



**ГБПОУ «Пермский политехнический колледж  
имени Н.Г. Славянова»**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ  
ДЛЯ ОБУЧАЮЩИХСЯ ПО ВЫПОЛНЕНИЮ  
ПРАКТИЧЕСКИХ РАБОТ**

для реализации Программы подготовки специалистов среднего звена  
по специальности

*09.02.06 Сетевое и системное администрирование*  
(технологический профиль профессионального образования)

**Рассмотрено и одобрено на заседании**  
Предметной цикловой комиссией  
*«Выпускающая студентов на  
государственную итоговую  
аттестацию*  
Протокол №2  
от 21 октября 2023 г.  
Председатель ПЦК

  
С.В. Вепрева

## СОДЕРЖАНИЕ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА.....	3
ПРИЛОЖЕНИЕ	
Методические указания для обучающихся по выполнению практических работ по учебным дисциплинам и междисциплинарным курсам .....	5

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Практические занятия относятся к основным видам учебных занятий и составляют важную часть теоретической и профессиональной практической подготовки, являются формой организации учебного процесса, направленной на выработку у обучающихся практических умений для изучения последующих учебных дисциплин, профессиональных модулей и для решения профессиональных задач.

Выполнение обучающимся практических занятий направлено на:

- обобщение, систематизацию, углубление, закрепление полученных теоретических знаний по конкретным темам учебных дисциплин профессиональных модулей;
- формирование умений применять полученные знания на практике, реализацию единства интеллектуальной и практической деятельности;
- развитие интеллектуальных умений у будущих специалистов: аналитических, проектировочных, конструктивных и др.;
- выработку при решении поставленных задач таких профессионально значимых качеств, как самостоятельность, ответственность, точность, творческая инициатива.

Практические занятия проводятся в учебных кабинетах, лабораториях, мастерских. Необходимыми структурными элементами практического занятия, помимо самостоятельной деятельности студентов, являются инструктаж, проводимый преподавателем, а также анализ и оценка выполненных работ и степени овладения студентами запланированными умениями.

Наряду с формированием умений и навыков в процессе практических занятий обобщаются, систематизируются, углубляются и конкретизируются теоретические знания, вырабатывается способность и готовность использовать теоретические знания на практике.

Содержание практического занятия определяется перечнем профессиональных умений по конкретной учебной дисциплине

(профессиональному модулю), а также характеристикой профессиональной деятельности выпускников, требованиями к результатам освоения основной профессиональной образовательной программы.

По каждой учебной дисциплине и междисциплинарному курсу для обучающихся разработаны методические указания по выполнению практических работ.

Работы, носящие репродуктивный характер, отличаются тем, что при их проведении студенты пользуются подробными инструкциями, в которых указаны: цель работы, пояснения (теория, основные характеристики), оборудование, аппаратура, материалы и их характеристики, порядок выполнения работы, таблицы, выводы (без формулировки), контрольные вопросы, учебная и специальная литература.

Работы, носящие частично поисковый характер, отличаются тем, что при их проведении студенты не пользуются подробными инструкциями, им не дан порядок выполнения необходимых действий, и требуют от студентов самостоятельного подбора оборудования, выбора способов выполнения работы в инструктивной и справочной литературе и др.

Работы, носящие поисковый характер, характеризуются тем, что студенты должны решить новую для них проблему, опираясь на имеющиеся у них теоретические знания.

Формы организации студентов на практических занятиях: фронтальная, групповая и индивидуальная.

При фронтальной форме организации занятий все студенты выполняют одновременно одну и ту же работу.

При групповой форме организации занятий одна и та же работа выполняется микро-группами по 2—5 человек.

При индивидуальной форме организации занятий каждый студент выполняет индивидуальное задание.

Оценки за выполнение практических работ являются показателями текущей успеваемости студентов по учебной дисциплине.

## ПРИЛОЖЕНИЕ

### Методические указания для обучающихся по выполнению практических работ по учебным дисциплинам и междисциплинарным курсам

Код	Наименование учебной дисциплины, профессионального модуля, междисциплинарного курса	№ Приложения
ОУД.01	Русский язык	1
ОУД.02	Литература	2
ОУД.03	Иностранный язык	3
ОУД.04	История	4
ОУД.05	Обществознание	5
ОУД.06	География	6
ОУД.07	Химия	7
ОУД.08	Биология	8
ОУД.09	Физическая культура	9
ОУД.10	Основы безопасности жизнедеятельности	10
ОУД.11	Математика	11
ОУД.12	Информатика	12
ОУД.13	Физика	13
ОУД.14	Основы исследовательской и проектной деятельности	14
ОУД.15	Введение в специальность	15
СГ.01	История России	16
СГ.02	Иностранный язык в профессиональной деятельности	17
СГ.03	Безопасность жизнедеятельности	18
СГ.04	Физическая культура	19
СГ.04	Адаптивная физическая культура	20
СГ.05	Основы бережливого производства	21
СГ.06	Основы финансовой грамотности	22
ОП.01	Элементы высшей математики	23
ОП.02	Дискретная математика с элементами математической логики	24
ОП.03	Теория вероятностей и математическая статистика	25
ОП.04	Основы алгоритмизации и программирования	26
ОП.05	Основы проектирования баз данных	27
ОП.06	Архитектура аппаратных средств	28
ОП.07	Операционные системы и среды	29
ОП.08	Информационные технологии	30
ОП.09	Правовое обеспечение профессиональной деятельности	31
ОП.10	Стандартизация, сертификация и техническое документоведение	32
ОП.11	Основы электротехники	33
ОП.12	Инженерная компьютерная графика	34
ОП.13	Технологии физического уровня передачи данных	35
МДК.01.01	Организация, принципы построения и функционирования компьютерных сетей	36

МДК.01.02	Настройка и техническое обслуживание объектов сетевой инфраструктуры	37
МДК.02.01	Администрирование сетевых операционных систем	38
МДК.02.02	Программное обеспечение компьютерных сетей	39
МДК.02.03	Организация администрирования компьютерных систем	40
МДК.03.01	Компьютерные сети	41
МДК.03.02	Безопасность компьютерных сетей	42
МДК.04.01	Проектирование и наладка беспроводных сетей	43
МДК.05.01	Веб-программирование	44

**Методические указания  
для обучающихся по выполнению практических  
работ по МДК.05.01 Веб-программирование**

**Автор: Кадочникова  
Наталья Владимировна,**  
ГБПОУ «Пермский  
политехнический колледж  
имени Н.Г. Славянова»,  
преподаватель высшей  
квалификационной категории

**СОДЕРЖАНИЕ**

<b>1</b>	<b>Пояснительная записка</b>	<b>3</b>
<b>2</b>	<b>Содержание практических работ</b>	<b>4</b>
	<b>Практическая работа №1 «Создание структуры Web-страницы»</b>	<b>4</b>
	<b>Практическая работа №2 «Форматирование текста»</b>	<b>5</b>
	<b>Практическая работа №3 «Вставка изображения»</b>	<b>7</b>
	<b>Практическая работа №4 «Создание гиперссылки»</b>	<b>9</b>
	<b>Практическая работа №5 «Создание списков»</b>	<b>14</b>
	<b>Практическая работа №6 «Создание простейших таблиц. Форматирование таблиц»</b>	<b>16</b>
	<b>Практическая работа №7 «Создание сложных таблиц»</b>	<b>18</b>
	<b>Практическая работа №8 «Создание форм»</b>	<b>20</b>
	<b>Практическая работа №9 «Разработка фреймов. Разработка плавающих фреймов»</b>	<b>21</b>
	<b>Практическая работа №10 «Внедрение элементов управления. Разработка макета сайта. Группировка элементов управления»</b>	<b>25</b>
	<b>Практическая работа №11 «Применение CSS»</b>	<b>32</b>
	<b>Практическая работа №12 «Простые скрипты JavaScript»</b>	<b>34</b>
	<b>Практическая работа №13 «Использование функций»</b>	<b>38</b>
	<b>Практическая работа №14 «Использование обработчиков событий»</b>	<b>46</b>
	<b>Практическая работа №15 «Использование условного оператора IF, SWITCH»</b>	<b>50</b>
	<b>Практическая работа №16 «Использование операторов цикла»</b>	<b>52</b>
	<b>Практическая работа №17 «Работа с датой и временем»</b>	<b>55</b>
	<b>Практическая работа №18 «Работа со строками. Решение задач»</b>	<b>57</b>
	<b>Практическая работа №19 «Установка и настройка ПО. Операторы в языке PHP. Создание формы. Обработка массивов данных. Работа со строками. Работа с файлами. Проверка данных. Обработка запросов с помощью PHP»</b>	<b>60</b>
<b>3</b>	<b>Список источников и литературы</b>	<b>75</b>

## Пояснительная записка

Методические указания по выполнению практических работ обучающихся по МДК.05.01 «Веб-программирование» предназначены для обучающихся по специальности 09.02.06.

Цель методических указаний: оказание помощи обучающимся в выполнении практических работ по МДК.05.01 «Веб-программирование».

Настоящие методические указания содержат работы, которые позволят обучающимся закрепить теоретические знания, сформировать необходимые умения и навыки деятельности по специальности 09.02.06, направлены на формирование следующих компетенций:

ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам

ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности

ОК 03 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях

ОК 04 Эффективно взаимодействовать и работать в коллективе и команде

ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста

ОК 06 Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения

ОК 07 Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях

ОК 08 Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности

ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках

ПК 5.1. Верстка веб-страниц

ПК 5.2. Кодирование на языках веб-программирования.

В результате выполнения практических работ по МДК.05.01 «Веб-программирование» обучающиеся должны:

**иметь практический опыт:**

- создания Web-сайтов на основе языков и редакторов HTML;
- создания тестов на основе шаблонов;
- применения нормативно-технической документации;

**уметь:**

- создавать Web-сайты с помощью языков и редакторов HTML;
- читать HTML-коды в шаблонах;

**знать:**

- основные принципы и этапы разработки Web-сайтов;
- основные структуры языка HTML;
- основные принципы работы в редакторах HTML;
- нормативно-техническую документацию: инструкции, регламенты, процедуры, технические условия и нормативы.

Описание каждой практической работы содержит: раздел, тему, количество часов, цели работы, что должен знать и уметь обучающийся, теоретическую часть,



порядок выполнения работы, контрольные вопросы, учебно-методическое и информационное обеспечение.

На выполнение практических работ по МДК.05.01 «Веб-программирование» отводится *74 часа*.

# Содержание практических работ

## Практическая работа №1

### «Создание структуры Web-страницы»

**Тема:** Язык HTML

**Количество часов:** 2

**Цели:** создания Web-сайтов на основе языков и редакторов HTML

**Задачи:** создавать Web-сайты с помощью языков и редакторов HTML, читать HTML-коды

**Теоретическая часть:**

Вид Web-страницы задаётся тэгами, которые заключаются в угловые скобки.

**Тэги:**

1. одиночные
2. парные: открывающийся и закрывающийся (перед обозначением /) парные называются **контейнером**.

Тэги записываются прописными или строчными буквами.

<HTML> </HTML>-начало и конец страницы, без этих тэгов браузер не определит формат документа.

Web-страница делится на 2 логические части: заголовок и содержание.

<HEAD> </HEAD>-заголовок Web-страницы: содержит название и справочную информацию о странице.

<TITLE> </TITLE>-название Web-страницы, отражается в строке заголовка браузера.

<BODY> </BODY>-основное содержание Web-страницы. Может включать в себя текст, таблицу, бегущие строки, ссылки на графику и звук.

Например:

```
<HTML>
<HEAD>
<TITLE> компьютер </TITLE>
</HEAD>
<BODY>
всё о компьютере
</BODY>
</HTML>
```

WEB-страницу необходимо сохранить в виде файла с расширением htm или .html.

Титульный файл сайта принято сохранять под именем index.htm

Следует различать **имя файла**, под которым Web-сайт хранится в файловой системе и **имя Web страницы**, которое в первую очередь анализируется поисковыми системами.

**Порядок выполнения работы:**

Создать 3 WEB-страницы:

Компьютер

Сохранить как index.html

Информатика

Сохранить как inform.html

Пример

Сохранить как proba.html

Web-страницы содержат только названия страниц. Содержание пустое. Написать HTML-коды страниц в тетради

**Критерии оценки за практическую работу:**

1. содержание Web-страницы соответствует заданию
2. текст на Web-странице не содержит ошибок
3. прописано название Web-страницы
4. правильный html-код
5. правильное название файла
6. правильное расположение файла на Личном диске студента

<b>5</b>	Пункты 1-6 верны
<b>4</b>	Один или два пункта из 1-4 содержат неточности. Пункты 5,6 верны
<b>3</b>	Три и более пунктов из 1-4 содержат неточности. Пункты 5,6 верны
<b>2</b>	Пункты 5,6 не верны или Пункты 1-4 не верны

## Практическая работа №2 «Форматирование текста»

**Тема:** Язык HTML

**Количество часов:** 4

**Цели:** создания Web-сайтов на основе языков и редакторов HTML

**Задачи:** создавать Web-сайты с помощью языков и редакторов HTML, читать HTML-коды

**Теоретическая часть:**

С помощью HTML-тэгов можно задавать различные параметры форматирования текста.

	Тег	Атрибут, значение	Описание
<b>1</b>	<Hn> </Hn> n=1..6	размер шрифта для заголовков в тексте	
		align =left	выравнивание слева
		align =right	выравнивание справа
<b>2</b>	<FONT> </FONT>	align =center	выравнивание по центру
		параметры форматирования шрифта	
		face=Arial	гарнитура шрифта
		size=4	размер шрифта
<b>3</b>	<B> </B>	color=blue	цвет шрифта
		color="'#0000FF'"	цвет шрифта (RGB-код)
<b>4</b>	<I> </I>	начертание – жирный	
<b>5</b>	<U> </U>	начертание – курсив	
<b>6</b>	<S> </S>	подчёркивание	
<b>7</b>	<SUP> </SUP>	перечёркнутый текст	
<b>8</b>	<SUB> </SUB>	верхний индекс	
<b>9</b>	<TT> </TT>	нижний индекс	
<b>10</b>	<PRE> </PRE>	моноширинный шрифт	
<b>11</b>	<HR>	обрамляет предварительно отформатированный текст	
		горизонтальная линия	
		align	выравнивание
		noshade	плоская линейка (без значения)
<b>12</b>	<P> </P>	size	толщина линии (в %)
		width	ширина линейки (в пикселях или в %)
<b>12</b>	<P> </P>	разделение текста на абзацы	
		align	выравнивание

		title	текст подсказка
13	<CENTER> </CENTER>	выравнивание по центру	
14	<BODY> </BODY>	содержание Web-страницы	
		text=red background=fon.jpg bgcolor=pink bgcolor="#RRGGBB"	цвет текста на Web-странице фоновое изображение цвет фона

### Порядок выполнения работы:

Задание 1
<div style="text-align: center;"> <h2>Компьютер</h2> </div> <div> Информатика<sup>2</sup> </div> <hr/> <div style="text-align: center;"> Информация<sub>1</sub> </div> <hr/> <div style="text-align: right;"> Знания Умения Факты </div> <hr/>

Добавить фон страницы  
Все слова разных цветов  
Сохранить Z1.html

Задание 2	параметры								
<p><i>Государственного бюджетного профессионального образовательного учреждения Пермский политехнический колледж имени Н.Г. Славянова</i></p> <hr/> <p>специальность</p> <p><b>09.02.01 Компьютерные системы и комплексы</b></p> <p>программа базовой подготовки</p> <p><b>Квалификация:</b> техник по компьютерным системам</p> <p><b>Форма обучения</b> - очная</p> <p><b>Нормативный срок обучения</b> 3 г. 10 мес. на базе основного общего образования</p> <p>Профиль получаемого профессионального образования <b>технический</b></p> <hr/> <p><b>План учебного процесса</b></p> <table> <tr> <td>Общеобразовательный цикл</td> <td><b>100 часов</b></td> </tr> <tr> <td>Общий гуманитарный и социально-экономический цикл</td> <td><b>100 часов</b></td> </tr> <tr> <td>Математический и общий естественнонаучный цикл</td> <td><b>100 часов</b></td> </tr> <tr> <td>Профессиональный цикл</td> <td><b>100 часов</b></td> </tr> </table>	Общеобразовательный цикл	<b>100 часов</b>	Общий гуманитарный и социально-экономический цикл	<b>100 часов</b>	Математический и общий естественнонаучный цикл	<b>100 часов</b>	Профессиональный цикл	<b>100 часов</b>	<p>Шрифт Arial</p> <p>Шрифт Tahoma</p> <p>Шрифт Verdana</p>
Общеобразовательный цикл	<b>100 часов</b>								
Общий гуманитарный и социально-экономический цикл	<b>100 часов</b>								
Математический и общий естественнонаучный цикл	<b>100 часов</b>								
Профессиональный цикл	<b>100 часов</b>								
<p>Применить параметры Добавить фоновое изображение Соблюдать цвета Сохранить Z2.html</p>									

### Критерии оценки за практическую работу:

1. содержание Web-страницы соответствует заданию
2. текст на Web-странице не содержит ошибок
3. прописано название Web-страницы
4. правильный html-код
5. правильное название файла
6. правильное расположение файла на Личном диске студента

<b>5</b>	Пункты 1-6 верны
<b>4</b>	Один или два пункта из 1-4 содержат неточности. Пункты 5,6 верны
<b>3</b>	Три и более пунктов из 1-4 содержат неточности. Пункты 5,6 верны
<b>2</b>	Пункты 5,6 не верны или Пункты 1-4 не верны

### Практическая работа №3 «Вставка изображения»

**Тема:** Язык HTML

**Количество часов:** 2

**Цели:** создания Web-сайтов на основе языков и редакторов HTML

**Задачи:** создавать Web-сайты с помощью языков и редакторов HTML, читать HTML-коды

**Теоретическая часть:**

На Web-страницах могут размещаться графические файлы форматов GIF, JPG, PNG. Преобразование может выполняться с помощью графических редакторов Photo Editor, Photoshop. На титульном листе целесообразно помещать картинку посвященную теме сайта.

	Тег	Атрибут, значение	Описание
<b>1</b>	<Img>	вставки графических изображений	
		src	место хранения файла на локальном компьютере или в интернете

<IMG SRC = "computer.gif "> файл на локальном компьютере в том же каталоге.

<IMG SRC = "c:\ computer\computer.gif "> в другом каталоге локального компьютера.

<IMC SRC = "http://www.server.ru/computer.gif "> на удаленном сервере в Internete.

**Иллюстрации** – неотъемлемая часть дизайна Web-страницы. Однако в целях экономии времени отключают в браузере загрузку графического файла и читают лишь тексты, поэтому имеет смысл на месте рисунка выводить соответствующий текст, а потом изображения.

	Тег	Атрибут, значение	Описание
<b>1</b>	<Img>	alt – альтернативный текст title – текст подсказка border – толщина границы вокруг рисунка height – высота изображения width – ширина изображения vspace – расстояние между рисунком и текстом сверху и снизу hspace – расстояние между рисунком и текстом справа и слева	


<IMC SRC ="computer.gif " ALT ="компьютер">

	Тег	Атрибут, значение	Описание
<b>1</b>	<Img>	Взаимное расположение графического объекта и текста	
		align=top вверх align=bottom вниз align=left слева align= right справа align= middle по середине dynsrc=123.avi вставка видео loop количество повторений (0 или 1)	


		height    высота видеовставки
		width    ширина видеовставки

### **Порядок выполнения работы:**


**Задание 1:** создать Web-страницы, содержащие графические изображения

Рисунок 1	
	Вставить рисунок, расположить его слева

Сохранить как ris1.html

Рисунок 2	
Вставить рисунок, расположить его справа	

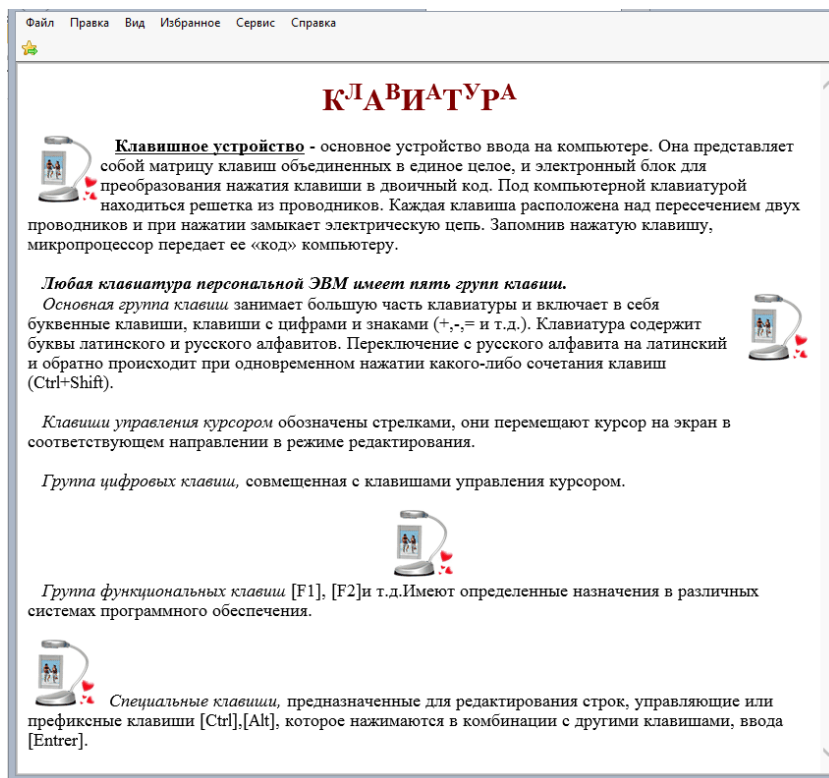
Сохранить как ris2.html

Рисунок 3	
	
Вставить рисунок, расположить по центру	

Сохранить как ris3.html

**Задание 2:** создать Web-страницу, содержащие графические изображения и текст.

Сохранить ris4.html



### Критерии оценки за практическую работу:

1. содержание Web-страницы соответствует заданию
2. текст на Web-странице не содержит ошибок
3. прописано название Web-страницы
4. правильный html-код
5. правильное название файла
6. правильное расположение файла на Личном диске студента

<b>5</b>	Пункты 1-6 верны
<b>4</b>	Один или два пункта из 1-4 содержат неточности. Пункты 5,6 верны
<b>3</b>	Три и более пунктов из 1-4 содержат неточности. Пункты 5,6 верны
<b>2</b>	Пункты 5,6 не верны или Пункты 1-4 не верны

### Практическая работа №4 «Создание гиперссылки»

**Тема:** Язык HTML

**Количество часов:** 2

**Цели:** создания Web-сайтов на основе языков и редакторов HTML

**Задачи:** создавать Web-сайты с помощью языков и редакторов HTML, читать HTML-коды

#### Теоретическая часть:

Первая титульная страница сайта должна иметь возможность перемещаться по сайту, т. е. должны быть организованы гиперссылками на другие Web-страницы. Для создания гиперсвязей необходимо создать Web-страницы, которые должны иметь заголовки, но пока могут быть без содержания.

Гиперссылка состоит из 2-х частей:

1. указатель ссылки – это то, что мы видим на Web-странице (текст, рисунок).  
Стандартно указатель ссылки выделяется синим цветом и подчеркивается.
2. адресная часть ссылки – это адрес документа, на который указывает ссылка.

Адрес ссылки может быть **абсолютным** и **относительным**.

**Абсолютный:** полностью определяет компьютер, каталог и файл, в котором расположен документ.

**C:/web – сайт/file name.htm** абсолютный адрес на локальном компьютере.

**http://www.host.ru/web-сайт/file name.htm** абсолютный адрес на удаленном компьютере.

**Относительный** адрес указывает на местоположение документа относительно того, в котором находится указатель ссылки.

При разработке сайтов рекомендуется входящие Web-страницы связывать относительными ссылками, чтобы не изменять адресную часть ссылки при перемещении сайта на другой локальный компьютер или публикации в Internet.

Для представления гиперссылок удобно выбрать названия страниц. Принято размещать гиперссылки в нижней части страницы, в одной строке, разделённых пробелами (&nbsp;). Такое размещение называется **панелью навигации**.

	Тег	Атрибут, значение	Описание
1	<A> </A>		Гиперссылка на другие документы или часть текущего документа
		href	Адресная часть
2	<A href="mailto:EMAIL"></A>		Гиперссылка вызова почтовой программы для написания письма автору документа
3	<A name="NAME"> </A>		Отмечает часть текста как цель для гиперссылок в документе
4	<A href="#NAME"> </A>		Гиперссылка на часть текущего документа

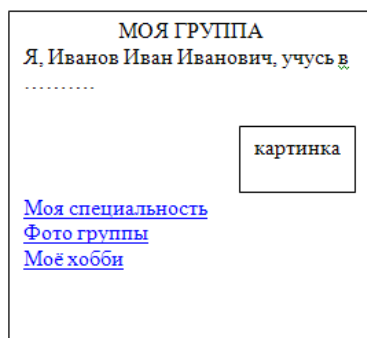
#### Порядок выполнения работы:

Задание 1: создать папку «Моя группа» (4 Web-страницы).



Создать в программе БЛОКНОТ 4 файла с именами:

- Index.html
- Specialnost.html
- Foto.html
- Hobby.html



Index.html

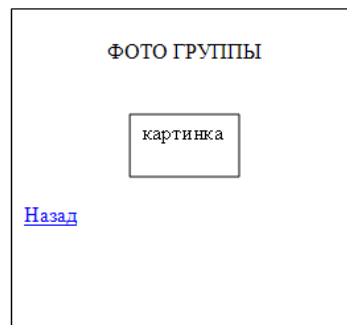
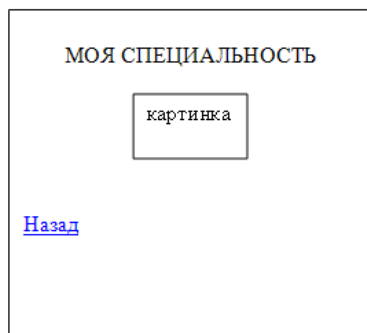
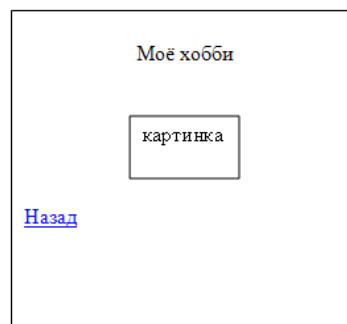


Foto.html



Specialnost.html



Hobby.html

Задание 2: создать гиперссылки на одной странице. Файл сохранить под именем GS2.html

## СОДЕРЖАНИЕ

[Информационное письмо 1](#)

[Информационное письмо 2](#)

[Информационное письмо 3](#)

[Информационное письмо 4](#)

[Информационное письмо 5](#)

### Информационное письмо 1

Методология планирования материальных ресурсов производства (MRP) обеспечивает ситуацию, когда каждый элемент производства, каждая комплектующая деталь находится в нужное время в нужном количестве (рис. 4.1)

На основании входных данных MRP-система выполняет следующие операции:

Состав изделия и описание материалов

Производственный график работы

MRP

План заказов и коррективы к нему

Отчеты о планировании,

Рис. 4.1. Структурная схема MRP

- Определяет количество конечных изделий для каждого периода времени планирования;
- К составу конечных изделий добавляются запасные части;
- Определяется общая потребность в материальных ресурсах в соответствии с ведомостью материалов и составом изделия;
- Общая потребность материалов корректируется с учетом состояния запасов для каждого периода времени планирования;
- Осуществляется формирование заказов на пополнение запасов с учетом необходимого времени опережения.

[вверх](#)

## Информационное письмо 2

Методология планирования материальных ресурсов производства (MRP) обеспечивает ситуацию, когда каждый элемент производства, каждая комплектующая деталь находится в нужное время в нужном количестве (рис. 4.1)



Рис. 4.1. Структурная схема MRP

- Определяет количество конечных данных изделий для каждого периода времени планирования;
- К составу конечных изделий добавляются запасные части;
- Определяется общая потребность в материальных ресурсах в соответствии с ведомостью материалов и составом изделия;
- Общая потребность материалов корректируется с учетом состояния запасов для каждого периода времени планирования;
- Осуществляется формирование заказов на пополнение запасов с учетом необходимого времени опережения.

[вверх](#)

## Информационное письмо 3

Методология планирования материальных ресурсов производства (MRP) обеспечивает ситуацию, когда каждый элемент производства, каждая комплектующая деталь находится в нужное время в нужном количестве (рис. 4.1)



Рис. 4.1. Структурная схема MRP

- Определяет количество конечных данных изделий для каждого периода времени планирования;
- К составу конечных изделий добавляются запасные части;
- Определяется общая потребность в материальных ресурсах в соответствии с ведомостью материалов и составом изделия;
- Общая потребность материалов корректируется с учетом состояния запасов для каждого периода времени планирования;
- Осуществляется формирование заказов на пополнение запасов с учетом необходимого времени опережения.

## Информационное письмо 4

Методология планирования материальных ресурсов производства (MRP) обеспечивает ситуацию, когда каждый элемент производства, каждая комплектующая деталь находится в нужное время в нужном количестве (рис. 4.1)



Рис. 4.1. Структурная схема MRP

- Определяет количество конечных данных изделий для каждого периода времени планирования;
- К составу конечных изделий добавляются запасные части;
- Определяется общая потребность в материальных ресурсах в соответствии с ведомостью материалов и составом изделия;
- Общая потребность материалов корректируется с учетом состояния запасов для каждого периода времени планирования;
- Осуществляется формирование заказов на пополнение запасов с учетом необходимого времени опережения.

[вверх](#)

## Информационное письмо 5

Методология планирования материальных ресурсов производства (MRP) обеспечивает ситуацию, когда каждый элемент производства, каждая комплектующая деталь находится в нужное время в нужном количестве (рис. 4.1)



Рис. 4.1. Структурная схема MRP

- Определяет количество конечных данных изделий для каждого периода времени планирования;
- К составу конечных изделий добавляются запасные части;
- Определяется общая потребность в материальных ресурсах в соответствии с ведомостью материалов и составом изделия;
- Общая потребность материалов корректируется с учетом состояния запасов для каждого периода времени планирования;
- Осуществляется формирование заказов на пополнение запасов с учетом необходимого времени опережения.

[вверх](#)

Задание 3: создать гиперссылки на одной странице. Файл сохранить под именем GS2.html

## Содержание

### Введение в HTML

### Инструментарий

### Пару слов про голову документа

## Введение в HTML

Чтобы сразу же ввести в курс дела нетерпеливых читателей, предложим им возможность создания веб-страницы без последовательного изучения правил HTML. По крайней мере, вы сумеете убедиться, что создание веб-страниц достаточно простая штука.

Чтобы посмотреть результат примера в действии, проделайте следующие шаги.

1. В **Windows** откройте программу Блокнот (Пуск> Выполнить> набрать «**notepad**» или Пуск> Программы> Стандартные> Блокнот).
2. Наберите или скопируйте код в Блокноте (рис. 1.1).

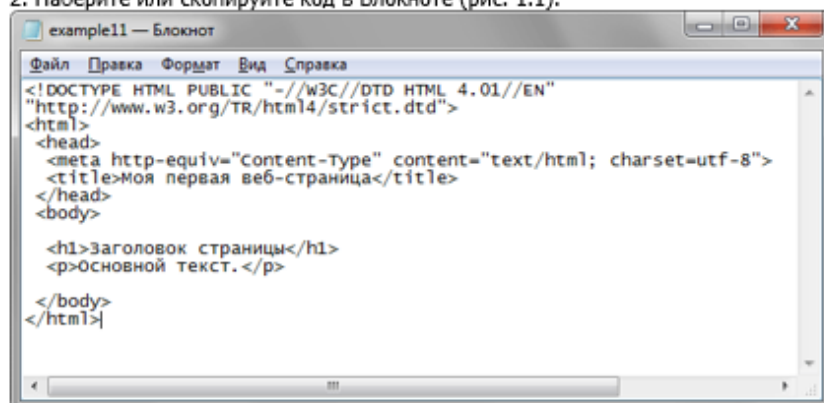


Рис. 1.1. Вид HTML-кода в программе Блокнот

6. Если все сделано правильно, то в браузере вы увидите результат, как показано на рис.1.3.

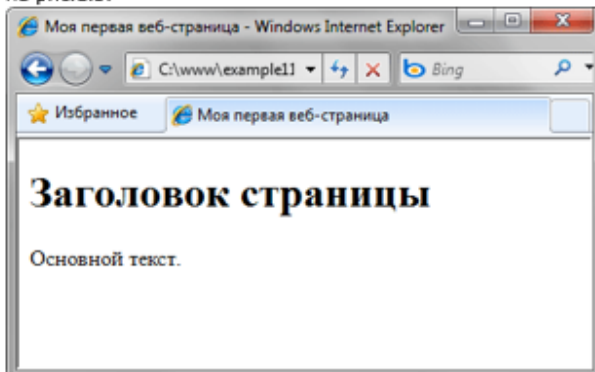


Рис. 1.3. Вид веб-страницы в окне браузера

[К содержанию](#)

## Инструментарий

Для эффективной работы не обойтись без необходимых и привычных инструментов, в том числе и при написании кода HTML. Поэтому для начальной разработки веб-страниц или даже небольшого сайта — так называется набор страниц, связанных между собой ссылками и единым оформлением, нам понадобятся следующие программы.

#### Валидатор

Валидация HTML-документа предназначена для выявления ошибок в синтаксисе веб-страницы и расхождений со спецификацией HTML. Соответственно, программа или система для такой проверки называется валидатором.

Как проверить HTML-файл на валидность.

Если есть доступ в Интернет, то следует зайти по адресу <http://validator.w3.org> и ввести путь к проверяемому документу или сайту в специальной форме. После проверки будут показаны возможные ошибки или появится надпись, что документ прошел валидацию успешно.

#### Tidy

Для проверки локального HTML-файла или при отсутствии подключения к Интернету, предназначена программа Tidy. Некоторые редакторы, например, PSPad, уже содержат встроенный Tidy и валидацию документа можно провести без дополнительных средств.

#### Графический редактор

Графический редактор необходим для обработки изображений и их подготовки для публикации на веб-странице. Самой популярной программой такого рода является Photoshop, ставший стандартом для обработки фотографий и создания графических изображений для сайтов. Но в большинстве случаев мощь Photoshop-а избыточна, и лучше воспользоваться чем-нибудь более простым и проворным. В частности, программа Paint.Net позволяет сделать все необходимые манипуляции с изображениями, добавок бесплатна для использования.

[К содержанию](#)

### Критерии оценки за практическую работу:

1. содержание Web-страницы соответствует заданию
2. текст на Web-странице не содержит ошибок
3. прописано название Web-страницы
4. правильный html-код
5. гиперссылки работают верно
6. правильное название файла
7. правильное расположение файла на Личном диске студента

5	Пункты 1-7 верны
4	Один или два пункта из 1-5 содержат неточности. Пункты 6,7 верны
3	Три и более пунктов из 1-5 содержат неточности. Пункты 6,7 верны
2	Пункты 6,7 не верны или Пункты 1-5 не верны

### Практическая работа №5 «Создание списков»

**Тема:** Язык HTML

**Количество часов:** 4

**Цели:** создания Web-сайтов на основе языков и редакторов HTML

**Задачи:** создавать Web-сайты с помощью языков и редакторов HTML, читать HTML-коды

#### Теоретическая часть:

Списки используются для размещения текста на Web-страницах.

*Варианты списков*

1. нумерованные
2. ненумерованные (маркированные).
3. списки определений
4. вложенные списки

	Тег	Атрибут, значение	Описание
1	<OL> </OL>	нумерованный список	
		type=a type=I type=1 (по умолчанию)	тип нумерации
		start	номер первого элемента списка
		compact	более компактный список
	<LI>	определяет каждый элемент списка и присваивает номер	
2	<OL> </OL>	ненумерованный список	
		type=circle type=disc type=square	тип маркера
	<LI>	определяет каждый элемент списка и присваивает номер	
3	<DL></DL>	список определений.	
	<DT>	определяет каждый из терминов списка	
	<DD>	описывает каждое определение	
4	<UL> </UL>	вложенный список	
		type	тип нумерации, маркера
	<LI>	определяет каждый элемент списка и присваивает номер	

### Порядок выполнения работы:

**Задание №1** нумерованные и ненумерованные (маркированные) списки

<b>I. Глава 1</b>	
1	<a href="#">Информация</a>
2	<a href="#">Виды информации</a>
3	<a href="#">Действия с информацией:</a>
	▪ накопление;
	▪ передача;
	▪ хранение;
	▪ обработка.
<b>II. Глава 2</b>	
1	<a href="#">Информационные технологии</a>
2	<a href="#">Информационные системы</a>

Сохранить spisok1.html

**Задание №2** списки определений

<b>Текстовый процессор</b>
программа для создания, редактирование, форматирования и макетирования текстового документа.
<b>Макет текстового документа</b>
совокупность упорядоченных по определённым правилам объектов, размещённых на странице.

Сохранить spisok2.html

### Задание №3 вложенные списки

#### Введение

#### Освоение системной среды Windows

1. Параметры файла
2. Работа с окнами
3. Настройка параметров рабочего стола
4. Антивирусная защита дисков

#### Освоение среды графического редактора

5. Общая характеристика
6. Создание и редактирование рисунка

#### Освоение среды текстового редактора

7. Создание текста
8. Редактирование текста
9. Форматирование текста

#### Освоение среды табличного редактора

10. Создание таблиц
11. Использование встроенных функций
12. Построение диаграмм

Сохранить spisok3.html

#### Критерии оценки за практическую работу:

1. содержание Web-страницы соответствует заданию
2. текст на Web-странице не содержит ошибок
3. прописано название Web-страницы
4. правильный html-код
5. гиперссылки работают верно
6. правильное название файла
7. правильное расположение файла на Личном диске студента

<b>5</b>	Пункты 1-7 верны
<b>4</b>	Один или два пункта из 1-5 содержат неточности. Пункты 6,7 верны
<b>3</b>	Три и более пунктов из 1-5 содержат неточности. Пункты 6,7 верны
<b>2</b>	Пункты 6,7 не верны или Пункты 1-5 не верны

### Практическая работа №6

#### «Создание простейших таблиц. Форматирование таблиц»

**Тема:** Язык HTML

**Количество часов:** 6

**Цели:** создания Web-сайтов на основе языков и редакторов HTML

**Задачи:** создавать таблицы в HTML

#### Теоретическая часть:

В практике Web-разработок таблицы очень широко используются для формирования шаблонов страниц, т.е. создаётся «невидимая» таблица, у которой отсутствуют линии сетки. В ячейках таблицы располагаются другие элементы Web-страницы или вложенные таблицы.

	Тег	Атрибут, значение	Описание
<b>1</b>	<TABLE> </TABLE>	таблица	
		align	выравнивание таблицы
		bgcolor	цвет фона
		border	граница
		bordercolor	цвет границы
		background	фоновый рисунок
		title	подсказка
		cellpadding	расстояние между границей и текстом в ячейке
		cellspacing	расстояние между ячейками
		frame= above	отображается верхняя граница таблицы
		frame= below	нижняя граница
		frame= hside	верхняя и нижняя
		frame= vside	правая и левая
		frame= lhs	левая
		frame= rhs	правая
		height	высота
		width	ширина
<b>2</b>	<CAPTION> </CAPTION>	заголовок таблицы	
		align=top (bottom, left, right)	выравнивание
<b>3</b>	<TR> </TR>	строка таблицы	
		align=left ( right, center, justify)	горизонтальное выравнивание
		valign=top (bottom, middle)	вертикальное выравнивание
		bgcolor	цвет фона строки
		bordercolor	цвет рамки строки
		height	высота строки
		width	ширина строки
<b>4</b>	<TD> </TD>	ячейка таблицы	
		align=left ( right, center, justify)	горизонтальное выравнивание
		valign=top (bottom, middle)	вертикальное выравнивание
		bgcolor	цвет фона ячейки
		bordercolor	цвет рамки ячейки
		colspan	количество столбцов для объединения
		rowspan	количество строк для объединения
<b>5</b>	<TH> </TH>	ячейки-заголовки (вместо <TD> </TD>)	

**Порядок выполнения работы:**

**Задание №1** (table1.html)

Закрасить клетки с текстом

A	B	C	D
E		F	
	G		H

A	B
	C
D	
E	F
	G
H	

Закрасить клетки с текстом

Закрасить клетки в первой и третьей строке

A			
B		C	
D	E	F	G

A	B	C	D
E		F	
G			

Закрасить клетки в первой и третьей строке

Задание №2 (table2.html)

Первый	Четвёртый	Седьмой
Второй	Пятый	Восьмой
Третий	Шестой	Девятый

Понедельник	Четверг
Вторник	Пятница
Среда	Суббота

Задание №3 (table3.html)

## Продажа билетов

Пункт	Январь	Февраль	Март	Апрель
Север	100	200	300	400
Юг	200	300	400	100
Запад	300	400	100	200
Восток	400	100	200	300
<b>ИТОГО</b>	<b>1000</b>	<b>1000</b>	<b>1000</b>	<b>1000</b>

Задание №4 (table4.html)

Сделать копию HTML-кода из задания Таблица 2 и внести изменения.

Пункт	Январь	Февраль	Март	Апрель
Север	100			
Юг	400	300	200	
Запад		400		
Восток		300		
ИТОГО	1000	1000	1000	1000

Критерии оценки за практическую работу:

1. содержание Web-страницы соответствует заданию
2. текст на Web-странице не содержит ошибок
3. прописано название Web-страницы



4. правильный html-код
5. гиперссылки работают верно
6. правильное название файла
7. правильное расположение файла на Личном диске студента

<b>5</b>	Пункты 1-7 верны
<b>4</b>	Один или два пункта из 1-5 содержат неточности. Пункты 6,7 верны
<b>3</b>	Три и более пунктов из 1-5 содержат неточности. Пункты 6,7 верны
<b>2</b>	Пункты 6,7 не верны или Пункты 1-5 не верны

### Практическая работа №7 «Создание сложных таблиц»

**Тема:** Язык HTML

**Количество часов:** 4

**Цели:** создания Web-сайтов на основе языков и редакторов HTML

**Задачи:** создавать встроенных таблиц в HTML

**Теоретическая часть:**

В практике Web-разработок таблицы очень широко используются для формирования шаблонов страниц, т.е. создаётся «невидимая» таблица, у которой отсутствуют линии сетки. В ячейках таблицы располагаются другие элементы Web-страницы или вложенные таблицы.

	Тег	Атрибут, значение	Описание
<b>1</b>	<TABLE> </TABLE>	таблица	
		align	выравнивание таблицы
		bgcolor	цвет фона
		border	граница
		bordercolor	цвет границы
		background	фоновый рисунок
		title	подсказка
		cellpadding	расстояние между границей и текстом в ячейке
		cellspacing	расстояние между ячейками
		frame= above	отображается верхняя граница таблицы
		frame= below	нижняя граница
		frame= hside	верхняя и нижняя
		frame= vside	правая и левая
		frame= lhs	левая
		frame= rhs	правая
		height	высота
		width	ширина
<b>2</b>	<CAPTION> </CAPTION>	заголовок таблицы	
		align=top (bottom, left, right)	выравнивание
<b>3</b>	<TR> </TR>	строка таблицы	
		align=left ( right, center, justify)	горизонтальное выравнивание
		valign=top (bottom, middle)	вертикальное выравнивание
		bgcolor	цвет фона строки

		bordercolor	цвет рамки строки
		height	высота строки
		width	ширина строки
4	<TD> </TD>	ячейка таблицы	
		align=left ( right, center, justify)	горизонтальное выравнивание
		valign=top (bottom, middle)	вертикальное выравнивание
		bgcolor	цвет фона ячейки
		bordercolor	цвет рамки ячейки
		colspan	количество столбцов для объединения
		rowspan	количество строк для объединения
5	<TH> </TH>	ячейки-заголовки (вместо <TD> </TD>)	

Порядок выполнения работы:

Задание №1 (table5.html)

<i>Первая строка</i>					
<i>Вторая строка</i>	<table> <tr> <td>внутренняя</td><td></td></tr> <tr> <td></td><td>таблица</td></tr> </table>	внутренняя			таблица
внутренняя					
	таблица				
<i>Третья строка</i>					

Задание №2 (table6.html)

### Личные данные

Фамилия	Имя	Возраст
Иванов	Иван	18
Петров	Петр	19
Средний возраст		18,5

Задание №3 (table7.html)

### Личные данные

Фамилия	Имя	Возраст
Иванов	Иван	18
Петров	Петр	19
Средний возраст		18,5
До 3-х месяцев работы, премия	Больше 3-х месяцев работы, премия	Премии будут на банковской карточке
100	300	

#### Задание №4 (table8.html)

1. Создать в программе MS Word таблицу «Расписание занятий на неделю», используя:
  - заливку ячеек
  - цветной текст
  - курсив, жирный шрифт
  - выравнивание текста в ячейках
  - цветные рамки.
2. Созданную таблицу сформировать на Web-странице. Результат (два файла) сохранить на своём пользователе в папке (название папки текущая дата)

#### Критерии оценки за практическую работу:

1. содержание Web-страницы соответствует заданию
2. текст на Web-странице не содержит ошибок
3. прописано название Web-страницы
4. правильный html-код
5. гиперссылки работают верно
6. правильное название файла
7. правильное расположение файла на Личном диске студента

<b>5</b>	Пункты 1-7 верны
<b>4</b>	Один или два пункта из 1-5 содержат неточности. Пункты 6,7 верны
<b>3</b>	Три и более пунктов из 1-5 содержат неточности. Пункты 6,7 верны
<b>2</b>	Пункты 6,7 не верны или Пункты 1-5 не верны

### Практическая работа №8 «Создание форм»

**Тема:** Язык HTML

**Количество часов:** 2

**Цели:** создания Web-сайтов на основе языков и редакторов HTML

**Задачи:** создавать Web-сайты с помощью языков и редакторов HTML, читать HTML-коды

#### Теоретическая часть:

Для того чтобы посетители сайта могли не только просматривать информацию, но и отправлять сведения его администраторам, на страницах размещают **формы**.

Формы включают в себя управляющие элементы различных типов:

- текстовые поля
- раскрывающиеся списки
- флажки
- переключатели и т. д.

	Тег	Атрибут, значение	Описание
1	<FORM></FORM>	задание формы	
2	<INPUT>	элемент управления	
		type=text	текстовое поле
		type=password	ввод пароля
		type=checkbox	флажки
		type=radio	переключатели
		type=submit	кнопка, отправляющая данные на Web-сервер
		type=reset	сброс текущих параметров

		type=file	кнопка вызова окна выбора файлов
		type=button	кнопка
		size	ширина текстового поля в пикселях
		value	значение или название
3	<SELECT> </SELECT>	выпадающий список значений	
	<OPTION>	отдельный элемент списка	
	SELECTED	выбранный элемент списка	
4	<TEXTAREA> </TEXTAREA>	многострочная текстовая область (с полосой прокрутки)	
	rows	высота области в строках	
	cols	ширина области в символах	

### Порядок выполнения работы:

Сохранить anketa.html

### Критерии оценки за практическую работу:

1. содержание Web-страницы соответствует заданию
2. текст на Web-странице не содержит ошибок
3. прописано название Web-страницы
4. правильный html-код
5. гиперссылки работают верно
6. правильное название файла
7. правильное расположение файла на Личном диске студента

5	Пункты 1-7 верны
4	Один или два пункта из 1-5 содержат неточности. Пункты 6,7 верны
3	Три и более пунктов из 1-5 содержат неточности. Пункты 6,7 верны
2	Пункты 6,7 не верны или Пункты 1-5 не верны

## Практическая работа №9

### «Разработка фреймов. Разработка плавающих фреймов»

Тема: Язык HTML

Количество часов: 4

Цели: создания Web-сайтов на основе языков и редакторов HTML

**Задачи:** создавать Web-сайты с помощью языков и редакторов HTML, читать HTML-коды

### **Теоретическая часть:**

Фреймы — это HTML-элементы, позволяющие разделить окно веб-браузера на несколько независимых окон, в каждое из которых можно загрузить отдельный HTML-документ.

Создание фрейма:

```
<FRAMESET>...</FRAMESET>
```

позволяет разделить экран на несколько вертикальных или несколько горизонтальных фреймов. Тэг <BODY> в этом случае не используется.

пример:

```
<HTML>
```

```
<HEAD>...</HEAD>
```

```
<FRAMESET>...</FRAMESET>
```

```
</HTML>
```

### *Атрибуты*

Данный тэг имеет два взаимоисключающих параметра: ROWS и COLS.

1. ROWS=список вертикальных подокон (сверху вниз)

Синтаксис используемых видов описания величин подокон:

value – простое числовое значение определяет фиксированную высоту подокна в пикселах.

value% – значение величины подокна в процентах от 1 до 100.

value\* – деление окна происходит в указанных пропорциях. Например, описание “3\*,\*,\*” говорит, что будет создано три фрейма с размерами 3/5 свободного пространства для первого фрейма и по 1/5 для двух других.

2. COLS=список горизонтальных подокон (справа налево)

То же самое, что и ROWS, но делит окно по горизонтали, а не по вертикали.

пример:

```
<FRAMESET COLS="50*,50">
```

описывает три фрейма, два по 50 точек справа и слева, и один внутри этих полосок.

```
<FRAMESET ROWS="20%,3*,*">
```

описывает три фрейма, первый из которых занимает 20% площади сверху экрана, второй 3/4 оставшегося от первого фрейма места (то есть, 60% всей площади окна), а последний 1/4 (то есть, 20% всей площади окна).

```
<FRAMESET ROWS="*,60%,*">
```

аналогично предыдущему примеру.

Тэги <FRAMESET> могут быть вложенными, то есть, например:

```
<FRAMESET ROWS="50%,50%">
```

```
<FRAMESET COLS="*,*">
```

```
</FRAMESET>
```

```
</FRAMESET>
```

### **Описание фрейма**

Тэг <FRAME SRC="Name№"> позволяет описать каждый фрейм, и присвоить имя гипертекстовому документу.

### *Атрибуты*

1. NAME=имя фрейма.
2. MARGINWIDTH= горизонтальный отступ (от 1 до 6) между фреймом и его границей.
3. MARGINHEIGHT=вертикальный отступ (от 1 до 6) между фреймом и его границей.
4. SCROLLING=прокрутка фрейма.  
SCROLLING=YES позволяет создать полосы прокрутки  
SCROLLING=NO полосы прокрутки отсутствуют

SCROLLING=AUTO полосы прокрутки отображаются в зависимости количества информации

5. SRC=задать гипертекстовый документ для этого фрейма.
6. TARGET=Name открыть ссылку во фрейме с именем Name.
7. NORESIZE позволяет создавать фреймы без возможности изменения размеров. По умолчанию, размер фрейма можно изменить при помощи мыши так же просто, как и размер окна Windows.

пример:

```
<FRAMESET ROWS="*,*">
<NOFRAMES>
<H1>Ваша версия WEB-браузера не поддерживает фреймы!</H1>
</NOFRAMES>
<FRAMESET COLS="65%,35%">
<FRAME SRC="link1.html">
<FRAME SRC="link2.html">
</FRAMESET>
<FRAMESET COLS="*,40%,*">
<FRAME SRC="link3.html">
<FRAME SRC="link4.html">
<FRAME SRC="link5.html">
</FRAMESET>
</FRAMESET>
```

Имя фрейма может быть использовано для ссылки на него из других документов (фреймов), с помощью параметра тэга <a> target (target="имя\_фрейма").

<a href="content.html" target="window-1">Главная</a>

Открываемый документ открывается во все окно

<a href="project.html" target="\_top">Мой проект о рыбах</a>

Открываемый документ открывается в новом окне

<a href="project.html" target="\_blank">Мой проект о рыбах</a>

Плавающий фрейм

```
<iframe width=670 height=550 src=00.html scrolling=yes frameborder="0" name=start>
</iframe>
```

<a href=26.html target="start">попечительский совет</a>

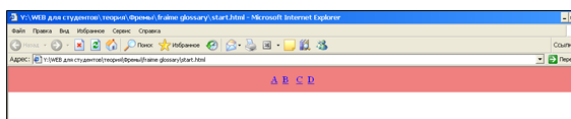
## Порядок выполнения работы:

### Задание 1

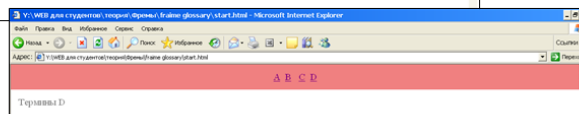
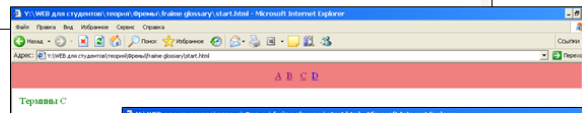
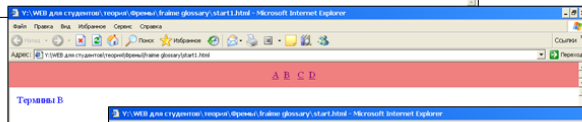
Создать Web-страницу, состоящую из двух фреймов (подокон), расположенных горизонтально.

- Первый фрейм содержит алфавитный ряд (каждая буква – гиперссылка).
- Второй фрейм содержит термины на соответствующую букву

Цвета текстов, фоны фреймов изменить, названия файлов сохранить



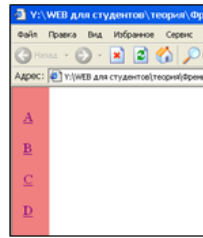
menu.html – файл с фреймами  
s.html – файл с ссылками  
a.html – файл с термином А  
b.html – файл с термином В  
c.html – файл с термином С  
d.html – файл с термином D



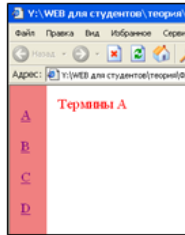
## Задание 2

Создать Web-страницу, состоящую из двух фреймов (подокон), расположенных вертикально.

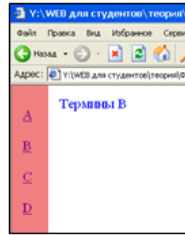
- Первый фрейм содержит алфавитный ряд (каждая буква – гиперссылка).
- Второй фрейм содержит термины на соответствующую букву
- Цвета текстов, фоны фреймов изменить, названия файлов сохранить



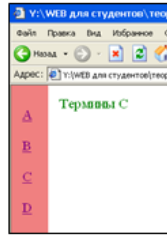
menu.html  
s.html



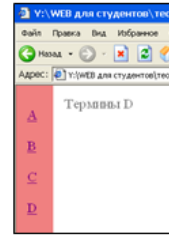
a.html



b.html



c.html

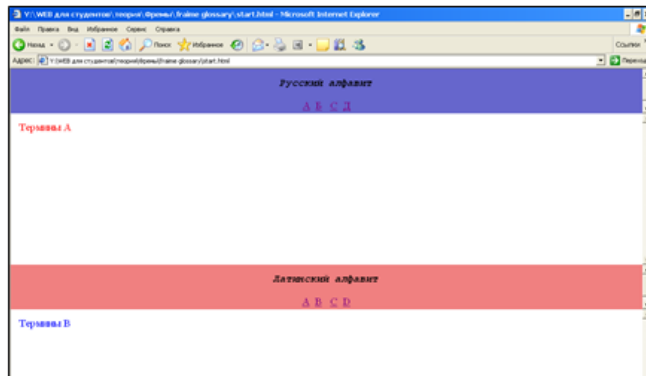
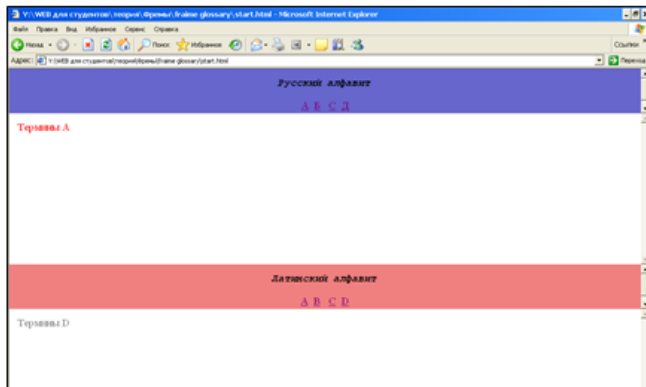


d.html

## Задание 3

Создать Web-страницу, состоящую из четырёх фреймов (подокон), расположенных вертикально.

- Первый (третий) фрейм содержит русский (латинский) алфавитный ряд (каждая буква – гиперссылка).
- Второй (четвёртый) фрейм содержит термины на соответствующую букву
- Цвета текстов, фоны фреймов изменить, названия файлов сохранить



**Критерии оценки за практическую работу:**

1. содержание Web-страницы соответствует заданию
2. текст на Web-странице не содержит ошибок
3. прописано название Web-страницы
4. правильный html-код
5. гиперссылки работают верно
6. правильное название файла
7. правильное расположение файла на Личном диске студента

<b>5</b>	Пункты 1-7 верны
<b>4</b>	Один или два пункта из 1-5 содержат неточности. Пункты 6,7 верны
<b>3</b>	Три и более пунктов из 1-5 содержат неточности. Пункты 6,7 верны
<b>2</b>	Пункты 6,7 не верны или Пункты 1-5 не верны

**Практическая работа №10****«Внедрение элементов управления. Разработка макета сайта. Группировка элементов управления»**

**Тема:** Выполнение индивидуального проекта

**Количество часов:** 10

**Цели:** создания Web-сайтов на основе языков и редакторов HTML

**Задачи:** создавать Web-сайты с помощью языков и редакторов HTML, читать HTML-коды

**Порядок выполнения работы:**

## создание макета Web-страницы

- Размещение элементов
- Графические изображения
- Шрифты
- Цветовые решения



**Задание 1: создать web-страницу, используя табличную верстку**

1000 п

800 п

<div>ЛОГОТ ИП</div> <div>ГБПОУ «Пермский политехнический колледж имени Н.Г. Славянова»</div> <div>Название страницы</div>													
<div>Основное меню</div> <table border="1"><tr><td>главная</td></tr><tr><td>гlossарий</td></tr><tr><td>тест</td></tr><tr><td>карта сайта</td></tr><tr><td>автор</td></tr></table> <div>Тематическое меню</div> <table border="1"><tr><td>Глава 1</td></tr><tr><td>Глава 2</td></tr><tr><td>Глава 3</td></tr><tr><td>Глава 4</td></tr><tr><td>Глава 5</td></tr><tr><td>Глава 6</td></tr><tr><td>Глава 7</td></tr></table> <div>250 п</div>	главная	гlossарий	тест	карта сайта	автор	Глава 1	Глава 2	Глава 3	Глава 4	Глава 5	Глава 6	Глава 7	<div>информация</div>
главная													
гlossарий													
тест													
карта сайта													
автор													
Глава 1													
Глава 2													
Глава 3													
Глава 4													
Глава 5													
Глава 6													
Глава 7													

Основное меню	
	главная
	гlossарий
	тест
	карта сайта
	автор

Тематическое меню	
	Глава 1
	Глава 2
	Глава 3
	Глава 4
	Глава 5
	Глава 6
	Глава 7

Фоновое изображение страницы  
Фон таблицы  
Логотип  
Цвет для текстов  
Гиперссылки  
Маркеры  
Шрифт Arial

Сохранить index.html в папке Шаблон

**Задание 2: добавить комментарии**

комментарии

<!--текст-->

```

<html>
<head>
<title>Название страницы</title>
</head>
<body background=82393198.jpg>
<!--ОСНОВНАЯ ТАБЛИЦА-->
<table width=1000px height=800px align=center bgcolor=darkred>
<tr height=100px>
<td colspan=2><p align=center><img src=logo.png height=100px width=100px align=left><font size=2><!--ГБПОУ "Пермский Политехнический колледж имени Н.Г. Славянова"/></font></p></td></tr>
<tr align=center valign=bottom><font size=9>Название страницы</font></p></td></tr>
<tr height=700px>
<td colspan=2><div style=display:flex; justify-content:space-between; align-items:center>
<div>
<!--ОСНОВНОЕ МЕНЮ-->
<table align=left>
<tr>
<td colspan=2>Основное меню</td>
</tr>
<tr>
<td><img src=success.png></td>
<td><a href=index.html>Главная</a></td>
</tr>
<tr>
<td><img src=success.png></td>
<td><a href=index.html>Глоссарий</a></td>
</tr>
<tr>
<td><img src=success.png></td>
<td><a href=index.html>Тест</a></td>
</tr>
<tr>
<td><img src=success.png></td>
<td><a href=index.html>Карта сайта</a></td>
</tr>
<tr>
<td colspan=2><img src=success.png></td>
</tr>
</table>
</div>
<div>
<!--ТЕМАТИЧЕСКОЕ МЕНЮ-->
<table align=right>
<tr>
<td colspan=2>Тематическое меню</td>
</tr>
<tr>
<td><img src=success.png></td>
<td><a href=index.html>Глава 1</a></td>
</tr>
<tr>
<td><img src=success.png></td>
<td><a href=index.html>Глава 2</a></td>
</tr>
<tr>
<td><img src=success.png></td>
<td><a href=index.html>Глава 3</a></td>
</tr>
<tr>
<td><img src=success.png></td>
<td><a href=index.html>Глава 4</a></td>
</tr>
<tr>
<td><img src=success.png></td>
<td><a href=index.html>Глава 5</a></td>
</tr>
<tr>
<td><img src=success.png></td>
<td><a href=index.html>Глава 6</a></td>
</tr>
<tr>
<td><img src=success.png></td>
<td><a href=index.html>Глава 7</a></td>
</tr>
</table>
</div>
</div>
</td></tr>
</table>
</body>
</html>

```

### Задание 3: преобразовать тематическое меню

Тематическое меню		
	Глава 1	
		пункт 1.1
		пункт 1.2
	Глава 2	
		пункт 2.1
		пункт 2.2
	Глава 3	
		пункт 3.1
		пункт 3.2
	Глава 4	
		пункт 4.1
		пункт 4.2
	Глава 5	
		пункт 5.1
		пункт 5.2
	Глава 6	
		пункт 6.1
		пункт 6.2
	Глава 7	
		пункт 7.1
		пункт 7.2

### Задание 4: создание макета web-сайта

Тематическое меню		
	Глава 1	
		пункт 1.1
		пункт 1.2
	Глава 2	
		пункт 2.1
		пункт 2.2
	Глава 3	
		пункт 3.1
		пункт 3.2
	Глава 4	
		пункт 4.1
		пункт 4.2

Создать копию файла index.html под названием index1.html

В файле index1.html преобразовать тематическое меню




Шаблон

Папка 1 содержит файлы:

1.html  
1-1.html  
1-2.html  
2.html  
2-1.html  
2-2.html  
3.html  
3-1.html  
3-2.html  
4.html  
4-1.html  
4-2.html

Организовать гиперсвязи


## index1.html

	ГБПОУ «Пермский политехнический колледж имени Н.Г. Славянова»																											
	<b>Название страницы</b>																											
<table border="1"> <tr><td colspan="2">Основное меню</td></tr> <tr><td><a href="#">главная</a></td><td></td></tr> <tr><td><a href="#">гlossарий</a></td><td></td></tr> <tr><td><a href="#">тест</a></td><td></td></tr> <tr><td><a href="#">карта сайта</a></td><td></td></tr> <tr><td><a href="#">автор</a></td><td></td></tr> </table>	Основное меню		<a href="#">главная</a>		<a href="#">гlossарий</a>		<a href="#">тест</a>		<a href="#">карта сайта</a>		<a href="#">автор</a>		<p>Главная страница</p>															
Основное меню																												
<a href="#">главная</a>																												
<a href="#">гlossарий</a>																												
<a href="#">тест</a>																												
<a href="#">карта сайта</a>																												
<a href="#">автор</a>																												
<table border="1"> <tr><td colspan="2">Тематическое меню</td></tr> <tr><td>Глава 1</td><td></td></tr> <tr><td>    пункт 1.1</td><td></td></tr> <tr><td>    пункт 1.2</td><td></td></tr> <tr><td>Глава 2</td><td></td></tr> <tr><td>    пункт 2.1</td><td></td></tr> <tr><td>    пункт 2.2</td><td></td></tr> <tr><td>Глава 3</td><td></td></tr> <tr><td>    пункт 3.1</td><td></td></tr> <tr><td>    пункт 3.2</td><td></td></tr> <tr><td>Глава 4</td><td></td></tr> <tr><td>    пункт 4.1</td><td></td></tr> <tr><td>    пункт 4.2</td><td></td></tr> </table>	Тематическое меню		Глава 1		пункт 1.1		пункт 1.2		Глава 2		пункт 2.1		пункт 2.2		Глава 3		пункт 3.1		пункт 3.2		Глава 4		пункт 4.1		пункт 4.2			
Тематическое меню																												
Глава 1																												
пункт 1.1																												
пункт 1.2																												
Глава 2																												
пункт 2.1																												
пункт 2.2																												
Глава 3																												
пункт 3.1																												
пункт 3.2																												
Глава 4																												
пункт 4.1																												
пункт 4.2																												


## 1.html

	ГБПОУ «Пермский политехнический колледж имени Н.Г. Славянова»																											
	<b>Название страницы</b>																											
<table border="1"> <tr><td colspan="2">Основное меню</td></tr> <tr><td><a href="#">главная</a></td><td></td></tr> <tr><td><a href="#">гlossарий</a></td><td></td></tr> <tr><td><a href="#">тест</a></td><td></td></tr> <tr><td><a href="#">карта сайта</a></td><td></td></tr> <tr><td><a href="#">автор</a></td><td></td></tr> </table>	Основное меню		<a href="#">главная</a>		<a href="#">гlossарий</a>		<a href="#">тест</a>		<a href="#">карта сайта</a>		<a href="#">автор</a>		<p>Глава 1</p>															
Основное меню																												
<a href="#">главная</a>																												
<a href="#">гlossарий</a>																												
<a href="#">тест</a>																												
<a href="#">карта сайта</a>																												
<a href="#">автор</a>																												
<table border="1"> <tr><td colspan="2">Тематическое меню</td></tr> <tr><td><a href="#">Глава 1</a></td><td></td></tr> <tr><td>    пункт 1.1</td><td></td></tr> <tr><td>    пункт 1.2</td><td></td></tr> <tr><td>Глава 2</td><td></td></tr> <tr><td>    пункт 2.1</td><td></td></tr> <tr><td>    пункт 2.2</td><td></td></tr> <tr><td>Глава 3</td><td></td></tr> <tr><td>    пункт 3.1</td><td></td></tr> <tr><td>    пункт 3.2</td><td></td></tr> <tr><td>Глава 4</td><td></td></tr> <tr><td>    пункт 4.1</td><td></td></tr> <tr><td>    пункт 4.2</td><td></td></tr> </table>	Тематическое меню		<a href="#">Глава 1</a>		пункт 1.1		пункт 1.2		Глава 2		пункт 2.1		пункт 2.2		Глава 3		пункт 3.1		пункт 3.2		Глава 4		пункт 4.1		пункт 4.2			
Тематическое меню																												
<a href="#">Глава 1</a>																												
пункт 1.1																												
пункт 1.2																												
Глава 2																												
пункт 2.1																												
пункт 2.2																												
Глава 3																												
пункт 3.1																												
пункт 3.2																												
Глава 4																												
пункт 4.1																												
пункт 4.2																												


## 1-1.html

	ГБПОУ «Пермский политехнический колледж имени Н.Г. Славянова»																																						
<b>Название страницы</b>																																							
<table><tr><td colspan="2">Основное меню</td></tr><tr><td><input type="checkbox"/></td><td>главная</td></tr><tr><td><input type="checkbox"/></td><td>гlossарий</td></tr><tr><td><input type="checkbox"/></td><td>тест</td></tr><tr><td><input type="checkbox"/></td><td>карта сайта</td></tr><tr><td><input type="checkbox"/></td><td>автор</td></tr></table> <table><tr><td colspan="2">Тематическое меню</td></tr><tr><td><input type="checkbox"/></td><td>Глава 1</td></tr><tr><td><input checked="" type="checkbox"/></td><td>пункт 1.1</td></tr><tr><td><input type="checkbox"/></td><td>пункт 1.2</td></tr><tr><td><input type="checkbox"/></td><td>Глава 2</td></tr><tr><td><input type="checkbox"/></td><td>пункт 2.1</td></tr><tr><td><input type="checkbox"/></td><td>пункт 2.2</td></tr><tr><td><input type="checkbox"/></td><td>Глава 3</td></tr><tr><td><input type="checkbox"/></td><td>пункт 3.1</td></tr><tr><td><input type="checkbox"/></td><td>пункт 3.2</td></tr><tr><td><input type="checkbox"/></td><td>Глава 4</td></tr><tr><td><input type="checkbox"/></td><td>пункт 4.1</td></tr><tr><td><input type="checkbox"/></td><td>пункт 4.2</td></tr></table>	Основное меню		<input type="checkbox"/>	главная	<input type="checkbox"/>	гlossарий	<input type="checkbox"/>	тест	<input type="checkbox"/>	карта сайта	<input type="checkbox"/>	автор	Тематическое меню		<input type="checkbox"/>	Глава 1	<input checked="" type="checkbox"/>	пункт 1.1	<input type="checkbox"/>	пункт 1.2	<input type="checkbox"/>	Глава 2	<input type="checkbox"/>	пункт 2.1	<input type="checkbox"/>	пункт 2.2	<input type="checkbox"/>	Глава 3	<input type="checkbox"/>	пункт 3.1	<input type="checkbox"/>	пункт 3.2	<input type="checkbox"/>	Глава 4	<input type="checkbox"/>	пункт 4.1	<input type="checkbox"/>	пункт 4.2	<b>Глава 1</b> <b>Пункт 1.1</b>
Основное меню																																							
<input type="checkbox"/>	главная																																						
<input type="checkbox"/>	гlossарий																																						
<input type="checkbox"/>	тест																																						
<input type="checkbox"/>	карта сайта																																						
<input type="checkbox"/>	автор																																						
Тематическое меню																																							
<input type="checkbox"/>	Глава 1																																						
<input checked="" type="checkbox"/>	пункт 1.1																																						
<input type="checkbox"/>	пункт 1.2																																						
<input type="checkbox"/>	Глава 2																																						
<input type="checkbox"/>	пункт 2.1																																						
<input type="checkbox"/>	пункт 2.2																																						
<input type="checkbox"/>	Глава 3																																						
<input type="checkbox"/>	пункт 3.1																																						
<input type="checkbox"/>	пункт 3.2																																						
<input type="checkbox"/>	Глава 4																																						
<input type="checkbox"/>	пункт 4.1																																						
<input type="checkbox"/>	пункт 4.2																																						


## 1-2.html

	ГБПОУ «Пермский политехнический колледж имени Н.Г. Славянова»																																						
<b>Название страницы</b>																																							
<table><tr><td colspan="2">Основное меню</td></tr><tr><td><input type="checkbox"/></td><td>главная</td></tr><tr><td><input type="checkbox"/></td><td>гlossарий</td></tr><tr><td><input type="checkbox"/></td><td>тест</td></tr><tr><td><input type="checkbox"/></td><td>карта сайта</td></tr><tr><td><input type="checkbox"/></td><td>автор</td></tr></table> <table><tr><td colspan="2">Тематическое меню</td></tr><tr><td><input type="checkbox"/></td><td>Глава 1</td></tr><tr><td><input type="checkbox"/></td><td>пункт 1.1</td></tr><tr><td><input checked="" type="checkbox"/></td><td>пункт 1.2</td></tr><tr><td><input type="checkbox"/></td><td>Глава 2</td></tr><tr><td><input type="checkbox"/></td><td>пункт 2.1</td></tr><tr><td><input type="checkbox"/></td><td>пункт 2.2</td></tr><tr><td><input type="checkbox"/></td><td>Глава 3</td></tr><tr><td><input type="checkbox"/></td><td>пункт 3.1</td></tr><tr><td><input type="checkbox"/></td><td>пункт 3.2</td></tr><tr><td><input type="checkbox"/></td><td>Глава 4</td></tr><tr><td><input type="checkbox"/></td><td>пункт 4.1</td></tr><tr><td><input type="checkbox"/></td><td>пункт 4.2</td></tr></table>	Основное меню		<input type="checkbox"/>	главная	<input type="checkbox"/>	гlossарий	<input type="checkbox"/>	тест	<input type="checkbox"/>	карта сайта	<input type="checkbox"/>	автор	Тематическое меню		<input type="checkbox"/>	Глава 1	<input type="checkbox"/>	пункт 1.1	<input checked="" type="checkbox"/>	пункт 1.2	<input type="checkbox"/>	Глава 2	<input type="checkbox"/>	пункт 2.1	<input type="checkbox"/>	пункт 2.2	<input type="checkbox"/>	Глава 3	<input type="checkbox"/>	пункт 3.1	<input type="checkbox"/>	пункт 3.2	<input type="checkbox"/>	Глава 4	<input type="checkbox"/>	пункт 4.1	<input type="checkbox"/>	пункт 4.2	<b>Глава 1</b> <b>Пункт 1.2</b>
Основное меню																																							
<input type="checkbox"/>	главная																																						
<input type="checkbox"/>	гlossарий																																						
<input type="checkbox"/>	тест																																						
<input type="checkbox"/>	карта сайта																																						
<input type="checkbox"/>	автор																																						
Тематическое меню																																							
<input type="checkbox"/>	Глава 1																																						
<input type="checkbox"/>	пункт 1.1																																						
<input checked="" type="checkbox"/>	пункт 1.2																																						
<input type="checkbox"/>	Глава 2																																						
<input type="checkbox"/>	пункт 2.1																																						
<input type="checkbox"/>	пункт 2.2																																						
<input type="checkbox"/>	Глава 3																																						
<input type="checkbox"/>	пункт 3.1																																						
<input type="checkbox"/>	пункт 3.2																																						
<input type="checkbox"/>	Глава 4																																						
<input type="checkbox"/>	пункт 4.1																																						
<input type="checkbox"/>	пункт 4.2																																						


## 2.html

	ГБПОУ «Пермский политехнический колледж имени Н.Г. Славянова»																																						
<b>Название страницы</b>																																							
<table><tr><td colspan="2">Основное меню</td></tr><tr><td><input type="checkbox"/></td><td>главная</td></tr><tr><td><input type="checkbox"/></td><td>гlossарий</td></tr><tr><td><input type="checkbox"/></td><td>тест</td></tr><tr><td><input type="checkbox"/></td><td>карта сайта</td></tr><tr><td><input type="checkbox"/></td><td>автор</td></tr></table> <table><tr><td colspan="2">Тематическое меню</td></tr><tr><td><input type="checkbox"/></td><td>Глава 1</td></tr><tr><td><input type="checkbox"/></td><td>пункт 1.1</td></tr><tr><td><input type="checkbox"/></td><td>пункт 1.2</td></tr><tr><td><input checked="" type="checkbox"/></td><td>Глава 2</td></tr><tr><td><input type="checkbox"/></td><td>пункт 2.1</td></tr><tr><td><input type="checkbox"/></td><td>пункт 2.2</td></tr><tr><td><input type="checkbox"/></td><td>Глава 3</td></tr><tr><td><input type="checkbox"/></td><td>пункт 3.1</td></tr><tr><td><input type="checkbox"/></td><td>пункт 3.2</td></tr><tr><td><input type="checkbox"/></td><td>Глава 4</td></tr><tr><td><input type="checkbox"/></td><td>пункт 4.1</td></tr><tr><td><input type="checkbox"/></td><td>пункт 4.2</td></tr></table>	Основное меню		<input type="checkbox"/>	главная	<input type="checkbox"/>	гlossарий	<input type="checkbox"/>	тест	<input type="checkbox"/>	карта сайта	<input type="checkbox"/>	автор	Тематическое меню		<input type="checkbox"/>	Глава 1	<input type="checkbox"/>	пункт 1.1	<input type="checkbox"/>	пункт 1.2	<input checked="" type="checkbox"/>	Глава 2	<input type="checkbox"/>	пункт 2.1	<input type="checkbox"/>	пункт 2.2	<input type="checkbox"/>	Глава 3	<input type="checkbox"/>	пункт 3.1	<input type="checkbox"/>	пункт 3.2	<input type="checkbox"/>	Глава 4	<input type="checkbox"/>	пункт 4.1	<input type="checkbox"/>	пункт 4.2	<b>Глава 2</b>
Основное меню																																							
<input type="checkbox"/>	главная																																						
<input type="checkbox"/>	гlossарий																																						
<input type="checkbox"/>	тест																																						
<input type="checkbox"/>	карта сайта																																						
<input type="checkbox"/>	автор																																						
Тематическое меню																																							
<input type="checkbox"/>	Глава 1																																						
<input type="checkbox"/>	пункт 1.1																																						
<input type="checkbox"/>	пункт 1.2																																						
<input checked="" type="checkbox"/>	Глава 2																																						
<input type="checkbox"/>	пункт 2.1																																						
<input type="checkbox"/>	пункт 2.2																																						
<input type="checkbox"/>	Глава 3																																						
<input type="checkbox"/>	пункт 3.1																																						
<input type="checkbox"/>	пункт 3.2																																						
<input type="checkbox"/>	Глава 4																																						
<input type="checkbox"/>	пункт 4.1																																						
<input type="checkbox"/>	пункт 4.2																																						

2-1.html

		ГБПОУ «Пермский политехнический колледж имени Н.Г. Славянова»																									
		<b>Название страницы</b>																									
<table border="1"><tr><td colspan="2">Основное меню</td></tr><tr><td>главная</td><td></td></tr><tr><td>гlossарий</td><td></td></tr><tr><td>тест</td><td></td></tr><tr><td>карта сайта</td><td></td></tr><tr><td>автор</td><td></td></tr></table>		Основное меню		главная		гlossарий		тест		карта сайта		автор		<div>Глава 2 Пункт 2.1</div>													
Основное меню																											
главная																											
гlossарий																											
тест																											
карта сайта																											
автор																											
<table border="1"><tr><td colspan="2">Тематическое меню</td></tr><tr><td>Глава 1</td><td></td></tr><tr><td>пункт 1.1</td><td></td></tr><tr><td>пункт 1.2</td><td></td></tr><tr><td>Глава 2</td><td></td></tr><tr><td>пункт 2.1</td><td></td></tr><tr><td>пункт 2.2</td><td></td></tr><tr><td>Глава 3</td><td></td></tr><tr><td>пункт 3.1</td><td></td></tr><tr><td>пункт 3.2</td><td></td></tr><tr><td>Глава 4</td><td></td></tr><tr><td>пункт 4.1</td><td></td></tr><tr><td>пункт 4.2</td><td></td></tr></table>		Тематическое меню		Глава 1		пункт 1.1		пункт 1.2		Глава 2		пункт 2.1		пункт 2.2		Глава 3		пункт 3.1		пункт 3.2		Глава 4		пункт 4.1		пункт 4.2	
Тематическое меню																											
Глава 1																											
пункт 1.1																											
пункт 1.2																											
Глава 2																											
пункт 2.1																											
пункт 2.2																											
Глава 3																											
пункт 3.1																											
пункт 3.2																											
Глава 4																											
пункт 4.1																											
пункт 4.2																											

2-2.html

		ГБПОУ «Пермский политехнический колледж имени Н.Г. Славянова»																									
		<b>Название страницы</b>																									
<table border="1"><tr><td colspan="2">Основное меню</td></tr><tr><td>главная</td><td></td></tr><tr><td>гlossарий</td><td></td></tr><tr><td>тест</td><td></td></tr><tr><td>карта сайта</td><td></td></tr><tr><td>автор</td><td></td></tr></table>		Основное меню		главная		гlossарий		тест		карта сайта		автор		<div>Глава 2 Пункт 2.2</div>													
Основное меню																											
главная																											
гlossарий																											
тест																											
карта сайта																											
автор																											
<table border="1"><tr><td colspan="2">Тематическое меню</td></tr><tr><td>Глава 1</td><td></td></tr><tr><td>пункт 1.1</td><td></td></tr><tr><td>пункт 1.2</td><td></td></tr><tr><td>Глава 2</td><td></td></tr><tr><td>пункт 2.1</td><td></td></tr><tr><td>пункт 2.2</td><td></td></tr><tr><td>Глава 3</td><td></td></tr><tr><td>пункт 3.1</td><td></td></tr><tr><td>пункт 3.2</td><td></td></tr><tr><td>Глава 4</td><td></td></tr><tr><td>пункт 4.1</td><td></td></tr><tr><td>пункт 4.2</td><td></td></tr></table>		Тематическое меню		Глава 1		пункт 1.1		пункт 1.2		Глава 2		пункт 2.1		пункт 2.2		Глава 3		пункт 3.1		пункт 3.2		Глава 4		пункт 4.1		пункт 4.2	
Тематическое меню																											
Глава 1																											
пункт 1.1																											
пункт 1.2																											
Глава 2																											
пункт 2.1																											
пункт 2.2																											
Глава 3																											
пункт 3.1																											
пункт 3.2																											
Глава 4																											
пункт 4.1																											
пункт 4.2																											

И так далее.....

Задание 5: создание плавающих фреймов

1.html



Задание 5: создание плавающих фреймов

1-1.html



Задание 5: создание плавающих фреймов

1-2.html



**Задание 6: создание форм в файле 2-1.html**

логотип

ГБПОУ «Пермский политехнический колледж имени Н.Г. Славянова»

Название страницы

Глава 2

Пункт 2.1

Основное меню

главная

гlossарий

тест

карта сайта

зетор

Тематическое меню

Глава 1

пункт 1.1

пункт 1.2

Глава 2

пункт 2.1

пункт 2.2

Глава 3

пункт 3.1

пункт 3.2

Глава 4

пункт 4.1

пункт 4.2

Фамилия:

Имя:

Отчество:

Группа:

Пол:

ж ☐

м ☐

*ПРИМЕЧАНИЕ: Группа – раскрывающийся список (6 любых групп)*

**Критерии оценки за практическую работу:**

1. содержание Web-страницы соответствует заданию
2. текст на Web-странице не содержит ошибок
3. прописано название Web-страницы
4. правильный html-код
5. гиперссылки работают верно
6. разработан дизайн страниц
7. правильное название файлов
8. правильное расположение файлов на Личном диске студента

<b>5</b>	Пункты 1-8 верны
<b>4</b>	Один или два пункта из 1-6 содержат неточности. Пункты 7,8 верны
<b>3</b>	Три и более пунктов из 1-6 содержат неточности. Пункты 7,8 верны
<b>2</b>	Пункты 7,8 не верны или Пункты 1-6 не верны

**Практическая работа №11  
«Применение CSS»**

**Тема:** Основы CSS

**Количество часов:** 6

**Цели:** применение каскадных таблиц стилей

**Задачи:** Отформатируйте страницу с помощью листа стилей, размещённого в теге `<style>...</style>`

**Порядок выполнения работы:**

Отформатируйте страницу, используя следующие стили:

#### Форматирование тела документа

Тело документа <body>: размер шрифта - 12 pt, цвет: #000099, выравнивание: по ширине, тип шрифта - Times New Roman, цвет фона: #ffffee, отступы слева, справа, сверху, снизу - по 5 пикселей

#### Форматирование заголовков

##### Заголовок 1 уровня

Для форматирования заголовка 1 уровня задайте следующие стили:

- размер шрифта - 22 pt,
- цвет: #990099,
- выравнивание: по левому краю,
- тип шрифта - Arial

##### Заголовок 2 уровня

Для форматирования заголовка 2 уровня задайте следующие стили:

- размер шрифта - 20 pt,
- цвет: #009900,
- выравнивание: по центру,
- тип шрифта - Times New Roman

##### Заголовок 3 уровня

Для форматирования заголовка 3 уровня задайте следующие стили:

- размер шрифта - 18 pt,
- цвет: #ff6600,
- выравнивание: по правому краю,
- тип шрифта - Courier

##### Заголовок 4 уровня

Для форматирования заголовка 4 уровня задайте следующие стили:

- размер шрифта - 16 pt,
- цвет: #0066ff,
- выравнивание: по ширине,
- тип шрифта - Tahoma

##### Заголовок 5 уровня

Для форматирования заголовка 5 уровня задайте следующие стили:

- размер шрифта - 20 pt,
- цвет: #6600ff,
- выравнивание: по центру,
- тип шрифта - Times New Roman,
- цвет фона: #ffffaa

##### Заголовок 6 уровня

Для форматирования заголовка 6 уровня задайте следующие стили:

- размер шрифта - 14 pt,
- цвет: #6600ff,
- выравнивание: по центру,
- тип шрифта - Arial,
- украшение текста: Зачеркнутый (line-through)

#### Форматирование параграфов

Параграф <p>: размер шрифта - 10 pt, цвет: #000000, выравнивание: по ширине, тип шрифта - Arial, цвет фона: #ffffee, отступы слева, справа - по 80 пикселей, отступы между абзацами - сверху 2 пикселя, снизу - 0 пикселей.

Sed ut perspiciatis, unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam eaque ipsa, quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt, explicabo. nemo enim ipsam voluptatem, quia voluptas sit, aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos, qui ratione voluptatem sequi nesciunt, neque porro quisquam est, qui dolorem ipsum, quia dolor sit, amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt, ut labore et dolore magnam aliquam quaerat voluptatem. ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? quis autem vel eum iure reprehenderit, qui in ea voluptate velit esse, quam nihil molestiae consequatur, vel illum, qui dolorem eum fugiat, quo voluptas nulla pariatur?

Sed ut perspiciatis, unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam eaque ipsa, quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt, explicabo. nemo enim ipsam voluptatem, quia voluptas sit, aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos, qui ratione voluptatem sequi nesciunt, neque porro quisquam est, qui dolorem ipsum, quia dolor sit, amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt, ut labore et dolore magnam aliquam quaerat voluptatem. ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? quis autem vel eum iure reprehenderit, qui in ea voluptate velit esse, quam nihil molestiae consequatur, vel illum, qui dolorem eum fugiat, quo voluptas nulla pariatur?

Параграфы с классами <p class="over">. Используются все те же стили, что и для предыдущего пункта, кроме того украшение текста - с приподнятой линией (overline)

Sed ut perspiciatis, unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam eaque ipsa, quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt, explicabo. nemo enim ipsam voluptatem, quia voluptas sit, aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos, qui ratione voluptatem sequi nesciunt, neque porro quisquam est, qui dolorem ipsum, quia dolor sit, amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt, ut labore et dolore magnam aliquam quaerat voluptatem. ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? quis autem vel eum iure reprehenderit, qui in ea voluptate velit esse, quam nihil molestiae consequatur, vel illum, qui dolorem eum fugiat, quo voluptas nulla pariatur?

Sed ut perspiciatis, unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam eaque ipsa, quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt, explicabo. nemo enim ipsam voluptatem, quia voluptas sit, aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos, qui ratione voluptatem sequi nesciunt, neque porro quisquam est, qui dolorem ipsum, quia dolor sit, amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt, ut labore et dolore magnam aliquam quaerat voluptatem. ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? quis autem vel eum iure reprehenderit, qui in ea voluptate velit esse, quam nihil molestiae consequatur, vel illum, qui dolorem eum fugiat, quo voluptas nulla pariatur?



Далее отформатируйте все элементы самостоятельно

#### Форматирование таблиц

Заголовок таблицы 1	Заголовок таблицы 2	Заголовок таблицы 2	Заголовок таблицы 2
Sed ut perspiciatis, unde omnis iste natus error sit voluptatem	Sed ut perspiciatis, unde omnis iste natus error sit voluptatem	Sed ut perspiciatis, unde omnis iste natus error sit voluptatem	Sed ut perspiciatis, unde omnis iste natus error sit voluptatem
Sed ut perspiciatis, unde omnis iste natus error sit voluptatem	Sed ut perspiciatis, unde omnis iste natus error sit voluptatem	Sed ut perspiciatis, unde omnis iste natus error sit voluptatem	Sed ut perspiciatis, unde omnis iste natus error sit voluptatem

#### Цены на процессоры

Процессор	Производитель	Стоимость
Pentium IV	Intel®	200 €
Pentium III	Intel®	100 €
Celeron 2400	Intel®	150 €
Celeron 1800	Intel®	70 €
Athlon 2400	AMD®	150 €
Duron	AMD®	50 €

#### Форматирование списков

##### Списки определенных

- Хлеб** - пицца, приготовленная из отборного зерна  
**Вода** - жидкость без цвета и запаха  
**Пиво** - напиток из хлеба и воды

##### Списки нумерованные

- Типы операционных систем
1. MS DOS
  2. Windows
  3. Linux
  4. UNIX
  5. MacOS
  6. Free BSD

#### Форматирование ссылок

Ссылка без подчеркивания: <http://www.modern-computer.ru>

Цвета ссылок: <http://www.modern-computer.ru>

#### Критерии оценки за практическую работу:

1. содержание Web-страницы соответствует заданию
2. текст на Web-странице не содержит ошибок
3. прописано название Web-страницы
4. правильный html-код
5. правильное название файла
6. правильное расположение файла на Личном диске студента

<b>5</b>	Пункты 1-6 верны
<b>4</b>	Один или два пункта из 1-4 содержат неточности. Пункты 5,6 верны
<b>3</b>	Три и более пунктов из 1-4 содержат неточности. Пункты 5,6 верны
<b>2</b>	Пункты 5,6 не верны или Пункты 1-4 не верны

### Практическая работа №12 «Простые скрипты JavaScript»

**Тема:** Основы JavaScript

**Количество часов:** 2

**Цели:** применение каскадных таблиц стилей

**Задачи:** применить простые скрипты на JS

**Порядок выполнения работы:**

## 1. Добавление сценариев Javascript в HTML



**Javascript** — скриптовый (сценарный) интерпретируемый язык программирования, интерпретатор встроен непосредственно в браузер. Поэтому обработка скрипта (или другими словами — *сценария*) происходит на стороне клиента без обращения к серверной стороне

Скрипты используются в html-страницах для увеличения функциональности и возможностей взаимодействия с посетителями сайта.

Для добавления сценария на страницу HTML используется дескриптор:

```
<script type="text/javascript">
</script>
```

Или сегодня можно использовать упрощенный вариант:

```
<script>
</script>
```

**Атрибуты дескриптора:**

- **type** — атрибут, пришедший на замену **language**; он сообщает браузеру, какой язык используется внутри дескрипторов;
- **src** — атрибут определяет URL внешнего исходного JavaScript-файла, сценарий которого прикрепляется к html-странице.

Таким образом, варианты тега **script**:

### 1. Устаревший вариант:

```
<script language="javascript">...</script>
```

### 2. Действующие варианты:

```
<script type="text/javascript">...</script>
```

```
<script>...</script>
```

### 3. Вариант с прикрепленным файлом скрипта:

```
<script src="/js/scripts/myscript.js">
</script>
```

Итак, кратко резюмируем то, что необходимо знать о javascript:

- тег **script** обычно помещается в html-страницу в область **head** или **body**;
- этот тег указывает на то, что внутри находится сценарий — исполняемый код, в нашем случае скрипт на языке javascript;
- когда html-парсер браузера, отображая последовательно структуру html, доходит до тега **script**, то он передает инициативу интерпретатору **javascript**;
- интерпретатор, в свою очередь, исполняет содержимое кода до закрывающего тега **script**, а затем опять передает управление html-парсеру.

## ДОБАВЛЕНИЕ JAVASCRIPT В HTML

Встраивание javascript в html происходит двумя основными способами, которые рассмотрим на примере:



**Пример 1:** Вывести в браузере сообщение «Это JavaScript!», используя скрипт на языке JavaScript.

```
Это JavaScript!
Это обычный HTML документ.
```

🔧 Решение:

## 1. Встраивание JavaScript непосредственно в HTML-страницу:

- Создайте html-страницу со следующим кодом:

```
<html><head></head>
<body>
<!-- Сценарий -->
<script type="text/javascript">
    document.write("Это JavaScript!");
</script>
<!-- Конец сценария -->
<hr>
Это обычный HTML документ.
</body></html>
```

**Пояснение:** В данном примере дескриптор `script` может находиться как в теле документа, т.е. тега `body` (как в примере), так и в области `head`.

- Откройте страницу в браузере и посмотрите на результат. Всё, что находится до горизонтальной линии — это результат работы скрипта; после чего идет текст в разметке html.

## 2. Размещение сценария во внешнем файле:

- Создайте две страницы: `lab1.html` и `myscript.js`. Расположите обе страницы в одном каталоге.
- В html-документе разместите код:

```
<html>
<head>
<!-- Прикрепление файла с кодом сценария -->
<script src="myscript.js"></script>
</head>
<body>
...
</body></html>
```

**Пояснение:** При прикреплении внешнего файла со скриптом тег `script` следует размещать в области `head`.

- В файле с расширением `js` (в данном конкретном примере — `myscript.js`) находится единственная строка — код для вывода сообщения:

```
document.write("Это JavaScript!");
```

- Откройте страницу `lab1.html` в браузере и посмотрите результат.

**Пояснение:** При прикреплении js-файла следует иметь в виду, что в html-файле надо указывать относительный путь к файлу со скриптом. Так, если файл со скриптом находится в каталоге `jscripts`, то код будет: `script src="jscripts/myscript.js"`.



- Функция `document.write()` используется для вывода информации на экран.
- Тогда как `document.writeln()` — используется для перевода на новую строку, если используется тег форматирования `pre`.
- `alert()` — это метод для вывода модального (диалогового) окна с сообщением

Выполните следующий пример, чтобы увидеть особенности работы метода `alert()`:




**Пример 2:** Поменяйте местами `alert` и `document.write`. Посмотрите на результат в браузере.

```
<script type="text/javascript">
    alert ("Hello?");
    document.write("Hello!");
</script>
```





**Важно:** особенность модального окна `alert()` состоит в том, что пользователь не может продолжить работу, пока не щелкнет по кнопке окна


Обратите внимание, как работает javascript, помещенный в теги оформления шрифта:

 **Пример 3:** Перенесите скрипт в **BODY** после тега **H1**. Посмотрите на результат.

```
<h1>
<script type="text/javascript">
  alert ("Hello?");
  document.write("Hello!");
</script>
</h1>
```

 **Важно:** Пример показывает, что при использовании метода `write()` на странице выводится не просто текст, а html-код. То есть данный код может содержать теги html, которые будут преобразованы в соответствующее форматирование текста.

 **Задание Js 1.** Вывести в окно браузера следующие данные: Ваше ФИО, возраст, хобби (каждое на новой строке)


 **Задание Js 2.** Написать сценарий (javascript) для вывода двух строк текста, красной и синей.

Красная строка


Синяя строка

Замечание:  
Теги html для оформления текста цветом:

```
<font color="red">Красная строка</font>
```

 **Задание Js 3.** Найдите и исправьте ошибки во фрагментах кода:

- ```
document.whrit("Проблемы?");
```
- ```
alert(Hello);
```

 **Вопросы для самоконтроля:**

- Что такое сценарий (скрипт)?
- Сформулируйте основные задачи, решаемые с помощью скриптов javascript.
- Опишите основные правила подключения сценариев к странице html.
- Какой метод javascript используется для вывода текста (html-кода) на страницу?

### Критерии оценки за практическую работу:

1. содержание Web-страницы соответствует заданию
2. текст на Web-странице не содержит ошибок
3. прописано название Web-страницы
4. правильный html-код
5. правильное название файла
6. правильное расположение файла на Личном диске студента

<b>5</b>	Пункты 1-6 верны
<b>4</b>	Один или два пункта из 1-4 содержат неточности. Пункты 5,6 верны
<b>3</b>	Три и более пунктов из 1-4 содержат неточности. Пункты 5,6 верны

2	Пункты 5,6 не верны или Пункты 1-4 не верны
---	---

## Практическая работа №13 «Использование функций»

**Тема:** Основы JavaScript

**Количество часов:** 2

**Цели:** применение JavaScript

**Задачи:** применить функции JS

**Порядок выполнения работы:**

### 2. Синтаксис JavaScript, основные понятия

JavaScript – язык интерпретируемый (простыми словами: интерпретируемый — исполняется последовательно команда за командой), но также оснащен JIT-компиляцией.

**JIT** или компиляция «на лету» (Just-in-time compilation) — технология увеличения скорости работы программных систем, использующих байт-код, путём компиляции байт-кода в машинный код непосредственно во время работы программы.

**Рассмотрим некоторые понятия, относящиеся к синтаксису языка:**

**Сценарий** — текст, состоящий из:

- операторов,
- блоков, т. е. взаимосвязанных наборов операторов, и
- комментариев.

**Операторы** могут содержать:

- *переменные* — могут изменять свое значение в программе,
- *константы* — не изменяют свое значение,
- выражения.

**Пояснение:** Вспомним, что *переменная* — это область памяти для хранения значений; для обращения к переменной используется ее имя (идентификатор). Кроме того, у переменной есть *тип данных* — это тип значения, которое принимает переменная.

**Идентификаторы (identifiers)** — имена переменных, методов и объектов:

- состоят из комбинации букв и цифр;
- должны начинаться либо с буквы, либо с символа подчеркивания;
- не должны содержать пробелов.



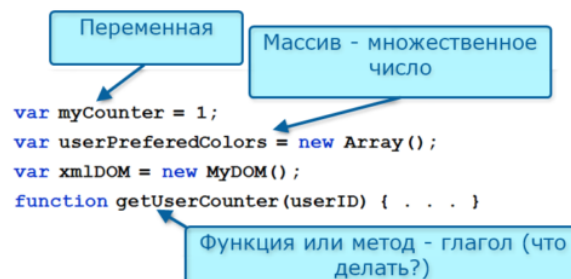
**Важно:** Язык JavaScript чувствителен к регистру:

```
//переменные различаются:  
counter=1  
Counter=1
```

### «Верблюжья нотация» в записи идентификаторов:

Есть определенные устоявшиеся среди программистов правила для идентификаторов (имён) переменных, функций, массивов и классов. Рассмотрим их:

- `num_docs` — знак подчеркивания между словами — хорошо, но есть способ лучше
- `numDocs` — вот такой «верблюжий» стиль превосходно подходит для именования переменных:
  - все имена строчными буквами,
  - на стыке слов — большая буква,
  - *переменные и свойства* — называем именами существительными,
  - *массивы и коллекции* — называем существительными во множительном числе,
  - *функции и методы* — называем глаголами,
  - *название классов* — с заглавной буквы.

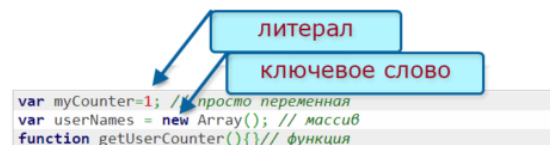


Пример:

```
var myCounter=1; // просто переменная  
var userNames = new Array(); // массив  
function getUserCounter(){} // функция
```

**Ключевые слова (keywords)** — предварительно определенные идентификаторы, составляющие основу языка программирования. Ключевые слова нельзя использовать для имен переменных, функций, объектов и методов.

**Литералы (literals)** — это постоянные значения JavaScript. Это значения, которые не изменяются во время выполнения сценария (целочисленные литералы, литералы с плавающей точкой, логические литералы (`true` и `false`), строковый литерал — это ноль и более символов, заключенных в двойные («») или одиночные («») кавычки).



## ПРАВИЛА ОФОРМЛЕНИЯ СКРИПТА JAVASCRIPT

- каждый оператор JavaScript лучше начинать с новой строки;
- каждый оператор заканчивается точкой с запятой;

**Пояснение:** сегодня точка с запятой в конце оператора не обязательна, но если написать в строку несколько операторов (это тоже разрешается), то необходимо их разделить через `;`

Такой код **не работает**:

```
a=5 document.write(a)
```

Код работает верно:

```
// способ 1:  
a=5  
document.write(a)  
  
// способ 2:  
a=5; document.write(a);  
  
// способ 3:  
a=5;  
document.write(a);
```

- блок — это набор операторов (составной оператор), заключенный в фигурные скобки `{ }`.

```
{  
  document.write(a);  
  alert(b);  
}
```

## JavaScript комментарии



В JavaScript допустимы два вида операторов комментария:

1. `//` — одна строка символов, расположенная справа от этого оператора, считается комментарием;
2. `/*...*/` — все, что заключено между `/*` и `*/`, считается комментарием; с помощью этого оператора можно выделить несколько строк в качестве комментария.

```
// проверка  
  
/* здесь может быть ошибка  
a=5;  
document.write(a);  
*/
```

**Пояснение:** Второй способ комментирования обычно используется при поиске ошибок: тот блок сценария, в котором может находиться потенциальная ошибка, комментируется.



**Задание Js 4.** Исправьте ошибки во фрагменте кода:

```
alert("Hello World!"); // это однострочный комментарий
```

## ОБЪЯВЛЕНИЕ ПЕРЕМЕННЫХ В JAVASCRIPT И ОПЕРАТОР ПРИСВАИВАНИЯ

**Переменная (variable)** — это имя, присваиваемое ячейке памяти компьютера, которая хранит определенные данные.

**JavaScript — нетипизированный язык.** Это значит, что переменные принимают тот тип данных, значение которого в них присваивается. Напрямую задавать тип переменной не надо.



Объявление переменной происходит при помощи служебного слова javascript `var` :

```
var + имя переменной + ;
```

Пример объявления переменной:

```
var i;
```

Оператор присваивания:

```
i=0;
```



Объявление переменной можно объединить с присваиванием:

```
var + имя переменной + = + значение + ;
```

Переменная всегда слева, справа – литерал (значение переменной).

Пример:

```
var i=0;
```

Таким образом, резюмируем. Существует три варианта объявления с присваиванием:

1.

```
var hello = "привет";
```

или

2.

```
var hello;  
hello="привет";
```

или

3.

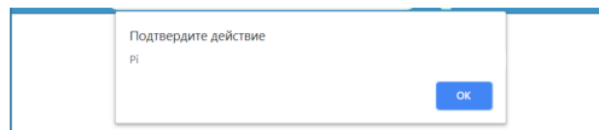
```
hello="привет";
```



Объявление переменной при помощи служебного слова `var` можно опускать



**Пример 4:** Создать переменную с идентификатором `myVal` со строковым значением «Pi». Вывести значение переменной, используя модальное окно (метод `alert()`).



👉 Решение:

- Создайте html-страницу с тегом `script`, предназначенным для размещения дальнейшего кода:

```
<html>  
<head></head>  
<body>  
<script type="text/javascript">  
  // будущий код javascript  
</script>  
</body>  
</html>
```

- Добавьте код для объявления, инициализации переменной и вывода ее значения в предназначенное для этого место:

```
var myVal; // объявляем переменную myVal  
myVal = "Pi"; /* присваиваем myVal некоторое значение */  
alert (myVal); // выводим значение
```

- Возможен также пример с необъявленной переменной:

```
myVal = "Pi"; /* присваиваем myVal некоторое значение */  
alert (myVal); // выводим значение
```

- Запустите страницу в браузере и посмотрите на результат.





**Задание Js 5.** Объявите две переменные разными способами и присвойте им значения. Выведите на экран значения переменных при помощи метода `alert()`



**Задание Js 6.** Исправьте ошибки в правильности объявления локальных переменных во фрагменте кода:

```
String s = "String";
int a = 10;
long b = 25;
```

## Константы в JavaScript



**Объявление константы:**

`const` + имя константы + `=` + значение + `;`

```
const MyX=2;
```

Данные, присвоенные константе, в течение программы не меняются!

## JAVASCRIPT ТИПЫ ДАННЫХ

Определение типа данных необходимо для установления операций, которые можно выполнить над переменными этого типа данных.



**Важно:** Но необходимо помнить, что в javascript типы переменных явно не указываются, т.к. это динамически типизированный, а не строго типизированный язык программирования.

Интерпретатор определяет тип переменной по правой части (по присвоенному ей значению). Объявление локальных переменных осуществляется при помощи ключевого слова `var`.

Тип данных	пример / объяснение
javascript undefined type	<pre>var x; alert (x);</pre> <p>значение, которое используется для переменных или свойств объекта, значения для которых не существует или оно не присвоено</p>
Null type	<pre>var x=1; x=null //пустое значение</pre> <p>значение, указывающее на отсутствие объекта</p>
Boolean type логический (true или false)	<pre>var x=false;</pre>
String type строковый	<pre>var x="Привет";</pre>
Number type числовой	<pre>var x=3.14; var y=-567;</pre>
RegExp регулярные выражения	
Object type	Программный объект, определяемый своими свойствами

## Логический тип (boolean)

Пример использования логического типа:

```
var a = true;
var b = false;
c = a && b; // результат false
c = a || b; // результат true
c = !a; // результат false
```

Javascript логические операторы:

Оператор	Название	Пример
!	Отрицание (логическое НЕ)	!X
&&	логическое И	X && Y
	логическое ИЛИ	X    Y



### Задание Js 7.

Выполните задание по шагам:

- Создайте 3 переменные с использованием ключевого слова `var` с идентификаторами: `a`, `b`, `c`.
- Переменной `a` присвойте значение `false`.
- Переменной `b` присвойте значение `null`.
- Переменная `c` должна принимать значение `undefined`.
- Отобразите значение 3-х переменных последовательно в модальных окнах (то есть с помощью метода `alert()`).

## Строковый тип (string)



**Строка** — набор символов, обрамляется либо в `""`, либо в `''`.

Три способа создания строкового объекта:

1. `имя_переменной = new String("строковое_значение");`

```
myString = new String ("Hello!");
```

2. `имя_переменной = "строковое_значение";`

```
myString = "Hello!";
```

3. `var имя_переменной = "строковое_значение";`

```
var myString = "Hello!";
```

### Операции над строками

- Конкатенация объединение строк:

Пример:

```
var x="При";
var y="вет";
var s=x+y; //"Привет"
```

- Специальные символы:

`\n` — новая строка

`\t` — табуляция



**Пример 5:** Реализуйте приведенный ниже код, чтобы посмотреть, как работают специальные символы в javascript. Запустите страницу в браузере:

```
alert("мама мыла раму");
alert("мама\n мыла\n раму");
```



**Задание Js 8.** Что должно быть в ответе на следующие присваивания?

"1"+2+3= ?  
1+2+"3"= ?



**Задание Js 9.** Вывести в окно браузера при помощи метода `alert()` следующие данные: Ваше ФИО, возраст, хобби (каждое на новой строке: использовать специальные символы)



**Задание Js 10.** С помощью javascript метода `document.write()` вывести в окно браузера строку: *Кто ты такой?* (с пробелами между словами).

Последовательно выполните:

1. Создать 4 переменные с использованием ключевого слова `var` с именами `str1`, `str2`, `str3`, `concatenation`.
2. Переменной `str1` присвоить фразу 'Кто', `str2` – 'ты', `str3` – 'такой?'
3. Локальной переменной `concatenation` присвоить результат конкатенации 3-х строк: `str1`, `str2`, `str3`.
4. Вывести в документ содержимое переменной `concatenation`.

## Числовой тип (number)



В JavaScript существуют такие числовые типы:

- `int` — целое,
- `long` — длинное целое,
- `float` — вещественное.

Но **явное указание типов** в коде при объявлении переменной **не нужно!**

Используется неявное объявление, без указания конкретного типа данных:

```
var x = 5; // целое
var y = 5.6; // вещественное
```

Другие примеры:

```
var x = 5e3; // 5000
var y = 5e-3; // 0.005
```

Префикс 16-ной системы в javascript `0x`:

```
var x = 0xFF; // 255
```

Префикс 8-ной системы в javascript `0`:

```
var x = 045; // 37
```

```
var x = 0/0; // NaN - не число (not a number)
var x = 1/0; // Infinity (бесконечность)
```



**Задание Js 11.** Исправьте ошибки при объявлении локальных переменных во фрагментах кода:

1.

```
String s = "String";
int a = 10;
long b = 25;
```

2.

```
var name = "Меня зовут Вася ";
var 2b = 10;
Var _@c = 15;
alert(Name);
```



#### Вопросы для самоконтроля:

- В каком месте html-документа находится стандартное расположение скрипта JavaScript?
- С помощью какого метода осуществляется вывод диалогового окна?
- Что такое типы данных? Какие типы поддерживает JavaScript?
- С помощью каких операторов создаются комментарии?

### 3. Арифметические операторы javascript

Операторы предназначены для составления выражений.

Оператор применяется к одному или двум данным, которые в этом случае называются операндами.

Например, оператор сложения применяется к двум операндам ( $a + b$ ), а оператор логического отрицания — к одному операнду ( $\neg a$ ).



#### Операторы присваивания:

- `=` — обычная операция присваивания;

```
y = 5;
alert(y); // вывод 5
```

- `+=`, `-=` — присваивание со сложением или вычитанием;

```
y = 5;
alert(y-=2); // вывод 3
```

- `*=`, `/=` — присваивание с умножением или делением.

```
y = 5;
alert(y*=2); // вывод 10
```

#### Арифметические операторы:

- сложение в javascript: `+`

```
x + y;
y = 5;
x = y + 3; // равно 8
```

- вычитание в javascript: `-`

```
x - y;
```

- умножение в javascript: `*`

```
x * y;
```

- деление в javascript: `/`

```
x / y;
```

- javascript остаток от деления или деление по модулю: `%`

```
x = 8;
y = 5;
x % y; // 3
```

- javascript инкремент или увеличение на 1: `++`

```
x = 8;
x++; // 9

/* Префиксный инкремент выполняется перед использованием переменной, пример:*/
var number = 100;
++number; // примет значение 101

/*Постфиксный инкремент выполняется после использования переменной, пример:*/
var number = 100;
number++; // примет значение 101
alert(number); // выведет число 101
```

- javascript декремент или уменьшение на 1: `--`

```
y=9;
y--; // 8
```



**Задание Js 12.** Какие значения выведет в окно браузера следующий фрагмент кода?

```
var str = "20";
var a = 5;
document.write(str + a + "<br/>");
document.write(str - a + "<br/>");
document.write(str * 2 + "<br/>");
document.write(str / 2 + "<br/>");
```

**Пояснение:** Имейте в виду, так как переменная `str` является строковым типом, то переменная `a` типа `Number` неявно преобразуется в строку и далее производится операция конкатенации. Но если операция сложения для строкового значения существует, то операции деления и вычитания для строк отсутствуют, соответственно, действия будут происходить с числами.



**Js 13.** Необходимо написать сценарий, определяющий площадь прямоугольного треугольника по заданным катетам ( $S = ab/2$ ). Сценарий разместить в разделе `body` документа. С помощью скрипта вывести в окно браузера инкремент площади.

Алгоритм решения задачи на javascript:

- Инициализация двух переменных.
- Вычисление площади.
- Вывод инкремента с использованием метода `write()`.



**Вопросы для самоконтроля:**

- Что выполняют арифметические операции инкремент и декремент?
- Какова разница выполнения инкремента?:

```
S++
++S
```

- Как обозначается операция остаток от деления?

### Критерии оценки за практическую работу:

1. содержание Web-страницы соответствует заданию
2. текст на Web-странице не содержит ошибок
3. прописано название Web-страницы
4. правильный html-код
5. правильное название файла
6. правильное расположение файла на Личном диске студента

<b>5</b>	Пункты 1-6 верны
<b>4</b>	Один или два пункта из 1-4 содержат неточности. Пункты 5,6 верны
<b>3</b>	Три и более пунктов из 1-4 содержат неточности. Пункты 5,6 верны
<b>2</b>	Пункты 5,6 не верны или Пункты 1-4 не верны

## Практическая работа №14 «Использование обработчиков событий»

**Тема:** Основы JavaScript

**Количество часов:** 2

**Цели:** применение JavaScript

**Задачи:** добавить сценарии JS

## Порядок выполнения работы:

### Динамический HTML. Объекты и события

#### События **onClick**, **onMouseOut**, **onMouseOver** и **onMouseUp**

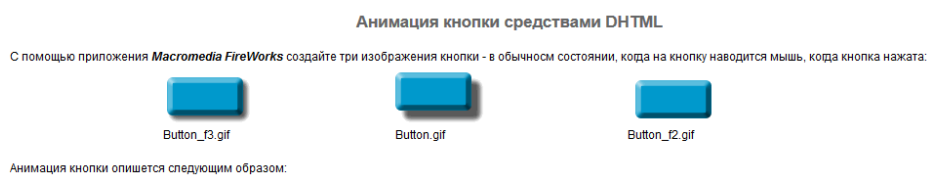
Событие **onClick** может использоваться, чтобы выполнить функции сценария, когда пользователь нажимает на специфический элемент. **Internet Explorer 4.0** и выше поддерживают это событие почти для всех HTML элементов, в то время как **Netscape** и более ранние версии **Internet Explorer** поддерживают это только в некоторых элементах (типа **links**, **image map** и т.д.).

Событие **onMouseOut** стартует, когда мышь пользователей оставляет область, определенную элементом.

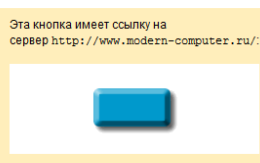
Событие **onMouseOver** стартует, когда мышь пользователей входит в область, определенную элементом.

Событие **onMouseUp** может использоваться, чтобы выполнить функции сценария после того, как мышь пользователей была нажата и отпущена - противоположность события **onMouseDown**.

В качестве примера рассмотрим анимацию кнопки с помощью динамического HTML



Пример:



Код HTML с использованием JavaScript:

```

```

Функция LoadServ() написана на Java-Script:

```
<script type="text/javascript">
<!--
function LoadServ(){
window.location="http://www.modern-computer.ru/";
}
//-->
</script>
```

### Использование форм в JavaScript.

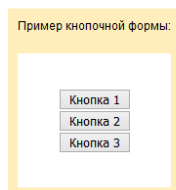
Предположим, что вы должны связать объект с обработчиком события объекта. Вы можете сделать это, добавив атрибут события к объекту HTML и установив его значение равным функции (или тексту функции). Например, чтобы связать функцию **JavaScript** по имени **MyFunction()** с событием кнопки **onClick**.

Кнопка с помощью атрибута **name=Button** запускает на выполнение сценария, когда пользователь нажимает ее.

#### Сценарий на JavaScript для кнопочной формы

В этом примере атрибут **name** для элемента формы **Button** при нажатии запускает сценарий, написанный на языке JavaScript

Пример:



Код HTML с использованием JavaScript:

```
<form>
<input type="submit" name="But1" value="Кнопка 1" onClick="Myf(1)">
<input type="submit" name="But2" value="Кнопка 2" onClick="Myf(2)">
<input type="submit" name="But3" value="Кнопка 3" onClick="Myf(3)">
</form>
```

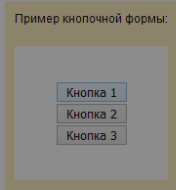
Функция Myf() написана на Java-Script:

```
function Myf(ButNum){
window.alert("Вы щелкнули на кнопке № "+ButNum);
}
```

**Сценарий на JavaScript для кнопочной формы**

В этом примере атрибут `name` для элемента формы `Button` при нажатии запускает сценарий, написанный на языке JavaScript

**Пример:**



**Код HTML с использованием JavaScript:**

```
<form>
  <button name="Кнопка 1" onClick="Myf(1)">Кнопка 1</button>
  <button name="Кнопка 2" onClick="Myf(2)">Кнопка 2</button>
  <button name="Кнопка 3" onClick="Myf(3)">Кнопка 3</button>
</form>
```

Функция `Myf()` написана на JavaScript:

```
function Myf(ButNum) {
  window.alert("Вы щелкнули на кнопке № "+ButNum);
}
```

## Событие `onChange`.

Если вы посещали Web-сервер *Microsoft*, то видели формы, используемые для перемещения. Вот как это работает: вы выбираете элемент из раскрывающегося списка, нажимаете кнопку и браузер открывает URL, связанный с этим элементом списка. Можно, используя сценарий, выполняющийся на сервере и обрабатывающий выбор пользователя, однако к рассмотрению предлагается метод, где на стороне клиента в Web-странице производится обработка события `onChange`.

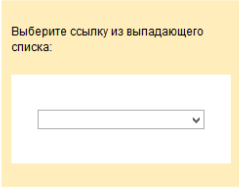
В примере приведён код HTML для формы с одиночным раскрывающимся списком, который открывает выбираемую пользователем Web-страницу при наступлении события `onChange`

**Сценарий на JavaScript, обрабатывающий событие `onChange`**

Функция `OpenURL()` связана с событием `onChange` дескриптора `<select>`. Когда пользователь выбирает элемент из списка формы, функция `OpenURL()` устанавливает значение `window.location` равным указанному в ссылке URL.

Если вы хотите сделать подобное в своей Web-странице, то должны модифицировать список формы таким образом, чтобы он содержал вход для каждого Web-узла, к которому нужен доступ пользователя. Затем надо добавить строки к функции `OpenURL()`, которые проверяют выбор пользователя и открывают соответствующие URL в браузере. Самый простой способ - это скопировать одну из существующих строк, заменив номер индекса так, чтобы он соответствовал форме, и затем заменить URL, присваиваемый `window.location`. Только не забудьте, что первый элемент в списке имеет индекс 0, второй - 1 и т.д.

**Пример:**



**Код HTML с использованием JavaScript:**

```
<form name="navigate">
  <select name="list" size="1" onChange="OpenURL(document.navigate.list.selectedIndex)">
    <option name="home">Домашняя страница
    <option name="serv">Практикум по Web
    <option name="trashop">Практикум по PhotoShop
    <option name="mail">www.mail.ru
    <option name="rambler">www.rambler.ru
    <option name="yandex">www.yandex.ru
  </select>
</form>
```

Функция `OpenURL()` написана на JavaScript:

```
function OpenURL(Index) {
  if (Index==1) {window.location = "http://www.modern-computer.ru/";}
  if (Index==2) {window.location = "http://www.modern-computer.ru/practice/web-design/practic-web-design-main.html";}
  if (Index==3) {window.location = "http://www.modern-computer.ru/practice/photoshop/photoshop-main.html";}
  if (Index==4) {window.location = "http://www.mail.ru";}
  if (Index==5) {window.location = "http://www.rambler.ru";}
  if (Index==6) {window.location = "http://www.yandex.ru";}
}
```

## Методы `click` и `alert`. Свойство `returnValue`

Метод `object.Click()` может использоваться, чтобы «нажать» определенный объект через создание сценария, вызывая событие `onClick` для специфического элемента.

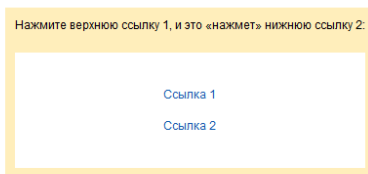
Метод `alert` выводит информационное окно.

Свойство `returnValue` объекта `event` определяет возвращаемое значение для события.

### Сценарий на JavaScript, вызывающий методы `object.Click()` и `alert`

В этом примере показано, как при нажатии на [ссылку 1](#) происходит вызов функции `lnkoncl()`, которая в конечном итоге «нажимает» нижнюю [ссылку 2](#).

Пример:



Код HTML с использованием JavaScript:

```
<a href="#" id="Link1"
title="Ссылка 1" class="ixx"
onclick="lnkoncl()" >
Ссылка 1</a>
<a href="#" id="Link2"
title="Ссылка 2" class="ixx"
onclick="alert('Нажата Ссылка 2')">
Ссылка 2</a>
```

Функция `lnkoncl()` написана на JavaScript:

```
function lnkoncl(){
window.alert ("Нажата Ссылка 1");
self.event.returnValue="False";
Link2.click();
}
```

Внимание!!! Пример корректно работает только в браузере MS Internet Explorer.

## Метод `insertAdjacentHTML`

Метод `insertAdjacentHTML` может использоваться для того, чтобы вставить новый HTML-элемент в документ. При этом удаленный предыдущий HTML-элемент удален не будет.

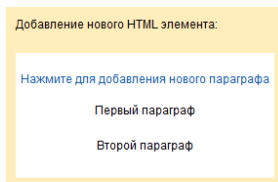
Возможные значения для позиционирования HTML-элементов, используемые методом `insertAdjacentHTML`:

- `BeforeBegin`
- `AfterBegin`
- `BeforeEnd`
- `AfterEnd`

### Позиционирование элементов с помощью метода `insertAdjacentHTML`

В текущем документе (`document`) из общей коллекции всех тегов (`all`) выбираются эти `<p>` и `<a>` (`tags("p")`, `tags("a")`) и перед (или после) элемента (в нашем примере тегов `<p>` и `<a>`) с номером `item` (номер) (нумерация начинается с нуля) вставляется текст и код HTML (`insertAdjacentHTML`).

Пример:



Код HTML с использованием JavaScript:

```
<a href="#" id="Link3"
title="Ссылка" class="ixx"
onclick="fClick()" >
Нажмите для добавления нового параграфа
</a>
<p>Первый параграф</p>
<p>Второй параграф</p>
```

Функция `fClick()` написана на JavaScript:

```
var i=2;
function fClick(){
i=i+1;
ptxt="<p>Вот вам "+i+"-й параграф</p>";
document.all.tags("p").item(i).insertAdjacentHTML("BeforeBegin",
ptxt);
document.all.tags("p").item(i).insertAdjacentHTML("AfterEnd",
"<a HREF=#null TITLE=Новая ссылка onclick=fClick()>
... и новая ссылка</a>");
document.all.tags("a").item(i-2).insertAdjacentHTML("AfterEnd",
"<br>" );
}
```

Внимание!!! Пример корректно работает только в браузере MS Internet Explorer.

## Метод `insertAdjacentText`

Метод `insertAdjacentText` идентичен методу `insertAdjacentHTML`, за исключением того, что он вставляет неотформатированный текст, а не HTML.

Использование метода `insertAdjacentText`

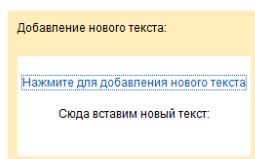


### Размещение текста на странице с помощью метода insertAdjacentText

На текущей web-странице Метод `insertAdjacentText` вставляет текст внутри или рядом с определенным элементом страницы. В нашем примере таким элементом стрницы являются эти `<div>`, после которых вставляется текст с номером `item (номер)` (нумерация начинается с нуля).

**Пример:**

**Код HTML с использованием JavaScript:**



```
<a href="#" id="Link4"
title="Ссылка" class="ixx"
onclick="insTxt()" ">
Нажмите для добавления нового текста
</a>
<div>Сюда вставим новый текст:</div>
```

**Функция insTxt( ) написана на Java-Script:**

```
var i = 0;
function insTxt(){
    i = i + 1;
    ptxt="<p>Это текст № " + i + ". Он вставлен после нажатия на ссылку";
    document.all.tags("div").item(0).insertAdjacentHTML("AfterEnd", "<br><br>");
    document.all.tags("div").item(0).insertAdjacentText("AfterEnd", ptxt);
}
```

Внимание!!! Пример корректно работает только в браузере MS Internet Explorer.

### Критерии оценки за практическую работу:

1. содержание Web-страницы соответствует заданию
2. текст на Web-странице не содержит ошибок
3. прописано название Web-страницы
4. правильный html-код
5. правильное название файла
6. правильное расположение файла на Личном диске студента

<b>5</b>	Пункты 1-6 верны
<b>4</b>	Один или два пункта из 1-4 содержат неточности. Пункты 5,6 верны
<b>3</b>	Три и более пунктов из 1-4 содержат неточности. Пункты 5,6 верны
<b>2</b>	Пункты 5,6 не верны или Пункты 1-4 не верны

### Практическая работа №15

#### «Использование условного оператора IF, SWITCH»

**Тема:** Основы JavaScript

**Количество часов:** 4

**Цели:** применение JavaScript

**Задачи:** применить условный оператор на JS

**Порядок выполнения работы:**

Условные операторы используются для выполнения различных действий, основанных на различных условиях.

#### Условные операторы

Очень часто, когда вы пишете код, вы хотите, чтобы выполнять различные действия для различных решений.

Вы можете использовать условные операторы в коде, чтобы сделать это.

В JavaScript мы имеем следующие условные операторы:

- Используйте **if** указать блок кода , который будет выполняться, если указанное условие истинно
- Используйте **else** , чтобы указать блок кода , который будет выполняться, если же условие ложно
- Используйте **else if** , **else if** задать новое условие , чтобы проверить, если первое условие ложно

- Используйте **switch** , чтобы указать множество альтернативных блоков кода , которые будут выполняться

### Если заявление

Используйте , **if** заявление , чтобы указать блок кода JavaScript будет выполняться , если условие истинно.

### Синтаксис

```
if (condition) {
    block of code to be executed if the condition is true
}
```

Обратите внимание , что **if** в строчными буквами. Прописные буквы ( If или IF ) будет генерировать ошибку JavaScript.

### пример

Сделать "Good day" приветствие , если час меньше 18:00:

```
if (hour < 18) {
    greeting = "Good day";
}
```

Результат приветствия будет:

Good dayGood day

### Остальное Заявление

Используйте **else** заявление , чтобы указать блок кода , который будет выполняться , если условие ложно.

```
if (condition) {
    block of code to be executed if the condition is true
} else {
    block of code to be executed if the condition is false
}
```

### пример

Если час меньше , чем 18, создать "Good day" приветствие, в противном случае "Good evening" :

```
if (hour < 18) {
    greeting = "Good day";
} else {
    greeting = "Good evening";
}
```

Результат приветствия будет:

Good dayGood day

### else if , else if заявление

Используйте **else if** , **else if** заявление , чтобы указать новое условие , если первое условие ложно.

### Синтаксис

```
if (condition1) {
    block of code to be executed if condition1 is true
} else if (condition2) {
    block of code to be executed if the condition1 is false and condition2 is true
} else {
    block of code to be executed if the condition1 is false and condition2 is false
}
```

### пример

Если время меньше , чем 10:00, создать "Good morning" приветствие, если нет, но время меньше 20:00, создать "Good day" приветствие, в противном случае "Good evening" :

```
if (time < 10) {
    greeting = "Good morning";
} else if (time < 20) {
```

```

    greeting = "Good day";
} else {
    greeting = "Good evening";
}

```

Результат приветствия будет:

Good dayGood day

#### Критерии оценки за практическую работу:

1. содержание Web-страницы соответствует заданию
2. текст на Web-странице не содержит ошибок
3. прописано название Web-страницы
4. правильный html-код
5. правильное название файла
6. правильное расположение файла на Личном диске студента

<b>5</b>	Пункты 1-6 верны
<b>4</b>	Один или два пункта из 1-4 содержат неточности. Пункты 5,6 верны
<b>3</b>	Три и более пунктов из 1-4 содержат неточности. Пункты 5,6 верны
<b>2</b>	Пункты 5,6 не верны или Пункты 1-4 не верны

### Практическая работа №16 «Использование операторов цикла»

**Тема:** Основы JavaScript

**Количество часов:** 2

**Цели:** применение JavaScript

**Задачи:** применить операторы циклов на JS

**Порядок выполнения работы:**

Задачи для решения

1. Выведите с помощью цикла столбец чисел от 1 до 100.
2. Выведите с помощью цикла столбец чисел от 100 до 1.
3. Выведите с помощью цикла столбец четных чисел от 1 до 100.
4. Заполните массив 10-ю иксами с помощью цикла.
5. Заполните массив числами от 1 до 10 с помощью цикла.
6. Заполните массив 10-ю случайными числами (дробями) от 0 до 1 с помощью цикла. Дробь округляйте до двух знаков в дробной части.
7. Заполните массив 10-ю случайными числами от 1 до 10 с помощью цикла.
8. Дан массив с числами. С помощью цикла выведите только те элементы массива, которые больше нуля и меньше 10-ти.
9. Дан массив с числами. С помощью цикла проверьте, что в нем есть элемент со значением **5**. Как только будет найден первый такой элемент - выведите 'Есть' и оборвите цикл. Если такого элемента нет - ничего не выводите.
10. Дан массив с числами. С помощью цикла найдите сумму элементов этого массива.
11. Дан массив с числами. С помощью цикла найдите сумму квадратов элементов этого массива.
12. Дан массив с числами. Найдите среднее арифметическое его элементов (сумма элементов, делить на количество).

#### Решение

*В следующих задачах вы должны вспомнить работу с циклами JavaScript.*

Выведите с помощью цикла столбец чисел от 1 до 100.

Решение:

```
for (var i = 1; i <= 100; i++) {  
    document.write(i + '<br>');  
}
```

Выведите с помощью цикла столбец чисел от 100 до 1.

Решение:

```
for (var i = 100; i > 0; i--) {  
    document.write(i + '<br>');  
}
```

Выведите с помощью цикла столбец четных чисел от 1 до 100.

Решение:

```
for (var i = 2; i < 100; i += 2) {  
    document.write(i + '<br>');  
}
```

Заполните массив 10-ю иксами с помощью цикла.

Решение:

```
var arr = [];  
for (var i = 0; i < 10; i++) {  
    arr[i] = 'x';  
}  
console.log(arr);
```

Заполните массив числами от 1 до 10 с помощью цикла.

Решение:

```
var arr = [];  
for (var i = 1; i <= 10; i++) {  
    arr.push(i);  
}  
console.log(arr);
```

Заполните массив 10-ю случайными числами (дробями) от 0 до 1 с помощью цикла. Дробь округляйте до двух знаков в дробной части.

Решение:

```
var arr = [];  
for (var i = 0; i < 10; i++) {  
    arr.push(Math.random().toFixed(2));  
}  
console.log(arr);
```

Заполните массив 10-ю случайными числами от 1 до 10 с помощью цикла.

Решение:

```
var arr = [];  
for (var i = 0; i < 10; i++) {  
    arr.push(Math.round(Math.random() * 10) + 1);  
}  
console.log(arr);
```

Дан массив с числами. С помощью цикла выведите только те элементы массива, которые больше нуля и меньше 10-ти.

Решение:

```
var arr = [1, 2, 4, -1, -3, 8, 7, 0, -5, 18, 12];
```

```

for (var i = 0; i < arr.length; i++) {
    if(arr[i] > 0 && arr[i] < 10) {
        console.log(arr[i]);
    }
}

```

Дан массив с числами. С помощью цикла проверьте, что в нем есть элемент со значением **5**. Как только будет найден первый такой элемент - выведите 'Есть' и оборвите цикл. Если такого элемента нет - ничего не выводите.

Решение:

```

var arr = [1, 2, 3, 4, 5, 6, 7, 8, 9];
for (var i = 0; i < arr.length; i++) {
    if(arr[i] == 5) {
        alert('Есть!');
        break;
    }
}

```

Дан массив с числами. С помощью цикла найдите сумму элементов этого массива.

Решение:

```

var arr = [1, 2, 3, 4, 5];
var sum = 0;
for (var i = 0; i < arr.length; i++) {
    sum += arr[i];
}
alert(sum);

```

Дан массив с числами. С помощью цикла найдите сумму квадратов элементов этого массива.

Решение:

```

var arr = [1, 2, 3, 4, 5];
var sum = 0;
for (var i = 0; i < arr.length; i++) {
    sum += arr[i] * arr[i];
}
alert(sum);

```

Дан массив с числами. Найдите среднее арифметическое его элементов (сумма элементов, делить на количество).

Решение:

```

var arr = [1, 2, 3, 7, 6, 9];
var sum = 0;
for (var i = 0; i < arr.length; i++) {
    sum += arr[i];
}
var result = sum / arr.length;
alert(result);

```

### Критерии оценки за практическую работу:

1. содержание соответствует заданию
2. текст не содержит ошибок
3. прописано название Web-страницы
4. правильный html-код
5. правильное название файла
6. правильное расположение файла на Личном диске студента

<b>5</b>	Пункты 1-6 верны
<b>4</b>	Один или два пункта из 1-4 содержат неточности. Пункты 5,6 верны
<b>3</b>	Три и более пунктов из 1-4 содержат неточности. Пункты 5,6 верны
<b>2</b>	Пункты 5,6 не верны или Пункты 1-4 не верны

## Практическая работа №17 «Работа с датой и временем»

**Тема:** Основы JavaScript

**Количество часов:** 2

**Цели:** применение JavaScript

**Задачи:** применить форматы Дата, Время на JS

**Порядок выполнения работы:**

Объект Date позволяет работать с датами (годы, месяцы, дни, часы, минуты, секунды и миллисекунды)

**Форматы JavaScript Дата**

Свидание JavaScript может быть записан в виде строки:

Tue Feb 19 2019 18:27:01 GMT+0500 (Екатеринбург, стандартное время) **Вс 6 ноября 2016**

**6:57:46 GMT + 0800 (中国标准时间)**

или в качестве номера:

1550582821195 **1478386666530**

Даты, написанные в виде чисел, определяет количество миллисекунд, прошедших с 1 января 1970 года, 00:00:00.

**Отображение Даты**

В этом уроке мы используем скрипт для отображения даты внутри `<p>` элемента с `id="demo"` :

**пример**

```
<p id="demo"></p>
```

```
<script>
```

```
document.getElementById("demo").innerHTML = Date();
```

```
</script>
```

Сценарий выше говорит: присвоить значение `Date()` к содержимому ( `innerHTML` ) элемента с `id="demo"` .

Вы узнаете, как отображать дату, в более удобном для чтения формате, в нижней части этой страницы.

**Создание объектов Date**

Объект Date позволяет нам работать с датами.

Дата, состоит из год, месяц, день, час, минута, секунда, и миллисекунды.

Дата объекты создаются с **`new Date()`** конструктор.

Есть **4 способа** инициирования даты:

```
new Date()
```

```
new Date(milliseconds)
```

```
new Date(dateString)
```

```
new Date(year, month, day, hours, minutes, seconds, milliseconds)
```

Используя `new Date()` , создает новый объект даты с **текущей датой и временем:**

**пример**

```
<script>
```

```
var d = new Date();
```

```
document.getElementById("demo").innerHTML = d;  
</script>
```

Использование новой даты (**строка даты**), создает новый объект даты с **указанной даты и времени**:

**пример**

```
<script>  
var d = new Date("October 13, 2014 11:13:00");  
document.getElementById("demo").innerHTML = d;  
</script>
```

Допустимые строки даты (форматы даты) описаны в следующей главе.

Используя новую дату (**число**), создает новый объект даты, как **нулевое время плюс число**.

Ноль времени 01 January 1970 00:00:00 UTC. Номер указывается в миллисекундах:

**пример**

```
<script>  
var d = new Date(86400000);  
document.getElementById("demo").innerHTML = d;  
</script>
```

JavaScript сроки рассчитываются в миллисекундах с 01 января 1970 00:00:00 всемирного времени (UTC). Однажды содержит 86,400,000 миллисекунду.

Использование new Date (**7 номеров**), создает новый объект даты с **указанной даты и времени**:

В 7 номеров указать год, месяц, день, час, минута, секунда, и миллисекунду, в таком порядке:

**пример**

```
<script>  
var d = new Date(99,5,24,11,33,30,0);  
document.getElementById("demo").innerHTML = d;  
</script>
```

Варианты приведенном выше примере давайте опустим любой из последних 4-х параметров:

**пример**

```
<script>  
var d = new Date(99,5,24);  
document.getElementById("demo").innerHTML = d;  
</script>
```

JavaScript имеет значение от 0 месяцев до 11 января 0. декабрь 11.

## Методы Дата

Когда объект Date создается целый ряд **методов** позволяют работать на нем.

Современные методы позволяют получить и установить год, месяц, день, час, минута, секунда, и миллисекунду объектов, используя либо местное время или UTC (универсальное, или GMT) время.

Методы Дата рассматриваются в следующей главе.

## Отображение Даты

При отображении объекта даты в HTML, он автоматически преобразуется в строку, с **toString()** метод.

**пример**

```
<p id="demo"></p>  
<script>  
d = new Date();  
document.getElementById("demo").innerHTML = d;  
</script>
```

Такой же как:

```
<p id="demo"></p>
```

```
<script>
```

```
d = new Date();
```

```
document.getElementById("demo").innerHTML = d.toString();
```

```
</script>
```

**toUTCString()** метод преобразует дату в формате UTC строки (стандартный дисплей даты).

**пример**

```
<script>
```

```
var d = new Date();
```

```
document.getElementById("demo").innerHTML = d.toUTCString();
```

```
</script>
```

**toDateString()** метод преобразовывает дату в более удобном для чтения формате:

**пример**

```
<script>
```

```
var d = new Date();
```

```
document.getElementById("demo").innerHTML = d.toDateString();
```

```
</script>
```

Дата объекты являются статическими. Время компьютер тикают, но дата объекты, после создания, не являются

#### Критерии оценки за практическую работу:

1. содержание Web-страницы соответствует заданию
2. текст на Web-странице не содержит ошибок
3. прописано название Web-страницы
4. правильный html-код
5. правильное название файла
6. правильное расположение файла на Личном диске студента

<b>5</b>	Пункты 1-6 верны
<b>4</b>	Один или два пункта из 1-4 содержат неточности. Пункты 5,6 верны
<b>3</b>	Три и более пунктов из 1-4 содержат неточности. Пункты 5,6 верны
<b>2</b>	Пункты 5,6 не верны или Пункты 1-4 не верны

### Практическая работа №18 «Работа со строками. Решение задач»

**Тема:** Основы JavaScript

**Количество часов:** 6

**Цели:** применение JavaScript

**Задачи:** проработать текстовые строки на JS

**Порядок выполнения работы:**

Строки JavaScript используются для хранения и обработки текста.

**JavaScript Строки**

Строка JavaScript просто хранит ряд символов , как "John Doe" .

Строка может быть любой текст в кавычках. Вы можете использовать одинарные или двойные кавычки:

**пример**

```
var carname = "Volvo XC60";
```

```
var carname = 'Volvo XC60';
```



Вы можете использовать кавычки внутри строки, до тех пор, пока они не совпадают кавычки вокруг строки:

**пример**

```
var answer = "It's alright";  
var answer = "He is called 'Johnny'";  
var answer = 'He is called "Johnny"';
```

**Длина строки**

Длина строки находится во встроенной в **длину** не подвижности:

**пример**

```
var txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";  
var sln = txt.length;
```

**Специальные символы**

Поскольку строки должны быть написаны в кавычках, JavaScript будет неправильно понять эту строку:

```
var y = "We are the so-called "Vikings" from the north."
```

Строка будет нарезанной на "We are the so-called " .

Решение , чтобы избежать этой проблемы, является использование **символа \ побег**.

Символ обратной косой черты побег превращает специальные символы в строку символов:

**пример**

```
var x = 'It\'s alright';  
var y = "We are the so-called \"Vikings\" from the north."
```

Экранирующий символ (\) также может быть использован для вставки других специальных символов в строке.

Это список специальных символов, которые могут быть добавлены в виде текстовой строки со знаком обратной косой черты:

<b>Код</b>	<b>Выходы</b>
------------	---------------

\'	апостроф
\"	двойные кавычки
\\	обратный слэш
\n	новая линия
\r	возврат каретки
\t	таб
\b	возврат на одну позицию
\f	прогон страницы

**Разрыв строк длинного кода**

Для лучшей читаемости, программисты часто любят, чтобы избежать строк кода длиной более 80 символов.

Если оператор JavaScript не помещается на одной строке, лучшее место, чтобы сломать это после того, как оператор:

**пример**

```
document.getElementById("demo").innerHTML =  
"Hello Dolly.";
```

Вы можете также разбить строку кода **внутри текстовой строки с одной обратной косой черты**:

**пример**

```
document.getElementById("demo").innerHTML = "Hello \  
Dolly!";
```

\ Метод не является ECMAScript (JavaScript) стандарт.

Некоторые браузеры не позволяют пробелы , лежащие в основе \ характер.

Самый безопасный (но немного медленнее) способ сломать длинную строку, чтобы использовать строку добавления:

**пример**

```
document.getElementById("demo").innerHTML = "Hello" +  
"Dolly!";
```

Вы не можете разбить строку кода с помощью обратной косой черты:

**пример**

```
document.getElementById("demo").innerHTML = \  
"Hello Dolly!";
```

**Строки могут быть объектами**

Обычно, строки JavaScript примитивные значения, созданные из литералов: **var firstName = "John"**

Но строки также могут быть определены как объекты с ключевыми словами новый: **var firstName = new String("John")**

**пример**

```
var x = "John";  
var y = new String("John");
```

```
// typeof x will return string
```

```
// typeof y will return object
```

Не создавайте строки в качестве объектов. Это замедляет скорость выполнения.

**new** ключевое слово усложняет код. Это может привести к неожиданным результатам:

При использовании оператора равенства **==**, равные строки выглядят равны:

**пример**

```
var x = "John";  
var y = new String("John");
```

```
// (x == y) is true because x and y have equal values
```

При использовании оператора **===** равенство, равные строки не равны, потому что оператор **===** ожидает, что равенство в обоих типов и стоимости.

**пример**

```
var x = "John";  
var y = new String("John");
```

```
// (x === y) is false because x and y have different types (string and object)
```

Или еще хуже. Объекты нельзя сравнивать:

**пример**

```
var x = new String("John");  
var y = new String("John");
```

```
// (x == y) is false because x and y are different objects
```

```
// (x == x) is true because both are the same object
```

Объекты JavaScript нельзя сравнивать.

**Критерии оценки за практическую работу:**

1. содержание Web-страницы соответствует заданию
2. текст на Web-странице не содержит ошибок
3. прописано название Web-страницы
4. правильный html-код
5. правильное название файла
6. правильное расположение файла на Личном диске студента

<b>5</b>	Пункты 1-6 верны
----------	------------------

<b>4</b>	Один или два пункта из 1-4 содержат неточности. Пункты 5,6 верны
<b>3</b>	Три и более пунктов из 1-4 содержат неточности. Пункты 5,6 верны
<b>2</b>	Пункты 5,6 не верны или Пункты 1-4 не верны

### **Практическая работа №19**

**«Установка и настройка ПО. Операторы в языке PHP. Создание формы. Обработка массивов данных. Работа со строками. Работа с файлами. Проверка данных. Обработка запросов с помощью PHP»**

**Тема:** PHP

**Количество часов:** 8

**Цели:** использование средств языка программирования PHP

**Задача 1:** Установка и настройка ПО

## Преимущества использования PHP

Для начала рассмотрим **положительные моменты использования** языка программирования PHP

- язык очень прост в изучении;
- PHP поддерживается практически на всех наиболее используемых платформах и почти во всех ОС;
- в основе языка лежат две наиболее популярные парадигмы программирования – объектная и процедурная;
- язык — скриптовый, это означает, что PHP-код можно встраивать непосредственно в HTML-документы;
- эффективное взаимодействие с широким спектром разнообразных баз данных, файловых систем, электронной почтой;
- встроена поддержка достаточно большого количества современных баз данных, включены функции для работы с файловой системой.

## Краткий алгоритм работы с DENWER

Для апробации результатов заданий необходимо иметь возможность работы с PHP. Мы предлагаем установить на локальной машине пакет denwer.

При работе с Denwer:

- разместить файл **index.php** в каталоге: `z:\home\localhost\www\ваш_сайт`
- в браузере ввести: `http://localhost/ваш_сайт/`

## Виды встраиваемого кода



- Стиль XML (является предпочтительным стилем)

```
1 <?php
2 PHP-код;
3 ?>
```

- Сокращенный стиль (не рекомендуется)

```
1 <?
2 PHP-код;
3 ?>
```

**Пример встраиваемого использования:**

```
<html>
<head>
<title>HTML_PHP</title>
<body>
<?php
PHP-код;
?>
</body></html>
```

## Задача 2: Операторы вывода, присваивания

### Операторы вывода

```
1 <?php
2 echo "Я изучаю PHP";
3 print "Я изучаю PHP";
4 ?>
```

```
<body>
<?php
echo "Текст скрипта PHP";
?>
<p>Обычный текст</p>
<?php
echo "Продолжение текста скрипта PHP";
?>
```

Операторы завершаются точкой с запятой

Можно вставлять скрипт частями. Части скрипта воспринимаются, как один

Рис. 1.1. Встраивание скрипта php. Синтаксис

Использование HTML-тегов внутри строк, выводимых оператором echo:

```
1 <?php
2 echo "<b>я изучаю PHP</b>";
3 echo "<h2>пример заголовка</h2>";
4 echo "<hr>";
5 ?>
```

Пример:

```
<body>
<p>
<?php
echo "<h1>Это базовый документ PHP</h1>";
?>
</p><p>
<?
echo "<p>PHP - это интересно</p>";
?>
</p>
</body>
```

Рис. 1.2. Пример использования тегов в коде PHP

Результат:

## Это базовый документ PHP

PHP – это интересно

## Типы данных

В PHP поддерживаются следующие типы данных:

### Скалярные типы данных:

- boolean (логический)
- integer (целочисленный)
- float (число с плавающей точкой)
- string (строковый)

### Смешанные типы:

- array (массив)
- object (объектный)
- callable

### Специальные типы:

- resource
- NULL

Примеры:

```
1 <?php
2 $x_bool = TRUE; // логический
3 $x_str1 = "foo"; // строковый
4 $x_str2 = 'foo'; // строковый
5 $x_int = 12; // целочисленный
6
7 echo gettype($x_bool); // выводит: boolean
8 echo gettype($x_str1); // выводит: string
9
10 ?>
```

## Комментарии в PHP

1.

```
<?php
# строчный комментарий
If (!$name){echo $error;}
?>
```

2.

```
<?php
$name=""; //строчный комментарий
?>
```

3.

```
<?php
/* блочный комментарий
echo "PHP-код";
echo "снова PHP-код";
*/
?>
```

## Оператор присваивания и слияние переменных

Переменные в php используются со знаком **\$** непосредственно перед идентификатором:

```
$myname = "Иван";
```

**Вывод значений переменных:**

```
echo "Я изучаю PHP";  
// или  
print "Я изучаю PHP";
```

**Присваивание значений других переменных:**

```
$test1 = $test2;  
echo $a, " ", $b;  
// или  
echo $a; echo " "; echo $b;
```

**Примеры:**

```
1 <?php  
2 $first = 'text1';  
3 $second = $first;  
4 $first = 'new text';  
5  
6 echo "Переменная first равна $first <br>";  
7 echo "Переменная second равна". $second;  
8 ?>
```

Как видно из примера, **оператор присваивания в php — это знак равенства.**

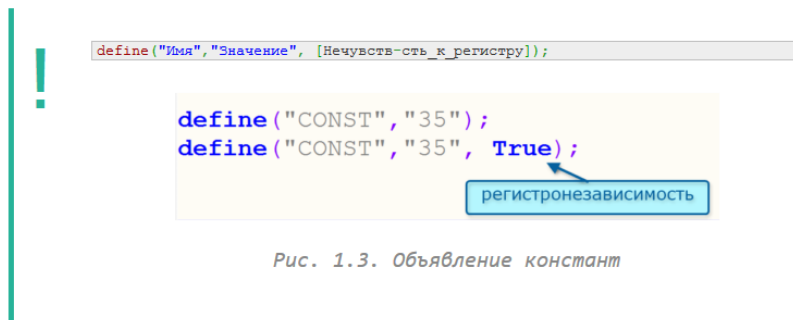
**Результатом** выполнения программы будет:

Переменная first равна new text

Переменная second равна text1

### Задача 3: Арифметические операции

#### Объявление констант и арифметические операции



Пример:

```
<?php
define("PASSWORD", "qwerty");
define("PI", "3.14", True);
echo (PASSWORD); //выведет значение qwerty

// ниже выведется «password» и предупреждение, поскольку мы ввели регистрозависимую констант
echo (password);
echo pi; //выведет 3.14, поскольку константа PI регистронезависима
?>
```

Обозначение операции	Наименование операции	Пример
+	Сложение	<code>\$a + \$b</code>
-	Вычитание	<code>\$a - \$b</code>
*	Умножение	<code>\$a * \$b</code>
/	Деление	<code>\$a / \$b</code>
%	Остаток от деления	<code>\$a % \$b</code>
++	Инкремент (увеличение значения переменной на единицу):	<code>\$a++;</code>
--	Декремент (уменьшение значения переменной на единицу)	<code>\$a--;</code>

#### Операторы присваивания и логические операторы

Обозначение операции	Пример
=	<code>\$a = (\$b = 4) + 5;</code> ( <code>\$a</code> будет равна 9, <code>\$b</code> будет равна 4)
+=	<code>\$a+=5;</code> (эквивалентно <code>\$a = \$a + 5;</code> )
.=	<code>\$b = "Привет"; \$b .= "всем";</code> (эквивалентно <code>\$b = \$b . «всем»;</code> ) В результате: <code>\$b=Привет всем</code>

Обозначение операции	Наименование операции	Пример
&&	и	<code>\$a &amp;&amp; \$b</code>
	или	<code>\$a    \$b</code>
!	Если <code>\$a=True</code> , то <code>!\$a=False</code> и наоборот	<code>!\$a</code>





#### Задание php1:

- Создайте переменную `$name` и присвойте ей строковое значение, содержащее Ваше имя;
- Создайте переменную `$age` и присвойте ей числовое значение — Ваш возраст;
- Выведите значения переменных на экран, каждое с новой строки.

## Задача 4: Условный оператор

### Условные операторы php

Условные операторы php представлены тремя основными конструкциями:

- оператор условия **if**,
- оператор переключения **switch**
- и **тернарный оператор**.

Рассмотрим подробнее каждый из них.

### PHP оператор if

```
$a=1;
if ($a==1)
{
    // если условие истинно
    echo "a равно 1";
    $a++;
}
```

Рис 3.1. Условный оператор IF, укороченный вариант

```
$a=1;
if ($a==1)
{
    // если условие истинно
    echo "a равно 1";
    $a++;
}
```

```

else
{
    // если условие ложно
    echo "а не равно 1";
    $a--;
}

```

Рис. 3.2. Синтаксис условного оператора IF ELSE

```

$a=1;
if ($a==1)
{
    // если условие истинно
    echo "а равно 1";
    $a++;
}
elseif($a==0)
{
    // если альтернативное условие истинно
    echo "а равно 0";
}
else
{
    // если условие ложно
    echo "а не равно 1 и не равно 0";
    $a--;
}

```

Рис. 3.3. Полный синтаксис условного оператора IF elseif

Подытожим:

#### Полный синтаксис:

```

1 if (условие)
2 {
3     // если условие истинно
4     operator1;
5     operator2;
6 }
7 elseif(условие)
8 {
9     operator1;
10    ...
11 }
12 else
13 {
14     // если условие ложно
15     operator1;
16     operator2;
17 }

```

- Укороченный синтаксис может не содержать части конструкции с `else` и не содержать дополнительного условия `elseif`
- Вместо служебного слова `elseif` можно писать `else if` (отдельно)
- В одной конструкции `if` может быть несколько `elseif`. Будет выполнено первое встретившееся выражение `elseif` равное `TRUE`.
- В случае наличия альтернативного условия `elseif` конструкция `else` должна идти последней в синтаксисе.

В условном операторе может использоваться двоеточие `:` вместо фигурных скобок. В таком случае заканчивается оператор служебным словом `endif`

```
$a=1;
if ($a==1):
    // если условие истинно
    echo "a равно 1";
    $a++;
elseif($a==0):
    // если альтернативное условие истинно
    echo "a равно 0";
else:
    // если условие ложно
    echo "a не равно 1 и не равно 0";
    $a--;
endif;
```

Рис. 3.4. Условный оператор If и Endif в php

Пример:

```
1 if($x > $y):
2     echo $x." больше, чем ".$y;
3 elseif($x == $y): // при использовании ":" нельзя писать отдельно else if
4     echo $x." равно ".$y;
5 else:
6     echo $x." не > и не = ".$y;
7 endif;
```



**Важно:** При использовании в конструкции двоеточия вместо фигурных скобок `elseif` нельзя писать в два слова!

## ЛОГИЧЕСКИЕ ОПЕРАЦИИ В УСЛОВИИ

В условии `if` в скобках могут присутствовать следующие операции:

<code>==</code>	равенство
<code>!=</code>	неравенство
<code>&gt;</code>	больше
<code>&lt;</code>	меньше
<code>&gt;=</code>	больше или равно
<code>&lt;=</code>	меньше или равно



**Пример:** проверить значение числовой переменной: если оно меньше либо равно 10, — выдавать сообщение «число меньше или равно 10», в обратном случае выдавать сообщение «число больше 10»

Решение:

```
1 $number=15;
2 if ($number <=10) {
3     echo "число меньше или равно 10";
4 }
5 else {
6     echo "число больше 10";
7 }
```

Блоки php кода можно разрывать, рассмотрим пример:



**Пример:** Выводить на экран html-код «a равно 4», если переменная `$a` действительно равна 4

### 1 Решение:

```
1 <?php
2 if ($a == 4):
3     echo "a равно 4";
4 ?>
```

### 2 Решение:

```
1 <?php if ($a == 4): ?>
2 А равно 4
3 <?php endif; ?>
```



**Задание php 3\_1:** Выводить перевод цветов с английского языка на русский, проверяя значение переменной (в которой присвоен цвет: `$a='blue'` )



**Задание php 3\_2:** Найти максимальное из трех чисел

## ОПЕРАЦИИ СРАВНЕНИЯ И ПРАВИЛО ЛЖИ

В конструкции `if` в скобках должно находиться логическое выражение или переменная, которые рассматриваются с точки зрения алгебры логики, возвращающая значения либо `true` либо `false`

Т.е. в качестве условия может выступать единственная переменная. Рассмотрим пример:

```
1 $a=1;
2 if ($a) {
3     echo $a;
4 }
```

В примере транслятор языка php будет рассматривать переменную в скобках на правило лжи:

**Правило ЛЖИ или что считается ложным:**

- логическое **False**
- целый ноль (**0**)
- действительный ноль (**0.0**)
- пустая строка и строка «**0**»
- массив без элементов
- объект без переменных
- специальный тип **NULL**

Таким образом, в рассмотренном примере переменная `$a` равна единице, соответственно условие будет истинным и оператор `echo $a;` выведет на экран значение переменной.



**Задание php 3\_3:** дана переменная `a` со строковым значением. Если `a` равна имени, то вывести «Привет, имя!», если `a` равна пустому значению, то вывести «Привет, незнакомец!»

## ЛОГИЧЕСКИЕ КОНСТРУКЦИИ И ИЛИ И НЕ В УСЛОВНОМ ОПЕРАТОРЕ

1. Иногда необходимо предусмотреть выполнение одновременно нескольких условий. Тогда условия соединяются **логическим оператором И** — `&&`:

```
1 $a=1;
2 if ($a>0 && $a<5) {
3     echo "a > 0 и a < 5";
4 }
```

2. При необходимости выполнения одного из нескольких условий применяется **логический оператор ИЛИ** — `||`:

```
1 $a=1;
2 if ($a>0 || $a>1) {
3     echo "a > 0 или a > 1";
4 }
```

3. Для указания ложности условия используется **логический оператор НЕ** — `!`:

```
1 $a=1;
2 if (!(a<0)) {
3     echo "a не < 0";
4 }
```

## Оператор Switch PHP

Оператор `switch` или «переключатель» заменяет несколько подряд идущих конструкций `if`. При этом он сравнивает одну переменную с множеством значений. Таким образом, это наиболее удобное средство для организации мультиветвления.

### Синтаксис:

```
1 switch($переменная) {
2   case "значение1": оператор1;
3   break;
4   case "значение2": оператор2;
5   break;
6   case "значение3": оператор3;
7   break;
8   [default: оператор4;
9   break;]
10 }
```

- Оператор может проверять как строковые значения (тогда они указываются в кавычках), так и числовые (без кавычек).
- Оператор `break` в конструкции обязателен. Он осуществляет выход из конструкции при истинном условии и выполнении соответствующего условию оператора. Без `break` будут выполняться операторы всех `case` независимо от их истинности.

```
$a=5;
switch($a) {
case 1: echo "a=1";
break;
case 2: echo "a=2";
break;
case 3: echo "a=3";
break;
default: echo "a не равно 1, 2, 3";
break;
}
```

Рис. 3.5. Условный оператор Switch



**Пример:** задан массив с полными мужскими именами. Проверять первый элемент массива и в зависимости от имени, выдавать приветствие с коротким именем.

### Решение:

```
1 $names=array("Иван", "Петр", "Семен");
2 switch($names[0]){
3   case "Петр":
4     echo "Привет, Петя!";
5     break;
6   case "Иван":
7     echo "Привет, Ваня!";
8     break;
9   case "Семен":
10    echo "Привет, Ваня!";
11    break;
12  default:
13    echo "Привет, $names[0]!";
14    break;
15 }
```



#### Задание php 3\_4:

- Создайте переменную `$day` и присвойте ей произвольное числовое значение
- С помощью конструкции `switch` выведите фразу «Это рабочий день», если значение переменной `$day` попадает в диапазон чисел от 1 до 5 (включительно)
- Выведите фразу «Это выходной день», если значение переменной `$day` равно числам 6 или 7
- Выведите фразу «Неизвестный день», если значение переменной `$day` не попадает в диапазон чисел от 1 до 7(включительно)

Дополните код:

```
1 ...  
2 switch(...){  
3 case 1:  
4 case 2:  
5 ...  
6         echo "Это рабочий день";  
7         break;  
8 case 6:  
9 ...  
10 default:  
11 ...  
12 }
```

## Тернарный оператор PHP

Тернарный оператор, т.е. с тремя операндами, имеет достаточно простой синтаксис, в котором слева от знака `?` записывается условие, а справа — два оператора разделенные знаком `:`, слева от знака выполняется оператор при истинности условия, а справа от знака `:` выполняется оператор при ложном условии.



```
условие ? оператор1 : оператор2;
```



**Пример:** если значение переменной `speed` меньше 60, то выводить сообщение «Показания скорости в норме», иначе выводить сообщение «Скорость превышена!»

```
1 <?php  
2 $speed = 45;  
3 echo ($speed < 60) ? "Показания скорости в норме" : "Скорость превышена!";  
4 ?>
```



**Задание php 3\_5:** выполнить предыдущее задание (php 3\_4), используя тернарный оператор и логические операторы

## Задача 5: Работа с массивами

### PHP: объявление массива

#### 1. Массивы с числовыми индексами

```
1 $fruits[0]="яблоко";
2 $fruits[1]="груша";
3 $fruits[2]="апельсин";
4 $fruits[3]="абрикос";
```

Данный вариант инициализации массива сегодня практически не используется. Используется следующий вариант работы с массивами:

#### 2. Ассоциативный массив

Данный вариант создания массива использует объект `array`:

```
1 $fruits = array(
2     1 => "яблоко",
3     2 => "груша",
4     3 => "апельсин",
5     4 => "абрикос"
6 );
7
8 // Начиная с версии PHP 5.4
9 $fruits = array(
10     1 => "яблоко",
11     2 => "груша",
12     3 => "апельсин",
13     4 => "абрикос"
14 );
```

### Доступ к элементам массива

```
1 $fruits[0]="яблоко";
2 $fruits[1]="груша";
3 $fruits[2]="апельсин";
4 $fruits[3]="абрикос";
5
6 echo "Первый элемент массива - ". $fruits[0] . "<br>";
7 echo "Второй элемент массива - ". $fruits[1] . "<br>";
8 echo "Третий элемент массива - ". $fruits[2] . "<br>";
```

Результатом выполнения программы будет:

```
Первый элемент массива - яблоко
Второй элемент массива - груша
Третий элемент массива - апельсин
```



**Важно:** В качестве индекса элемента в ассоциативном массиве может быть не число, а слово (тип `string`)

Рассмотрим пример:



**Пример:** Создать массив `student` со значениями имен, фамилий и возраста студентов. Вывести на экран значения элементов массива

```
1 $student = array(
2     'Name' => John,
3     'Surname' => Smith,
4     'Age' => 20
5 );
6 echo "имя пользователя " . $student['Name'] . "<br>";
7 echo "фамилия пользователя " . $student['Surname'] . "<br>";
8 echo "возраст пользователя " . $student['Age'] . "<br>";
```

или

```
1 $student['Name'] = 'John';
2 $student['Surname'] = 'Smith';
3 $student['Age'] = 20;
4 echo "имя пользователя " . $student['Name'] . "<br>";
5 echo "фамилия пользователя " . $student['Surname'] . "<br>";
6 echo "возраст пользователя " . $student['Age'] . "<br>";
```

Результат:

```
имя пользователя John
фамилия пользователя Smith
возраст пользователя 20
```



**Важно:** Строковые ключи (индексы) нужно всегда заключать в кавычки



#### Индексированные массивы без ключа:

Также следует учесть, что индексы в массиве вообще можно не писать при инициализации массива значениями. Тогда мы создаем так называемый массив-коллекцию (объект-коллекцию):

```
$student = array('John', 'Smith', 20);
```

В таком случае интерпретатор сам присвоит им числовые индексы, начиная с 0



#### Пример преобразования типов и перезаписи элементов.

Обратите внимание на использование процедуры `var_dump()`

```
1 <?php
2 $my_array = array(
3     0 => "a",
4     "1" => "b",
5     2 => "c",
6     2 => "d",
7 );
8 var_dump($my_array);
9 ?>
```

Результат:

```
array(3)
{
    [0]=> string(1) "a"
    [1]=> string(1) "b"
    [2]=> string(1) "d"
}
```



#### Пример использования многомерного массива и организации доступа к его элементам

```
1 <?php
2 $array_odnom = array(
3     "first" => "1",
4     2 => 2,
5     "multi" => array(
6         "1_1" => "1_1"
7     )
8 );
9 var_dump($array_odnom ["first"]);
10 var_dump($array_odnom [2]);
11 var_dump($array_odnom ["multi"] ["1_1"]);
12 ?>
```

Результат:

```
string(1) "1"
int(2)
string(3) "1_1"
```



**Пример:** создать двумерный массив размерностью 3 x 2. Заполнить его значениями для первой строки: «1\_1», «1\_2», «1\_3»; для второй строки: «2\_1», «2\_2», «2\_3». Вывести на экран первый элемент массива



#### Выполнение:

```
1 $array_odnom = array(  
2     1 => array(  
3         1 => "1_1",  
4         2 => "1_2",  
5         3 => "1_3"),  
6     2 => array(  
7         1 => "2_1",  
8         2 => "2_2",  
9         3 => "2_3"),  
10 );  
11 );  
12 echo $array_odnom[1][1];
```



**Задание php2\_1:** создайте двумерный массив 3 x 3 — верхний левый угол таблицы умножения Пифагора(1, 2, 3; 4, 6, 8 ...)



#### Лабораторная работа:

Допустим, у Вас есть какое-то объявление и несколько разных людей, которым нужно это объявление отправить. Для этого Вы делаете заготовку с содержанием объявления, внутри которого есть ряд изменяющихся параметров: массива имен людей и массив событий. Вывести один вариант итогового объявления на экран. Для задания массивов использовать и ассоциативный массив.

**Используйте шаблон объявления и ориентируйтесь по цветам:**

Красным – массивы.

Коричневым – числовая переменная.

Синим – константа.

Уважаемый (ая), *Иван Иванович!*

Приглашаем Вас на *день открытых дверей*.

Дата события: *12 мая*.

С уважением, *Василий*.

#### Дополните код:

```
1 // объявление константы  
2 define("SIGN", "С уважением, Вася");  
3 // массив для имен адресатов  
4 $names = array(...);  
5 // массив для событий  
6 $events = array(  
7     "op_doors" => "день открытых дверей",  
8     "vistavka" => "выставку",  
9     ...  
10 );  
11 $str = "Уважаемый(ая), $names[0]! <br>";  
12 $str .= ...;  
13 $str .= ...;  
14 echo ...;
```

#### Критерии оценки за практическую работу:

1. содержание Web-страницы соответствует заданию
2. текст на Web-странице не содержит ошибок
3. прописано название Web-страницы
4. правильный html-код
5. правильное название файла
6. правильное расположение файла на Личном диске студента

<b>5</b>	Пункты 1-6 верны
<b>4</b>	Один или два пункта из 1-4 содержат неточности. Пункты 5,6 верны
<b>3</b>	Три и более пунктов из 1-4 содержат неточности. Пункты 5,6 верны
<b>2</b>	Пункты 5,6 не верны или Пункты 1-4 не верны

## Список источников и литературы

Основные источники:

1. Сутуга, О. Н., Многостраничный дизайн. Практикум : учебное пособие / О. Н. Сутуга. — Москва : КноРус, 2023. — 245 с. — ISBN 978-5-406-12194-8. — URL: <https://book.ru/book/950617>
2. Математика с MathCad, Власова А.М., 2017 - <https://nashol.com/2018012398588/matematika-s-mathcad-vlasova-a-m-2017.html>

Электронные издания:

1. <http://labs.org.ru/javascript-2/>
2. [http://www.webpupil.ru/javascript\\_step\\_view.php?id=5](http://www.webpupil.ru/javascript_step_view.php?id=5)
3. <http://www.websovet.com/javascript-dlya-nachinayushhix-vash-pervyj-skript>
4. <https://myrusakov.ru/javascript-skripty.html>
5. [http://uroki-javascript.ru/js/js\\_examples.php](http://uroki-javascript.ru/js/js_examples.php)