроенного объекта Date»

**Цель работы:** Научится создавать объекты Date и использовать методы объекта Date для создания различных сценариев.

**Задание 1.** Создать системные часы.

*Указания к выполнению*

Создайте новый html-документ js3.html, добавьте в него стандартные теги.

В контейнере тегов <body></body> добавьте следующую таблицу

```
<table>
<tr>
<td id=h></td><td>:</td><td id=m></td><td>:</td><td id=s></td></tr>
</table>
```

В ячейку <td id=h></td> будем выводить текущий час, в ячейку <td id=m></td> - текущую минуту и наконец в ячейку <td id=s></td> - текущую секунду.

Теперь создадим функцию time(), выводящую системные часы в html-документ. Т.к. время будет меняться каждую секунду, необходимо чтобы функция обновляла время на странице каждую секунду. Для этого используем глобальный метод setTimeout('time()',1000), который будет вызывать функцию time() каждую секунду.

```
functiontime()
    { vard=newDate(); /*Создание объекта d с текущим датой и временем */
    varh=d.getHours(); /*выделяет текущий час*/
    varm=d.getMinutes(); /*выделяет текущую минуту*/
    vars=d.getSeconds(); /*выделяет текущую секунду*/
    if (h<10) document.getElementById('h').innerHTML='0'+h
    else document.getElementById('h').innerHTML=h;
    if (m<10) document.getElementById('m').innerHTML='0'+m
    else document.getElementById('m').innerHTML=m;
```

```
if (s<10) document.getElementById('s').innerHTML='0'+s
else document.getElementById('s').innerHTML=s;
setTimeout('time()',1000)}
```

Чтобы вывести значение текущего часа в ячейку, используем свойство `innerHTML`, но для начала получим доступ к ячейке с помощью ее `id`, т.е. используем следующую конструкцию `document.getElementById('h').innerHTML=h`;

Для того чтобы время выводилось в формате ЧЧ:ММ:СС, в случае когда часы(минуты, секунды) меньше десяти добавим впереди нуль с помощью следующей конструкции `document.getElementById('h').innerHTML='0'+h`.

Теперь для того чтобы функция `time()` появилась(и вызывала сама себя каждую секунду) в html-документе используем обработчик событий, помещенный в тег `<body onload=time()>`

**Задание 2.** Определить количество дней до вашего дня рождения и на какой день недели он выпадает.

*Указания к выполнению*

1. Допустим, вы родились 25.10.1990. Для того, что бы определить, сколько дней осталось до вашего дня рождения, потребуется создать объект класса `Date`, и применить к нему либо метод `getTime` либо метод `Parse`. Метод `getTime()` без значений возвратит кол-во миллисекунд, прошедших с 1 января 1970 года по текущее время, а `getTime(текущий год,10,25)` возвратит кол-во миллисекунд, прошедших с 1 января 1970 года по 25 октября текущего года. Если взять разницу между этими двумя значениями метода `getTime` и разделить на количество миллисекунд в сутках( $24*60*60*1000$ ) то вы получите количество дней до вашего дня рождения. Округлите результат с помощью метода `toFixed(0)`.

```
function birth()
{var date=new Date();
var date1=new Date(2013,10,5);
var dni=(date1.getTime(2013,10,5)-date.getTime())/(24*60*60*1000)
document.write(dni.toFixed(0));}
```

2. Для того чтобы определить на какой день недели выпадает день рождения нужно взять метод `getDay` с значением даты дня рождения.

```
var date=new Date();
var den=date.getDay(2013,10,5);
```

Т.к. вы получите результат в виде числа из диапазона `[0;6]`, вам необходимо использовать оператор `switch` чтобы вывести результат в формате Понедельник, Вторник, и т.д. (! Не забудьте что по международным стандартам неделя начинается с Воскресенья).

**Самостоятельная работа.** Определите свой возраст в днях.

Составить отчет о выполнении.

## Практическая работа №16

### «Массивы»

**Цель работы:** Формирование навыков использования массивов при разработке сайтов.

**Задание 1.** Создать массив из изображений, которые можно посмотреть посредством перекликивания мышью.

*Указания к выполнению*

Создайте новый html-документ `js4.html`, добавьте в него стандартные теги. Скачайте восемь изображений из Интернета и сохраните их в свою папку как `0.jpg`, `1.jpg`, ..., `7.jpg`. Создайте изображения

```
<imgsrc="0.jpg" width="300" height="200" name="pic" onClick=kartinka() />
```

Функцию kartinka() нужно создать на отдельном файле 2.js и подключим его к файлу js4.html с помощью тега <script type="text/javascript" src="2.js">

В файле 2.js объявите глобальную переменную и массив. Т.к. изначально на странице будет src="0.jpg", т.е. изображение 0.jpg, то при переключении изображения следующим будет изображение 1.jpg, поэтому и объявляем глобальную переменную var r=1;

В массиве img\_array = newArray() будут находиться изображения, каждое из которых само является объектом img\_array[i] = new Image(), с помощью свойства src объекта Image заполните массив изображениями. Затем создайте функцию kartinka(), которая вызывалась бы посредством клика мыши на изображении.

```
var r=1;
var img_array = new Array();
for (var i=0; i<8; i++)
    img_array[i] = new Image();
for (var i=0; i<img_array.length; i++)
    img_array[i].src = [i]+'.jpg';/*инициализация массива*/
functionkartinka(){
if(r==8) r=0 /*условие при котором после последнего изображения, следовало
первое(нулевое)*/
document.pic.src=img_array[r].src;/*присваивание элементу изображения на странице
элементов массива*/
r++;}
```

**Самостоятельная работа.** Создать слайд-шоу из массива изображений.

Составить отчет о выполнении.

**Задание 2.** Объединить два массива и отсортировать их.

*Указания к выполнению*

Создать форму с двумя текстовыми полями, для ввода массивов

```
<form name="form">
ВВЕДИТЕСТРОКИ<br>
<input type=text name=pole1 size=20><br>
<input type=text name=pole2 size=20>
<br>
<input type="button" value="Сортировать" onClick=sort()></input>
</form>
<hr>
<span id="s"></span></form>
```

Функция sort() объединит введенный массивы и отсортирует их. Т.к. данные с полей приходят в виде строки, поэтому мы объявим два объекта String, а затем преобразуем их в массивы.

```
function sort()
{s=String()/*
k=new Array()
s=form.pole1.value;/*инициализация строки данными из первого поля*/
k=s.split(")/*преобразование строки в массив*/
s1=new String();
k1=new Array();
s1=form.pole2.value;/*инициализация строки данными из второго поля*/
k1=s1.split(");
k2=newArray(); k2=k.concat(k1);/*объединение массивов*/
```

```
k2=k2.sort();/*сортировка объединенного массива*/
document.getElementById("s").innerHTML=k2;}
```

**Самостоятельная работа.** Введите два массива, объедините их следующим образом, если массив *x* больше чем массив *y*, тогда массив *x* в объединении будет первым. Элементы, с нечетными индексами выделите жирным шрифтом.

Составить отчет о выполнении.

### Практическая работа №17 «Встроенный объект String»

**Цель работы:** Изучение принципов создания строк и обработки их с помощью методов.

**Задание 1.** Каждое слово в введенной строке должно начинаться с прописной буквы.

*Указания к выполнению*

Строка должна вводиться в текстовом поле, слова в ней разделяться пробелами. Создайте поле для ввода комментариев

```
<form name="forma">
ВВЕДИТЕ СТРОКУ<br>
<textarea name="pole" cols=40 rows=8>
</textarea><br>
<input type="button" value="Изменить" onClick=my()></input>
</form>
<hr>
```

```
<span id="b"></span>
```

В элемент `<span id="b"></span>` будет выводиться результат

Теперь создайте функцию `my()`. Создайте новый объект `s=String()`, для того чтобы получить доступ к полю, в частности к строке, обратимся к значению поля с помощью имен элементов, т.е. `s=forma.pole.value`;

Разбейте строку на слова, т.к. между словами пробелы используем метод `split(" ")`, для того чтобы разбить строку на слова. Слова станут элементами массива, поэтому создадим объект-массив `d=new Array()`, в котором будут храниться в качестве элементов слова из строки `d=s.split(" ")`

Элементы массива тоже строка, для того чтобы выделить первый элемент строки (элемента массива) используйте метод `charAt(0)`. Поменять строчную букву на прописную можно с помощью метода `toUpperCase()`, а точнее, с помощью следующей конструкции `d[i]=d[i].replace(d[i].charAt(0),d[i].charAt(0).toUpperCase());`

Далее из массива снова получите строку, с пробелами между словами `k=d.join(' ')`

```
function my()
{s=String()
k=new String()
s=forma.pole.value;
d=new Array()
d=s.split(" ")
for(i=0;i<d.length;i++)
d[i]=d[i].replace(d[i].charAt(0),d[i].charAt(0).toUpperCase());
k=d.join(' ')
document.getElementById("b").innerHTML=k}/*выводна html-страницуполученнойстроки*/
```

**Задание 2.** Дана последовательность слов. Напечатать все слова, предварительно выполнив преобразования по правилу: заменить в каждом слове первую встречную букву а буквой о, удалив все остальные (если в слове нет такой буквы, то ничего не делать).

*Указания к выполнению*

```
<html>
<head>
<scriptlanguage="Javascript">
var str=prompt("Введите строку", "") /*ввод исходной строки*/
document.writeln("<b>Исходная строка: </b>" +str+"<br>")/*вывод исходной строки для
сравнения*/
l=str.length /*длина исходной строки
s=new Array
slovo=new Array
k=0
for (i=0;i<=l-1;++i){
if (str.slice(i,i+1)== " ") k=k+1 }/*считаетколичествопробелов*/
s1=""/*создаетпустуюстроку*/
slovo=str.split(" ")/*разбиваетстрокунамассивизслов*/
for (i=0;i<=k;++i){
l=slovo[i].length/*определяетдлину слова*/
a=slovo[i].indexOf("a")/*определяетиндексбуквы 'а'*/
s1=s1+slovo[i].slice(0,a)+"о"+" " /*удалениевсехэлементампослепервойпопавшейсябуквы
'а'(включаяеесаму), добавлениебуквы 'о' ипробела*/
document.writeln("<b>Преобразованнаястрока: </b>" +s1+"<br>")
</script>
</head>
</html>
```

**Самостоятельная работа.** Дана строка со словами, разделенными пробелом, найти самое длинное слово в строке.

Составить отчет о выполнении.

## Практическая работа №18

### «Регулярные выражения»

**Цель работы:** Формирование навыков создания регулярных выражений и использование методов их обработки на конкретных задачах.

**Задание 1.** Проверка заполнения формы.

*Указания к выполнению*

Создайте новый html-документ jsb.html, добавьте в него стандартные теги. Для начала создайте форму с полями для ввода имени, фамилии, телефона и электронного адреса.

```
<form name="f">
<span id="s" class=c></span>
<table>
<tr><td>Имя</td><td><input type=text name=p1 size=20></td><td class=c>*</td></tr>
<tr><td>Фамилия</td><td><input type=text name=p2 size=20></td><td class=c>*</td></tr>
<tr><td>Телефон</td><td><input type=text name=p3 size=20></td><td>8(917)23-24-
456</td></tr>
<tr><td>Электронныйадрес</td><td><input type=text name=p4 size=20></td><td
class=c>*</td></tr>
<tr><td colspan=3><input type="button" value="Отправить"
onClick=check()></input></td></tr>
<tr><td colspan=3>* - Поля обязательные для заполнения</td></tr>
```

</form>

Имя	<input type="text"/>	*
Фамилия	<input type="text"/>	*
Телефон	<input type="text"/>	8(917)23-24-456
Электронный адрес	<input type="text"/>	*
<input type="button" value="Отправить"/>		

\* - Поля обязательные для заполнения

Теперь создайте функцию check(), в ней нужно создать два регулярных выражения. Для создания объектов регулярного выражения используйте разные способы. Регулярное выражение номера телефона должно соответствовать шаблону 8(000)00-00-000. Для проверки регулярного выражения использовать метод RegExp.test(), аргументом которого является строка и возвращает истину, если регулярное выражение соответствует строке и ложь в противном случае.

```
function check()
{
  var reg1=/8\{0-9\}\{3\}\{2\}[0-9]\{3\}/ /*регулярное выражения номера телефона*/
  var reg2=new RegExp("[0-9a-z_]+@[0-9a-z_]+\.[a-z]\{2,3\}", "i") /*регулярное выражения эл. адреса*/
  str1=f.p3.value; /*получения строки из поля с номером*/
  str2=f.p4.value; /*получения строки из поля с адресом*/
  if (!reg1.test(str1)) document.getElementById("s").innerHTML='Введите номер телефона правильно'
  if (!reg2.test(str2)) document.getElementById("s").innerHTML='Введите электронный адрес правильно'
  /*проверка регулярных выражений с использованием условного оператора*/
}
```

**Задание 2.** Дана строка с датой рождения определить, возраст.

*Указания к выполнению*

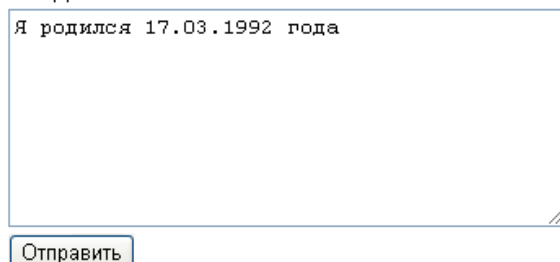
Строка будет вводиться в текстовом поле, необходимо в строке найти дату, которая будет вводиться в формате ЧЧ.ММ.ГГГГ. Затем определить возраст по дате.

```
<form name="f1">
ВВЕДИТЕ СТРОКУ<br>
<textarea name="pole" cols=40 rows=8>
</textarea><br>
<input type="button" value="Отправить" onClick=check2()></input>
</form>
```

Функция check2() должна обработать введенную строку. Используйте метод exec() объекта RegExp чтобы найти дату в строке. Т.к. данный метод возвращает массив из состоящий из самой даты и частей даты, которые заключены в скобки регулярного выражения. Пусть reg=/([0-9]\{2\})\.([0-9]\{2\})\.([0-9]\{4\})/ - регулярное выражение соответствующее шаблону ЧЧ.ММ.ГГГГ, тогда reg.exec() возвратит следующим массив из четырех элементов {ЧЧ.ММ.ГГГГ, ЧЧ, ММ, .ГГГГ}. Элементы массива используйте в качестве значений при создании объекта Date.

```
function check2()
{
  var reg=/([0-9]\{2\})\.([0-9]\{2\})\.([0-9]\{4\})/; /*создание регулярного выражения*/
  str=f1.pole.value; /*получение введенной строки из формы*/
}
```

```
mas=reg.exec(str);/*создание массива {ЧЧ.ММ.ГГГГ, ЧЧ, ММ, .ГГГГ}*/
date1=newDate(mas[3],mas[2],mas[1]); /*Создание объекта Date со значениями элементов массива*/
date=newDate();/*Создание текущей даты*/
vozrast=(date.getTime()-date1.getTime())/(24*60*60*1000*365) /*определениевозраста*/
document.getElementById("b").innerHTML='Вашвозраст '+vozrast.toFixed(0); }
ВВЕДИТЕ СТРОКУ
```



Ваш возраст 21

**Самостоятельная работа.** Дана строка, заменить все числа в ней на звездочки (для этого используйте метод replace).

Составить отчет о выполнении.

### Практическая работа № 19 «Проверка заполнения форм»

**Цель работы:** Формирование навыков проверки html-форм с помощью сценариев javascript.

**Задание 1.** Создайте новый html-документ js7.html, добавьте в него стандартные теги и создайте в нем форму для заполнения анкеты со следующими полями

1. Фамилия
2. Имя
3. Дата рождения
4. Пол
5. Знак зодиака(любимый цвет)

*Указание к выполнению*

Первое, второе и третье поля в форме должны быть текстовыми полями, чтобы получить доступ к полю воспользуйтесь следующей конструкцией: document.getElementById("name").value. Третье поле должно вводиться в формате ДД.ММ.ГГГГ, значит значение поля нужно будет проверить на соответствие регулярному выражению (соответствующему формату ДД.ММ.ГГГГ). Тип четвертого поля установите как type="radio", например можно установить два поля (мужской пол и женский пол). Получить доступ к полю можно с помощью следующей конструкцией: document.getElementById("M").checked, его значение true – если стоит галочка и false – если ее нет. Пятое поле – выпадающий список, чтобы получить доступ к этому полю нужна следующая конструкция: x=document.getElementById("link"); y=x.options[x.selectedIndex].text, тогда в y будет значение выбранного поля. Заполненные поля должны отправляться с помощью кнопки, которая в свою очередь запускает функцию обрабатывающую форму. Данными полученными из полей вы можете инициализировать поля объекта или массив. Проверяйте поле на правильность заполнения, если оно заполнено не правильно выведите сообщение методом alert("ВВЕДИТЕ ДАТУ ПРАВИЛЬНО"); , полученные данные из полей выведите на экран с помощью метода confirm("Это вы"+"



"+mas.toString()), чтобы пользователь смог проверить правильность заполнения анкеты. Примерный вид формы указан ниже, скрипт помещайте либо в отдельном файле, либо в <head></head> .

```
<form><table align=center >
<tr><td>Имя <input type="text" id="name"></input></td>
<td>Фамилия <input type="text" id="surname"></input></td></tr>
<tr><td colspan=2>Дата рождения <input type="text" id="date"></input></td></tr>
<tr><td>Пол </td><td>М <input type="radio" id="M" name="M"></input>
Ж <input type="radio" id="W" name="W"></input></td></tr>
<tr><td colspan=2><select id="cod"><option >blue</option>
<option>red</option>
<option>yellow</option>
<option>green</option>
<option>pink</option></select></td></tr>
<tr><td colspan=2><input type="submit" onclick="f()"></td></tr>
</table></form>
```

Определим функцию обработки формы

```
function f()
{ mas=new Array();/*объявлениямассива*/
i=0;/*инициализация массива данными с полей форма*/
mas[i]=document.getElementById("name").value; i++;
mas[i]=document.getElementById("surname").value; i++;
mas[i]=document.getElementById("date").value;
if(/(\d{2})\.\d{2}\.\d{4}/.test(mas[i])) /*проверка ввода даты*/
mas[i]=RegExp.$1/*сохранение дня рождения в качестве элемента массива*/
else alert("ВВЕДИТЕ ДАТУ ПРАВИЛЬНО");
i++;
y=document.getElementById("M").checked/*получение данных из радио-полей*/
z=document.getElementById("W")
if((y==true)) mas[i]="Мужчина"
else
if((y==false)&&(z.checked==true)) mas[i]="Женщина"
else alert("ВВЕДИТЕ ВАШ ПОЛ ПРАВИЛЬНО");
i++;
x=document.getElementById("cod")
mas[i]=x.options[x.selectedIndex].text;/*получение выбранного значения из списка
выбора*/
confirm("Это вы"+" "+mas[0].toString()+" "+mas[1].toString()+
" рожденный(ая) "+mas[2].toString()+" числа "+mas[3].toString()+
", любящий(ая) цвет: "+mas[4].toString());/*вывод полученного массива в диалоговом
окне*/
}
```

**Самостоятельная работа.** Добавьте новые поля в форму и получите данные из них и обработайте данные javascript-сценарием  
Составить отчет о выполнении.

## Практическая работа № 20

### «Работа с файлами в PHP»

**Цель работы:** Изучение основ работы с файлами в PHP, загрузка файла на сервер и сохранение данных формы в созданном файле.

Для выполнения лабораторной работы вам понадобится сконфигурированная связка MySQL, Apache, PHP, phpMyAdmin, либо просто Denwer. Скачать Denwer можно с официального сайта <http://www.denwer.ru/>, на странице есть ссылки по установке и эксплуатации данного программного обеспечения.

**Задание 1.** Загрузить файл на сервер.

*Указания к выполнению*

Скопируйте html-документ js7.html в папку с вашим сайтом в Denwer. Загрузка файла на сервер осуществляется с помощью multipart-формы, поэтому в нашу форму добавим следующие атрибуты: `<form name="forma" method="post" action="2.php" enctype="multipart/form-data">`. Добавим новое поле в нашу форму `<input type="file" name="filename">`.

Теперь создадим документ 2.php и в него добавим символы для скриптов PHP. Скрипты можно добавить следующими способами:

1. `<?php ?>`
2. `<? ?>`
3. `<script language="php"></script>`
4. `<% %>`

Комментарии

```
<?php
```

```
// Это однострочный комментарий
```

```
/* Это многострочный комментарий*/
```

```
?>
```

После отправки файла в PHP создается суперглобальный двумерный массив `$_FILES`, содержащий информацию о файле.

- `$_FILES['filename']['name']` - имя файла до его отправки на сервер, например, заяка.gif
- `$_FILES['filename']['size']` - размер принятого файла в байтах
- `$_FILES['filename']['type']` - тип принятого файла, например: image/gif
- `$_FILES['filename']['tmp_name']` - содержит имя файла во временном каталоге, например: /tmp/phpV3b3qY;
- `$_FILES['filename']['error']` - Код ошибки, которая может возникнуть при загрузке файла. Ключ `['error']` был

Прежде чем загрузить файл нужно определиться с его размером, если он превышает 2Мбайт, мы его просто не загрузим.

```
if($_FILES["filename"]["size"] > 1024*2*1024)
{echo "Размер файла превышает два мегабайта";
exit;}
```

Функция `exit` прекращает выполнение скрипта.

Теперь переместим файл из каталога для временного хранения файлов в нужный нам каталог, для этого создайте новую папку в вашей папке `www` именем `loads` (еще ее можно создать с помощью функции `mkdir("loads");`). Функция `move_uploaded_file()` перемещает файл из временного хранилища в место указанное во втором аргументе, и возвращает `TRUE` в положительном исходе перемещения, и `FALSE` если файл не был перемещен. Второй аргумент это выражение типа: `loads/заяка.gif`, где `.` складывает две строки.

```

if (move_uploaded_file($_FILES['filename']['tmp_name'],
"loads/".$_FILES['filename']['name']))
{echo "Файл успешно загружен на сервер";}
else { echo " Не удалось загрузить файл на сервер!"; exit; }
Ваш скрипт должен выглядеть следующим образом:
<?php
mkdir("loads");
// Каталог, в который мы будем принимать файл:
if($_FILES["filename"]["size"] > 1024*3*1024)
{
echo "Размер файла превышает три мегабайта";
exit;
}
print $_FILES['filename']['name'];
// Копируем файл из каталога для временного хранения файлов:
if (move_uploaded_file($_FILES['filename']['tmp_name'],
"loads/".$_FILES['filename']['name']))
{
echo "<h3>Файл успешно загружен на сервер</h3>";}
else { echo "<h3>Ошибка! Не удалось загрузить файл на сервер!</h3>"; exit; }
// Выводим информацию о загруженном файле:
echo "<h3>Информация о загруженном на сервер файле: </h3>";
echo "<p><b>Оригинальное имя загруженного файла:</b>";
echo "<p><b>Mime-тип загруженного файла: ".$_FILES['filename']['type']."</b></p>";
echo "<p><b>Размер загруженного файла в байтах: ".$_FILES['filename']['size']."</b></p>";
echo "<p><b>Временное имя файла: ".$_FILES['filename']['tmp_name']."</b></p>";
?>

```

**Задание 2.** Создать файл в PHP и записать в него данные из формы.

*Указания к выполнению*

Сохраним полученные из формы данные в текстовой файл. Для начала создадим этот файл, для этого используем стандартную функцию fopen() в режиме w+. В файле 2.php удалите старый скрипт и добавьте следующую строку:

```
$fp = fopen('1.txt', 'w+') or die;
```

Убедитесь, что в вашей папке www появился файл 1.txt. Функция die синоним exit, прекращает работу скрипта. Теперь заполним файл информацией, для этого воспользуемся функцией fwrite(). Данные из полей будем записывать на новых строках, PHP\_EOL определяет символ конец строки. В html-документе js7.html, в форме все атрибуты id заменить на name, т.к. данные из полей посредством глобального массива передаются на сервер по значениям атрибута name. Все данные формы отправляются посредством метода передачи данных post, поэтому при передачи на сервер создается глобальный массив \$\_POST['имя поля']. Значение первого поля «Имя» в глобальном массиве будет выглядеть как \$\_POST['name'].

```
fwrite($fp,$_POST['name']);
```

```
fwrite($fp,PHP_EOL);
```

```
fwrite($fp,$_POST['surname']);
```

ит.д.

```
fclose($fp); // закрытие файла
```

**Самостоятельная работа.** Открыть файл с экзаменационными вопросами и посчитать количество символов и строк в файле, используйте функцию `fgetc()`. Составить отчет о выполнении.

### Практическая работа №21

#### «Сохранение данных из формы в базе данных MySQL»

**Цель работы:** Сформировать практические умения сохранение данных из формы в базе данных с помощью PHP скриптов.

**Задание 1.** Сохранить данные в базе.

*Указание к выполнению*

1. Создайте простую форму, подобную той которую создали в файле `js7.html`, можете взять и ее. Скрипт можете поместить в файл с формой, либо создать отдельно, например, создайте файл `3.php`.

2. Создайте в MySQL или в phpMyAdmin базу данных `base` и в ней таблицу `table`, в которой создайте поля, соответствующие форме (`name`, `surname`, `age`, `pol`, `color`), тип каждого поля пусть будет TEXT. Созданная вами база должна появиться в папке `data` в установочной папке MySQL, например `*\mysql-5.5\data\base`. Проверьте!

3. Теперь напишем скрипт. Откроем соединение с сервером MySQL, с помощью функции `mysql_connect()`, где первый аргумент, в приведенном ниже примере, - локальный сокет, второй – встроенный пользователь `root`. В случае удачного соединения функция возвращает в `$date` указатель на соединение с MySQL, в противном случае возвращается FALSE и функция `mysql_error()` возвратит строку с ошибкой.

```
$date=mysql_connect("localhost","root") or die (mysql_error ());
```

Далее выбираем базу данных с помощью функции `mysql_select_db()`, которая возвращает TRUE при удачном соединении с БД, в противном случае появится строка с ошибкой.

```
mysql_select_db("base",$date) or die(mysql_error());
```

Теперь оформим запрос, в нем мы в таблицу `table` в соответствующие поля заполняем информацией полученной из формы методом POST.

```
$sql="INSERT INTO table SET
```

```
name='{$_POST['name']}',surname='{$_POST['surname']}',age='{$_POST['date']}',pol='{$_POST['M']}',color='{$_POST['S']}';
```

Отправим запрос с помощью функции `mysql_query()`, в качестве первого аргумента выступает сам запрос `$sql`, а второй аргумент это указатель на активное соединение. Если запрос был выполнен, тогда, в зависимости от запроса, в `$res` будет либо указатель на результат запроса, либо TRUE (как в нашем случае), в противном случае будет FALSE и появится строка с ошибкой.

```
$res=mysql_query("$sql",$date)
```

```
ordie(mysql_error());
```

Теперь данные сохранились в БД на сервере и можно закрыть соединение

```
mysql_close($date) or die (mysql_error ());
```

Проверьте базу данных `base` на наличие данных из формы.

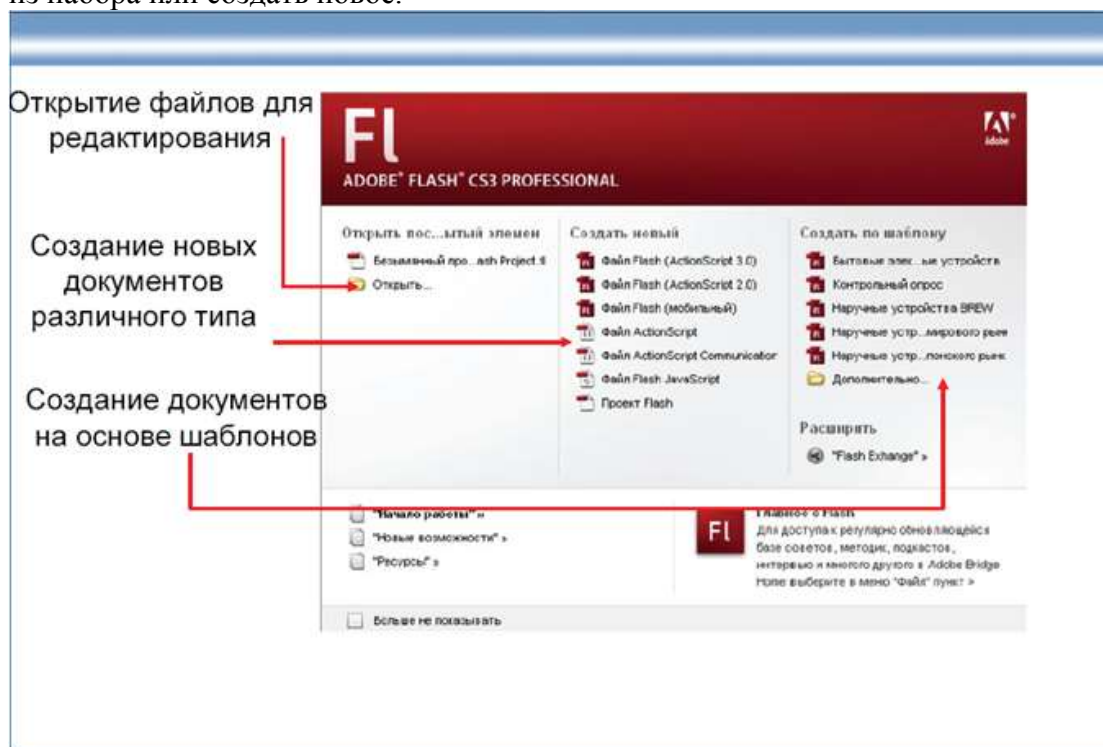
### Практическая работа №22

#### «Знакомство с интерфейсом редактора Flash CS3. Рисование»

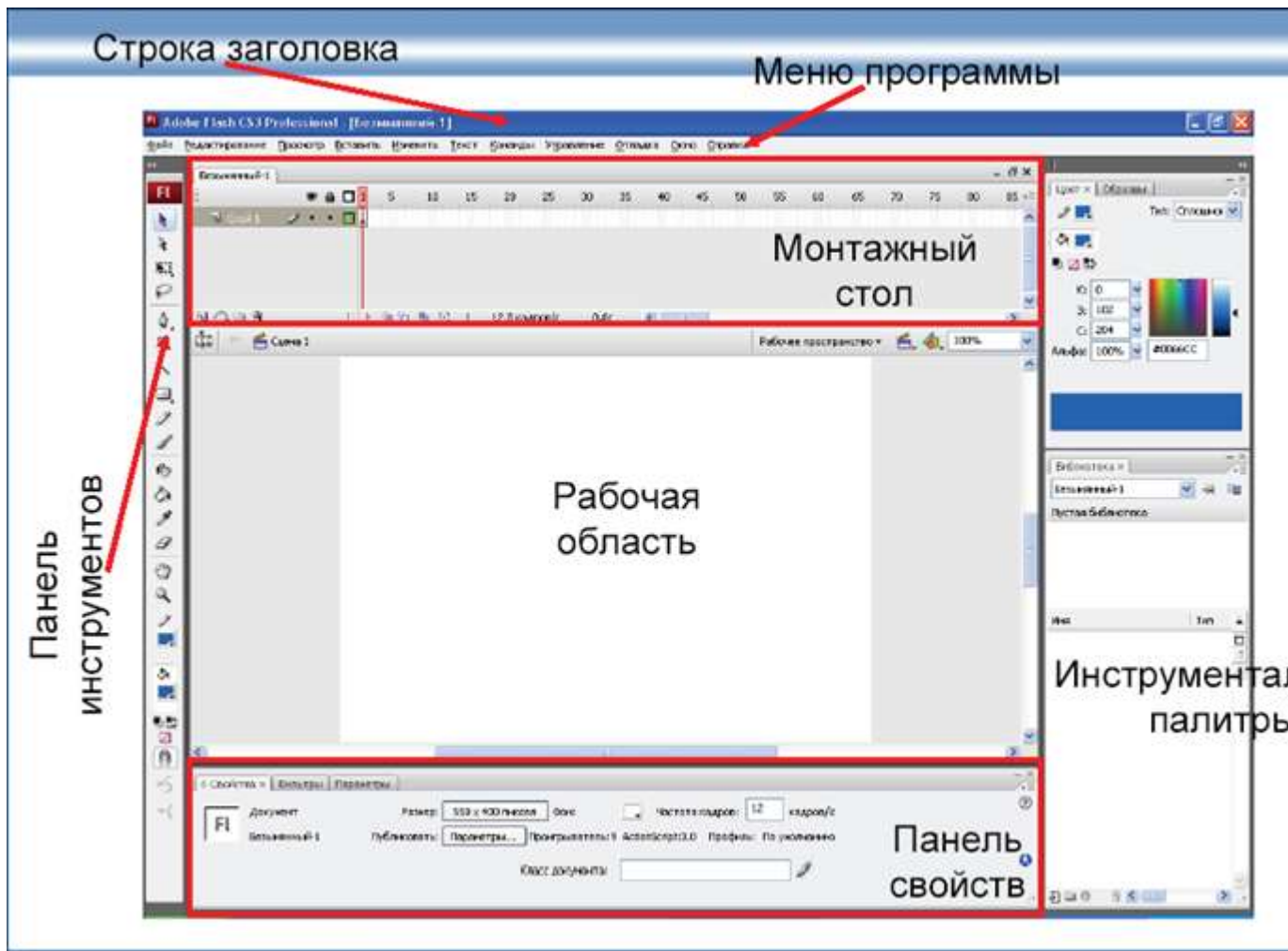
Цель работы: Создание документа и манипуляции с рабочей средой

**Теория: интерфейс программы**

Для создания документов и файлов и управления ими используются такие элементы интерфейса, как палитры, панели и окна. Размещение этих элементов называется *рабочим пространством/средой*. Рабочие пространства/среды разных приложений Adobe Creative Suite 3 имеют похожий вид, что позволяет пользователю легко переходить с одной среды в другую. Естественно, возможна настройка каждого приложения в соответствии со потребностями пользователя. Для этого необходимо выбрать рабочее пространство/среду из набора или создать новое.



1. Окно разделено на отдельные блоки:
  - *Открытие файлов.* Здесь же находятся последние открытые документы.
  - *Создание новых файлов* (в том числе и для создания контента для мобильных телефонов).
  - *Использование шаблонов для создания файлов.*
  
2. Для продолжения работы в Adobe Flash CS3 создайте файл Flash (Action Script 3.0).  
Окно Adobe flash имеет:



- заголовок, содержащий имя приложения и название открытого файла;
- строку меню;
- панель инструментов (инструменты в палитре поделены на группы);
- под строкой меню находится монтажный стол, состоящий из палитры слоев и шкалы времени (этот инструмент предназначен для работы с анимацией);
- панель навигации, которая содержит элементы, управляющие рабочим пространством;
- в центральной части располагается рабочая область;
- в нижней части окна находится панель свойств;
- справа отображаются инструментальные палитры (Цвета, Библиотека).

*Панель «Инструменты»:* здесь содержатся инструменты для создания и редактирования изображений, графических объектов, элементов страниц и т.д. Связанные по смыслу и назначению инструменты располагаются в группах.

Flash, Dreamweaver и Fireworks имеют *Инспектор свойств*, который показывает настройки текущего выбранного в данный момент элемента или инструмента.

*Окно документа:* отображает файл, который находится в работе. В окна документа можно поместить закладки, а в некоторых случаях, их можно группировать и закреплять.

*Палитры* упрощают отслеживание и изменение обрабатываемого объекта. Их можно группировать, располагать в один ряд или закреплять.

### **Задание 1. Интерфейс рабочего окна**

1. В окне *Начало работы* выберите *Создать новый* → файл *Flash (Action Script 3.0)*.
2. Выполните команду *Окно* → *Панель инструментов* → *Основная*.

3. Посмотрите, какая панель инструментов добавилась на экране.
4. Расположите панель *Цвета*, как показано на рисунке, перетащив её нажатой левой кнопкой мыши.
5. Скройте временную шкалу.
6. Сохраните рабочее пространство, назовите его «*Рисование*».
7. Скройте все панели с помощью клавиши *F4*.
8. Восстановите среду по умолчанию (*Рабочеепространство* → *По умолчанию*).

### Практическая работа №23

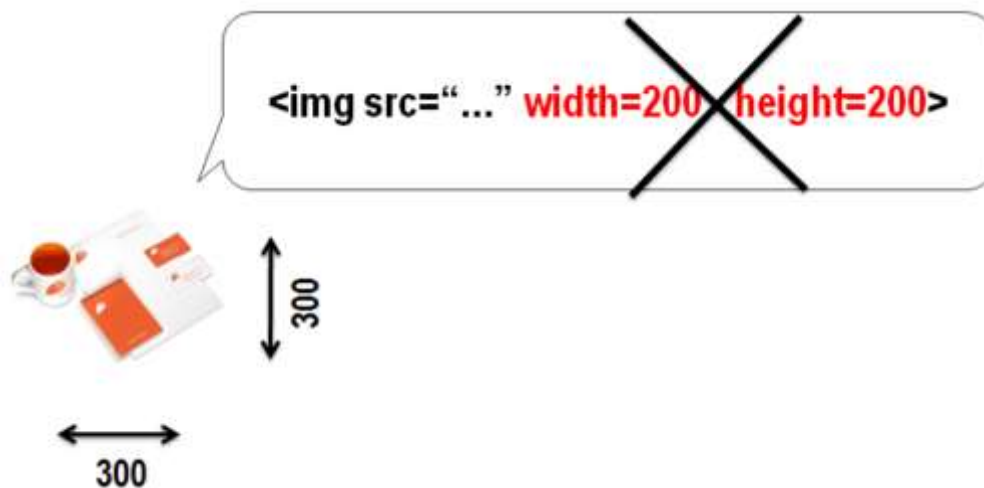
#### «Требования к иллюстрациям в Internet и подготовка графики для Веб в Photoshop»

Цель работы: изучение алгоритма подготовки графических элементов веб-страницы

Требования к иллюстрациям для ВЕБ

Общие пожелания для подготовки изображений:

- Графические элементы должны быть представлены в формате **GIF** или **PNG**.
- Фотографии должны быть представлены в формате **JPEG**.
- Обеспечьте показ изображений с малым разрешением, пока идет загрузка больших изображений.
- Используйте **GIF**-изображения с чересстрочным форматом.
- Укажите атрибуты **height** и **width** элемента **img**.
- Сократите количество анимации.
- По возможности используйте пиктограммы изображений.
- Обязательно создавайте «альтернативный» текст (атрибут **alt**)



#### Градиент на фон в Photoshop

При размещении градиента на фон, подбирается либо горизонтальный либо вертикальный градиент, и, в зависимости от этого, вырезается лишь тонкая линия данного градиента.

При размещении в CSS устанавливаются соответствующие свойства фона (см. рис. 1.1)

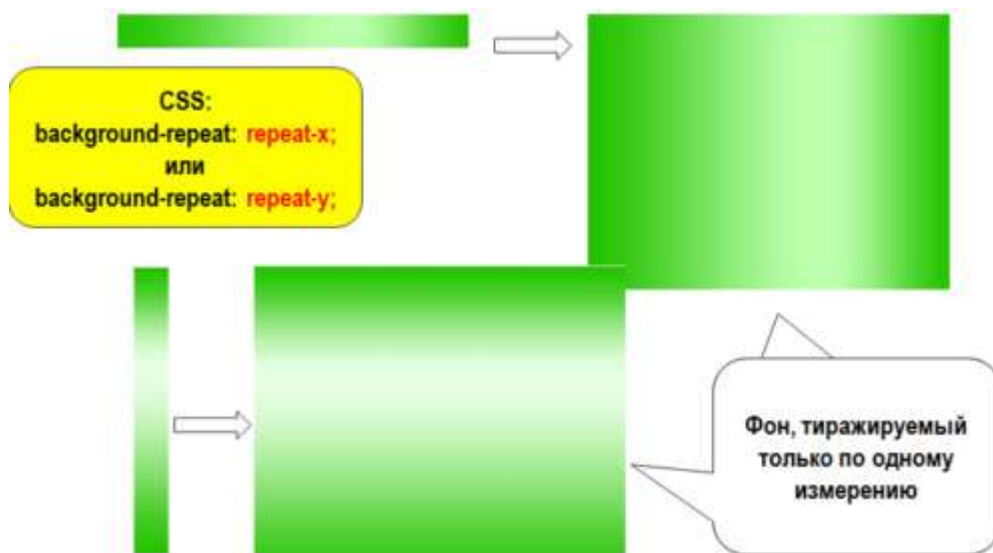


Рис. 1.1. Градиент на фон в Photoshop

### Диалоговое окно Сохранить для Интернета и устройств

Прежде чем использовать файл изображения в веб-странице, необходимо в графическом редакторе подготовить его для размещения. При этом важно не только определить в photoshop размер изображения, но и правильно сохранить: используя **диалоговое окно Сохранить для Веб и устройств** (см. рис. 1.2):

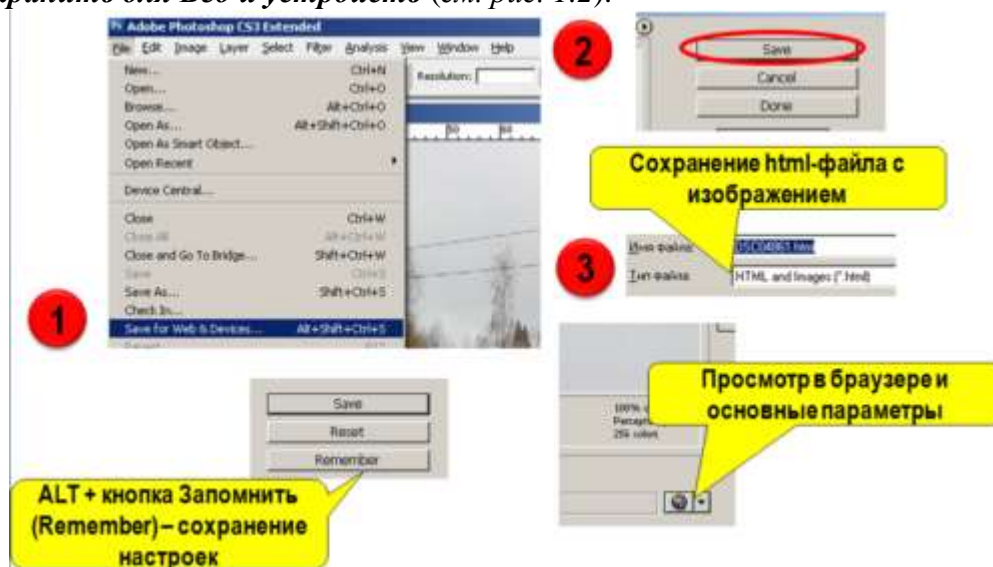


Рис. 1.2. Диалоговое окно Сохранить для Интернета и устройств

Устанавливаем настройки диалогового окна **Сохранить для Интернета и устройств**:



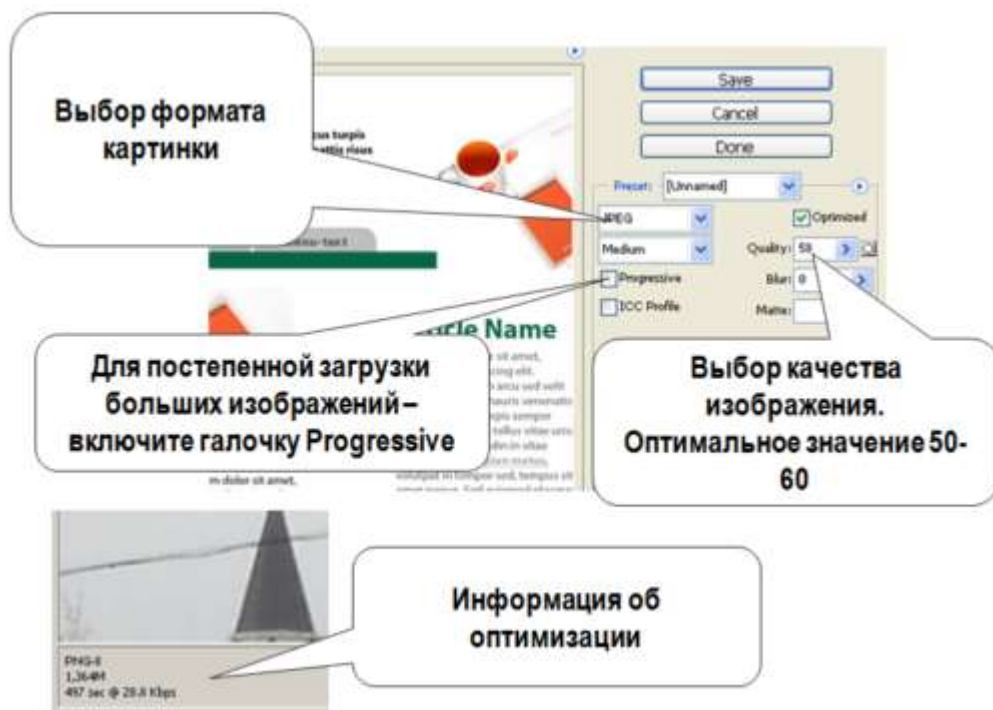


Рис. 1.3. Настройки диалогового окна Сохранить для Интернета и устройств

#### Оптимизация и сжатие до заданного размера файла:



Рис. 1.4. Оптимизация и сжатие до заданного размера файла

#### Практическая работа Photoshop 1\_1:

1. Откройте изображение в Photoshop и выберите «Файл» -> «Сохранить для Интернета и устройств».
2. В диалоговом окне «Сохранить для Интернета и устройств» щелкните вкладку «Оптимизированный».
3. В меню «Стиль» выберите «Низкое качество JPEG».
4. Щелкните вкладку «Размер изображения».
5. Убедитесь, что установлен параметр «Сохранить пропорции» и введите ширину. Для электронной почты подходит размер **400** пикселей.

6. Нажмите кнопку «Сохранить». Введите имя файла и путь для сохранения файла. Убедитесь, что в меню «Формат» выбран вариант «Только изображения». Нажмите кнопку «Сохранить».

### Установка параметров оптимизации JPEG

Для размещения качественных фото, необходимо подготовить в Photoshop изображение в формате JPEG:



Рис. 1.5. Параметры оптимизации JPEG

### Установка параметров оптимизации для форматов GIF и PNG



Метод Смешение цветов и метод псевдополуптонов:  
GIF-изображения с 0% псевдополуптонов (слева) и 100% псевдополуптонов (справа)

Рис. 1.6. Параметры оптимизации для форматов GIF и PNG

### Практическая работа Photoshop 1\_2: JPEG-сохранение

- Откройте в редакторе изображение jpg формата (или скачайте изображение желтого цветка) и вызовите диалог «Сохранить для Веб и устройств» (Save for Web & Devices) в меню File.
- Выберите вкладку «Оптимизированный»(Optimized)
- Откройте список «Наборы параметров»(Preset) и выберите один из 12 предлагаемых вариантов – JPEG Medium(средний)
- Оцените качество изображения и размер итогового файла

- Уровень качества, можно грубо задать в меню, а можно точно настроить ползунком «*Качество*» (*Quality*): передвиньте ползунок на 70.
- Включите флажок опции «*Оптимизированный*» (*Optimized*): позволяет чуть уменьшить размер файла в обмен на ограничение совместимости со старинными браузерами.

### **Практическая работа Photoshop 1\_3: GIF-сохранение**

- Скачайте файл для работы.
- Из «*списка форматов*» выберите пункт «*GIF*».
- Выберите (или впечатайте) количество оттенков в поле «*Цвета*» (*Colors*) и посмотрите, как это отразилось на изображении и размере итогового файла (установите 90)
- Для уменьшения исходной гаммы картинке до заданного количества, можно воспользоваться одним из 9 доступных методов редукции цвета. Выберите одну из четырёх верхних опций: установите «*Ограниченная*» (*Restrictive*)
- Смещение цветов (*Dithering*) отчасти компенсирует узость гаммы, «конструируя» отсутствующие цвета из набора имеющихся. Выберите «*Диффузия*» (*Diffusion*) и передвиньте ползунок *Dither* (степень количества полутонов) на 80.
- В таблице цветов выберите один из цветов (желтый), дважды щелкнув по цвету, и замените его другим оттенком. Для выбора можно воспользоваться *Пипеткой* в левой части экрана.
- Щелкните кнопку *Done*, чтобы сохранить параметры оптимизации в исходный документ

### **Практическая работа Photoshop 1\_4: Взвешенная оптимизация**

- Скачайте файл для работы.
- Откройте файл в редакторе. Выделите фрагмент картинке (цветок) *Прямолинейным Лассо* или любым другим инструментом.
- Сохраните выделение в альфа-канал (*Select → Save Selection (Выделение → Сохранить выделение)*). Дайте ему имя (введите в поле *Имя*).
- Откройте диалог «*Сохранить для Веб и устройств*», выберите «*JPEG*» в качестве формата на выходе и, если нужно, настройте уровень качества.
- Щёлкните кнопку «*маска*» по соседству с полем «*Качество*» (*Quality*). Открывается диалоговое окно.
- Выберите в меню «*Канал*» (*Channel*) только что сохранённый альфа-канал. Настройте «*Минимальное*» и / или «*Максимальное*» значения по вкусу и нажмите *Ok*.

### **Практическая работа Photoshop 1\_5: Настройка параметров вывода**

- В диалоговом окне «*Сохранить для Интернета и устройств*» в раскрывающемся меню «*Оптимизация*» выберите пункт «*Изменить настройки вывода*» (*Edit output settings*).
- Отметьте флажок *Вывод XHTML*: При экспорте создаются web-страницы, соответствующие стандарту XHTML.
- Щелкните по кнопке *Следующая* и настройте параметры в группе «*Фрагменты*».
- Настройте вывод фона щелкнув по *Следующая*.
- Щелкните *OK* и сохраните оптимизированный файл.
- Просмотрите HTML-код результирующего файла.

## **Практическая работа №24.**

### **Выбор доменного имени. Хостинг. Выбор хостинга**

**Хостинг** - это размещение и поддержка Web - страниц пользователя на сервере.

Почему возникает необходимость в хостинге? Предположим, Вы создали сайт на своем компьютере. Теперь перед Вами возникает следующий вопрос: как его разместить в Интернете, чтобы некий пользователь в любой момент времени мог получить к нему доступ? Разумеется, можно сделать так, чтобы из Интернета данную страничку любые желающие могли просматривать прямо с Вашего компьютера. Но это значит, что:

*во-первых*, ваш компьютер должен постоянно работать;

*во-вторых*, должен быть постоянно подключен к Интернету;

*в-третьих*, соединение с Интернетом должно происходить с большой скоростью, ведь если несколько человек сразу захотят посмотреть Ваш Web-сайт, то скорости обычного телефонного соединения будет недостаточно.

### **Виртуальный хостинг**

Если Web-сайт небольшой (например, размером 20-50 Мбайт), то для него не требуется установка и использование отдельного сервера (компьютера), наиболее эффективным и популярным решением является так называемый виртуальный сервер, *когда один сервер провайдера Интернет (хостинг-провайдера) обеспечивает функционирование нескольких (сотни тысячи) виртуальных Web-сайтов*. Эта услуга называется хостинг виртуальных серверов или просто **виртуальный хостинг**.

Виртуальный хостинг обеспечивает доступ к вашему Web-сайту любого пользователя Интернет в любое время из любой точки мира по электронному адресу, который был вами ранее зарегистрирован (доменное имя).

### **Физический хостинг (colocation)**

**Физический хостинг** - это размещение вашего компьютера (сервера) на территории телекоммуникационного узла (серверной) хостинг-провайдера, подключение компьютера к сети Интернет и обеспечение технического обслуживания со стороны провайдера[17].

**Размещение вашего собственного сервера у хостинг-провайдера** (это уже **будет называться не хостинг, а colocation**) обойдется в определенную ежемесячную абонентскую плату (она составляет обычно от 50 до 300 долларов) плюс оплата части входного трафика (обычно оплачивается только входящий западный трафик, но провайдер может брать деньги и за весь трафик).

**Основные отличия физического хостинга от виртуального** заключаются в следующем:

- отсутствие ограничений на используемое программное обеспечение и технологии (эти ограничения обязательно присутствуют при использовании виртуального хостинга);
- использование технических ресурсов сервера только для Вашего проекта (при виртуальном хостинге ресурсы находятся в общем пользовании всех виртуальных серверов);

### **Бесплатный хостинг**

Если Ваш проект не так уж глобален, чтобы платить деньги за его размещение в сети, можно воспользоваться "бесплатными" сервисами. Но не стоит забывать, что ничего бесплатного не бывает. Вы будете "расплачиваться" рекламой на вашем сайте (за которую, между прочим, бесплатные хостинговые компании получают деньги), низкой скоростью загрузки, перебоями в работе и т.д. и т.п.

Приведем примеры нескольких хостинг-провайдеров, которые предлагают **бесплатный хостинг**:

**www.iho.ru** - это качественный платный хостинг, предоставляющий на определенных условиях услуги хостинга бесплатно. С этими условиями можно ознакомиться на сайте хостинг провайдера [www.iho.ru](http://www.iho.ru) по адресу <http://www.iho.ru/discount.html>

**www.narod.ru** - один из самых первых хостингов в России. Обещают неограниченное пространство. Есть готовые шаблоны для начинающих web-мастеров или просто для тех, кто хочет сделать домашнюю страницу. Ставят свой баннер.

**www.wallst.ru** – "Неограниченное дисковое пространство, FTP доступ. Возможность устанавливать собственные CGI и PHP-скрипты". Открытие баз данных MySQL. Размещение рекламы.

У хостинг-провайдера ИНО.RU тарифные планы начинаются с \$4.5 в месяц. За эти деньги пользователь получает качественный быстрый хостинг! Но это уже платный хостинг.

### **Платный хостинг**

**Платный хостинг** - это коммерческая услуга аренды дискового пространства на сервере хостинг-провайдера. Нужна эта услуга для размещения интернет-представительства любого уровня в сети Интернет.[17]

Платный хостинг - это наилучший вариант публикации сайта, особенно если Вы планируете создать серьезный проект, на котором в дальнейшем собираетесь зарабатывать деньги. В этом случае необходим именно платный хостинг. Вы получите качественный хороший сервис, быструю загрузку сайта, на вашем сайте не будут размещать чужую рекламу... Услуги хостинга стоят от 4 и даже меньше долларов в месяц. Вы получаете хостинг с поддержкой программных языков: CGI, PHP, Perl, MySQL и т.д., что несомненно Вам понадобится [17] .

Значимость доменного имени в раскрутке сайта тоже важна. По возможности имя/адрес сайта должен быть коротким, запоминающимся и интуитивно понятным. URL такого домена наверняка быстрее отложится в памяти пользователя, нежели длинный и сложный адрес, предоставленный бесплатным хостингом.

Сайты расположенные на платных хостингах вызывают больше доверия. И если у Вас, например, Интернет-магазин или Вы что-либо продаёте со своего сайта, то покупатели будут Вам больше доверять, чем если бы Ваш сайт находился на бесплатном хостинге.

Платный хостинг предоставляет такие услуги, которые бесплатно предоставить нет никакой возможности - это в основном разнообразные инструменты для управления сайтом, анализа его посещаемости, необходимые популярные языки веб-программирования. Без этого набора хостинг-сервисов сегодня практически невозможно сделать бизнес-сайт, неважно что это будет: сайт-визитка с формой отправки сообщения владельцу либо крупный интернет-магазин.

Сегодня невозможно иметь фирменный сайт на бесплатном хостинге. Это выглядит настолько несерьезно, что в итоге может сильно ударить по престижу фирмы!

Сегодня плата за хостинг (за вычетом прибыли владельцу и содержания персонала) идет по трем направлениям:

1. Оплата коммуникаций - международных и местных информационных потоков (траффика)
2. Содержание материально-технической базы, аренда помещений в крупных центрах, администрирование серверов, мониторинг оборудования
3. Служба технической поддержки пользователей

**Задание.** Провести аналитический обзор серверов, предоставляющие платный/бесплатный хостинг, заполнить таблицы.

**Российские серверы, предоставляющие бесплатный хостинг:**

Название и адрес сервера	Имя вашего сайта	Место под сайт	Поддержка скриптов	Поддержка FTP	Дополнительные сервисы	Реклама хостера на вашей странице
<u>Narod.Ru</u>						
<u>Boom.Ru</u>						
<u>Chat.Ru</u>						
<u>By.Ru</u>						
<u>Agava.Ru</u>						
<u>NewMail.Ru</u>						
<u>Lgg.Ru</u>						
<u>Webservis.ru</u>						
<u>Halyava.Ru</u>						

**Стоимость разных видов хостинга у некоторых провайдеров (оказываемые услуги приводить в сокращенном виде):**

Название провайдера	Минимальный план		Средний план		Максимальный план		Collocation (размещение физических серверов)
	Услуги	Стоимость в месяц	Услуги	Стоимость в месяц	Услуги	Стоимость в месяц	Стоимость в месяц
<u>Corbina Telecom</u>							
<u>Zenon N.S.P</u>							
<u>MastakSoft</u>							
<u>350mb.Ru</u>							
<u>ARIADNA media</u>							
<u>Sorix</u>							
<u>Majordomo.Ru</u>							
<u>Highway.Ru</u>							
<u>Net.Ru</u>							
<u>Newcomport.Ru</u>							

### Практическая работа №25.

#### Перенос сайта с помощью FTP-клиента на хостинг

**Цель:** ознакомиться с условиями и возможностями публикации Web-сайтов в мировом информационном пространстве WWW, получить навыки публикации сайта через FTP-соединение

#### Публикация сайта через FTP-соединение

Для отправки файлов клиентам Web-сервером используется протокол HTTP (HyperText TransferProtocol — протокол передачи гипертекста). Существует еще один интернет-протокол — FTP (File TransferProtocol — протокол передачи файлов), используемый для передачи в сети файлов. Причем по FTP несложно организовать передачу файлов как от клиента серверу (используется специальный **FTP-сервер**), так и от сервера клиенту; инициатором передачи всегда выступает клиент.

Это происходит так: клиент с помощью специальной программы (называемой **FTP-клиентом**) устанавливает соединение с FTP-сервером и посылает ему или загружает с него файлы. Особенностью протокола FTP является то, что клиент работает с FTP-сервером как с локальным диском своего компьютера; он может создавать, изменять и удалять файлы и папки, копировать и переносить файлы куда угодно.

Эта особенность протокола FTP и легла в основу третьего и самого распространенного способа публикации Web-сайтов.

Заключается он в следующем. Администратор Web-сервера запускает и настраивает на компьютере программу FTP-сервера. После этого по запросу пользователя, которому нужно опубликовать на сервере свой сайт, он создает корневую папку для этого сайта. Пользователь с помощью программы *FTP-клиента* (например, CuteFTP) подключается к FTP-серверу и отправляет во вновь созданную корневую папку файлы своего сайта. После этого администратор конфигурирует Web-сервер так, чтобы он "увидел" новый сайт (для этого администратор записывает во внутренние списки сайтов Web-сервера путь к новой корневой папке и интернет-адрес нового сайта). Такова довольно грубая схема публикации сайта по третьему способу.

Для того чтобы подключиться к FTP-серверу, пользователю нужно ввести свои имя и пароль. Их пользователь получает у администратора сервера. Такая мера предосторожности не дает добраться до сайта злоумышленникам и просто посторонним людям. Любая программа FTP-клиента требует перед подключением ввести имя пользователя и пароль.

С помощью FTP-клиента пользователь может впоследствии изменять, переименовывать и удалять файлы и папки на сервере. То есть выполнять все операции с файлами и папками так, словно они находятся на жестком диске его компьютера. Однако помните, что администратор сервера может наложить некоторые ограничения на файловые операции, например, запретить удаление файлов в какой-либо папке.

Этот способ публикации сайтов используется на большинстве бесплатных и корпоративных Web-серверов.

### **Последовательность действий для осуществления публикации сайта на сервере по FTP-протоколу**

1. Пользователь регистрируется на бесплатном сервере, задает имя и пароль, под которыми он будет заходить на FTP-сервер. После этого на дисках сервера создается корневая папка его сайта, чье имя, как правило, совпадает с именем пользователя. Если пользователь публикует сайт на корпоративном сервере, имя и пароль он должен получить у администратора; администратор же вручную создает на диске сервера корневую папку сайта.

2. Пользователь с помощью программы FTP-клиента подключается к серверу и, как правило, сразу попадает в свою корневую папку. Если подключение не произошло, нужно проверить, правильно ли введены имя пользователя и (или) пароль. Если все введено правильно, но подключения все равно не происходит, пользователь должен обратиться к администратору сервера.

3. Пользователь загружает на сервер свой сайт. Если пользователю нужно что-либо изменить, он загружает на сервер только измененные файлы, а также манипулирует файлами на сервере: удаляет, переименовывает, копирует и переносит из папки в папку. Если изменения, которые необходимо сделать, значительны по объему, проще удалить все содержимое корневой папки и загрузить сайт заново.

4. Пользователь отключается от FTP-сервера. Обязательно сделайте это, ведь FTP-сервер для поддержания соединения с пользователем забирает ресурсы компьютера, которые всегда ограничены.

5. Пользователь запускает Web-обозреватель и тестирует свой сайт. Если обнаружатся ошибки, он вносит исправления и повторяет все шаги со 2 по 5.



Существуют программы, называемые Web-редакторами, обладающие функциями управления сайтом, т. е. в их среде предоставляется возможность публикации по протоколу FTP. К ним относятся - Macromedia Dreamweaver, Microsoft FrontPage — это хорошие программы, с помощью которых возможна публикация Web-сайтов. Более того, программное обеспечение этих программных продуктов ведет учет, какие файлы были изменены пользователем, и загружают на сервер только измененные файлы. Также с их помощью автоматически удаляются ненужные файлы на сервере. Пользователю необходимо будет только правильно ввести данные о FTP-сервере.

**Задание. Опубликовать свой сайт на выбранном FTP-сервере.**