



**Министерство науки и высшего образования
Российской Федерации
Рубцовский индустриальный институт (филиал)
ФГБОУ ВО «Алтайский государственный технический
университет им. И.И. Ползунова»**

И.Б. ШУЛЬМАН

ОСНОВЫ WEB- ТЕХНОЛОГИЙ

Методические указания для студентов очной формы обучения
направления «Информатика и вычислительная техника»

Рубцовск 2019

УДК 004.77

Шульман И.Б. Основы Web- технологии: Методические указания для студентов очной формы обучения направления 09.03.01 Информатика и вычислительная техника / Рубцовский индустриальный институт. – Рубцовск, 2019. – 62 с.

Методические указания содержат материал для организации учебных занятий и самостоятельной работы студентов очной формы по предмету «Основы Web- технологий». Данное пособие включает в себя описание лабораторных работ по основным разделам дисциплины, методические рекомендации по их выполнению. Настоящие указания предназначены для студентов направления Информатика и вычислительная техника технического факультета.

Рассмотрены и одобрены
на заседании научно-
методического совета РИИ
Протокол № 10 14.05.2019

Рецензент: Начальник ИТО

А.Н. Цыганков

© Рубцовский индустриальный институт, 2019

Оглавление

Лабораторная работа № 1	4
Лабораторная работа № 2	6
Лабораторная работа №3	8
Лабораторная работа №4	12
Лабораторная работа № 5	16
Лабораторная работа № 6	20
Лабораторная работа №7	24
Лабораторная работа № 8	33
Лабораторная работа № 9	44
СПИСОК ЛИТЕРАТУРЫ.....	55

Лабораторная работа № 1

Идея взята с сайта <https://sites.google.com/site/inftech11/home/homub/blog>

1. Зарегистрируйтесь на сайте google.com и создайте почтовый ящик вида Имя.Фамилия.Группа@gmail.com, например: Maria.Ivanova.IVT11@gmail.com. 15

2. Создайте блог, в котором вы будете опубликовывать свои отчеты о выполненных заданиях. Отключите проверку CAPTCHA при публикации комментариев (см. видео). Настройте конфиденциальность (см. п. II) 15+5=20

3. Прочитайте краткий обзор облачных сервисов хранения информации и зарегистрируйтесь на одном из них.

4. В качестве отчета о выполненной работе:

○ добавьте в друзья мой аккаунт - teacher.web.2014@gmail.com (для удобства публикации в дальнейшем); 10

○ загрузите текстовый файл (с первыми впечатлениями и трудностями в этой работе) в облачное хранилище (задание №3); 10

○ опубликуйте в блоге первое сообщение, в котором напишите приветствие преподавателю по дисциплине "Web-технологии" и вставьте вашу фотографию; Не забудьте поделиться им с преподавателем. 5+5=10

○ опубликуйте во втором сообщении ссылку на файл с впечатлениями и оставьте краткий комментарий о выполненной работе. 10

5. Заполните регистрационную форму: [Регистрация](#) 10

Проверить правильность введения [Итоговая таблица](#)

Если обнаружите ошибки при заполнении формы, то заполните ее вновь, а старый вариант я удалю.

Дополнительные материалы:



I. [Инструкция по созданию аккаунта Google](#)



[Создание аккаунта Google](#) (видео без звука! Следите за указателем мыши.)



II. [Создание блога \(видео\)](#)

(Сервисы blogger; шаблон Простой)

Зайдите в настройки своего блога (**Настройки - Основное**) и в поле **Конфиденциальность** отключите поиск вашего блога в системе Blogger и поисковыми системами.

Конфиденциальность

Добавить блог в наши каталоги? ?
 Да Нет

Вы хотите, чтобы поисковые системы находили ваш блог?
 Да Нет

[Сохранить изменения](#) [Отмена](#)



[Обзор облачных сервисов хранения файлов](#)



[Инструкция по созданию аккаунта Dropbox](#)

[Переход к форме регистрации Dropbox](#)

[Установка Dropbox и получение ссылки для блога](#)



[Загрузка файлов в Dropbox через браузер](#)

[Загрузка файлов в Яндекс.Диск](#)

[Загрузка файлов в Google Drive](#)

[What is Dropbox?](#)

[iCloud Harmony TV Ad](#)

[Руководство к выполнению Лабораторной работы](#)

Лабораторная работа № 2.

Аналитический обзор возможностей основных online графических редакторов

Идея взята с сайта <https://sites.google.com/site/inftech11/home/homub/for/----online>

1. Выберите онлайн-графический редактор (самостоятельно или из предложенного списка), например:

SumoPaint.com, Pixlr, FanStudio, MyPicture Resize, Editor.pho.to, FotoFlexer, Picnic, Lunapic, Photoshop on-line, SplashUp.com и др.

2. Подготовьте аналитический обзор выбранного ресурса, по следующему плану:

- название, url
- история создания (кем, когда, для каких целей)
- назначения
- основные функциональные возможности
- необходимость регистрации
- наличие справочного материала
- простота использования
- какие форматы поддерживает
- возможность сохранения созданного изображения

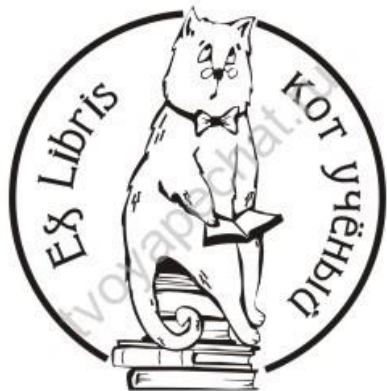
3. Разместите аналитический обзор в блоге. Не забудьте поделиться им с преподавателем.

4. Создайте логотип в выбранном редакторе. Творчески оформленный логотип должен содержать **ваши инициалы на прозрачном фоне**. Для этого вам необходимо сохранить итоговое изображение в формате .gif или .png. Этот логотип должен присутствовать в любом графическом изображении (или другом документе), который вы прикрепляете в блог в качестве отчета по лабораторной работе.

Примеры логотипов и их размещение на изображении:



Некоторые примеры экслибрисов (авторский знак на книги):



5. Поместите в блоге свой логотип и краткое описание выполненных действий по его созданию.

Лабораторная работа №3

(в журнале JS 1)

Основные положения

Программа (сценарий) на языке JavaScript представляет собой последовательность операторов с "точкой с запятой" (;) между ними. Если каждый оператор размещается на одной строке, то разделитель можно не писать. Один оператор может располагаться на нескольких строках.

В программах на JavaScript можно использовать комментарии. Для того чтобы задать комментарий, располагающийся на одной строке, достаточно перед его текстом поставить две косые черты (//). Если же поясняющий текст занимает несколько строк, то его следует заключать между символами /* и */. В JavaScript строчные и прописные буквы алфавита считаются разными символами. Любой язык программирования оперирует с постоянными и переменными величинами. В JavaScript это литералы и переменные.

Опр.: Простейшие данные, с которыми может оперировать программа, называются литералами.

Литералы не могут изменяться. Литералы целого типа могут быть заданы в десятичном (по основанию 10), шестнадцатеричном (по основанию 16) или восьмеричном (по основанию 8) представлении. Шестнадцатеричные числа включают цифры 0-9 и буквы *a, b, c, d, e, f*. Шестнадцатеричные числа записываются с символами *0x* перед числом, например, *0x25, 0xa1, 0xff*. Запись вещественного литерала отличается от записи вещественного числа в математике тем, что вместо запятой, отделяющей целую часть от дробной, указывается точка, например, 123.34, -22.56. Кроме того, для записи вещественных чисел можно использовать так называемую экспоненциальную форму.

Кроме целых и вещественных значений в языке JavaScript могут встречаться так называемые логические значения. Существуют только два логических значения: истина и ложь. Первое представляется литералом *true*, второе - *false*. В некоторых реализациях JavaScript может быть использована единица в качестве *true*, и ноль в качестве *false*. Строковый литерал представляется последовательностью символов, заключенной в одинарные или двойные кавычки. Примером строкового литерала может быть строка *"результат"* или *'результат'*.

Опр.: Элемент, используемый для хранения данных, называется переменной. Тип переменной зависит от хранимых в ней данных, при изменении типа данных меняется тип переменной. Определить переменную можно с помощью оператора *var*, например: *var test1*.

В данном случае тип переменной *test1* не определен и станет известен только после присвоения переменной некоторого значения. Оператор *var* можно использовать и для инициализации переменной, например, конструкцией *var test2=276* определяется переменная *test2* и ей присваивается значение 276.

Значение переменной изменяется в результате выполнения оператора присваивания. Оператор присваивания может быть использован в любом месте программы и способен изменить не только значение, но и тип переменной. Оператор присваивания выглядит так: *a=b*, где *a* - переменная, которой мы хотим

задать некоторое значение; *b* - выражение, определяющее новое значение переменной.

Пусть в сценарии описаны следующие переменные

```
var n=3725
var x=2.75
var p=true
var s="Выполнение завершено"
```

n и *x* имеют тип *number*, тип переменной *p* - логический, переменная *s* имеет тип *string*. В JavaScript определен тип *function* для всех стандартных функций и функций, определяемых пользователем. Объекты JavaScript имеют тип данных *object*. Переменные типа *object* часто называют просто объектами, они могут хранить объекты.

Опр.: Выражения строятся из литералов, переменных, знаков операций, скобок. В зависимости от типа вычисленного значения выражения можно разделить на арифметические, логические и строковые. Арифметические выражения получаются при выполнении операций, перечисленных в [табл.1.1](#).

Операции отношения применимы к операндам любого типа. Результат операции - логическое значение *true*, если сравнение верно, и *false* - в противном случае.

Приоритет операций определяет порядок, в котором выполняются операции в выражении. В [табл.1.4](#) перечислены рассмотренные операции в порядке убывания приоритетов.

Сценарии, написанные на языке JavaScript, могут располагаться непосредственно в HTML-документе между тегами `<script>` и `</script>`.

Одним из параметров тега `<script>` является *language*, который определяет используемый язык сценариев. Для языка JavaScript значение параметра равно "JavaScript". Если применяется язык сценариев VBScript, то значение параметра должно быть равным "VBScript". В случае использования языка JavaScript параметр *language* можно опускать, т. к. этот язык выбирается браузером по умолчанию.

Обычно браузеры, не поддерживающие какие-либо теги HTML, эти теги просто игнорируют. Попытка браузера проанализировать содержимое не поддерживаемых тегов может привести к неверному отображению страницы. Чтобы избежать такой ситуации, рекомендуется помещать операторы языка JavaScript в теги комментария `<!-- ... -->`. Для правильной работы интерпретатора перед закрывающим тегом комментария `-->` следует поставить символы `//`.

Итак, для размещения сценария в HTML-документе следует написать следующее:

```
<script language="JavaScript">

</script>
```

Документ может содержать несколько тегов `<script>`. Все они последовательно обрабатываются интерпретатором JavaScript. В следующем примере в раздел `<body>` (в тело) HTML-документа вставлены операторы языка JavaScript.

Пример 1. Вычисление площади треугольника

Необходимо написать сценарий, определяющий площадь прямоугольного треугольника по заданным катетам. Сценарий разместим в разделе <body> HTML-документа (листинг 1).

Листинг 1. Первый сценарий в документе :

```
<HTML>
<HEAD>
<title>Первый сценарий в документе</title>
</HEAD>
<BODY>
<P>Страница, содержащая сценарий.</P>
<script>
<!--
var a=8; h=10 /*Инициализируются две переменные*/
document.write ("Площадь прямоугольного треугольника равна ", a*h/2, ".") /*Для формирования вывода
используется метод write объекта document*/
//-->
</script>
<P>Конец формирования страницы, содержащей сценарий</P>

</body>
</html>
```

Задания

1. Проверить пример из лабораторной работы. (5)
2. Составить сценарий, в котором вычисляется: (номер задачи выбирать согласно Вашему номеру в списке группы, 1 вариант у 1, 6, 11, 16, 21, 26 студентов и т.д.) (10)
 - 2.1. площадь круга по заданному радиусу.
 - 2.2. гипотенуза по заданным катетам.
 - 2.3. длина окружности по заданному радиусу ...
 - 2.4. радиус по заданной площади круга
 - 2.5. расстояние между двумя точками на плоскости, заданными своими координатами
- 2.n площадь треугольника по формуле Геррона (задание для отсутствующих на паре)
3. Обобщить задачу, позволив пользователю вводить значения переменных. (с помощью метода prompt или через форму) (11)
4. Изучить статью [Язык сценариев JavaScript](#) . Проверить и проанализировать примеры приведенных программ: Листинг 1.3., Листинг 1.4., Листинг 1.5., Листинг 1.6., Листинг 1.7., Листинг 1.8, Листинг 1.9, Листинг 1.10, Листинг 1.11, Листинг 1.12. , Листинг 1.15. . (55)
(в текстах программ встречается ошибки! иногда 1 заменяется на l (L) например numl вместо num1)

Если ссылки не работают: Арифметические выражения получаются при выполнении операций; операции в порядке убывания приоритетов:

<http://javascript.ru/arithmic>

<http://www.snkey.net/books/webmaster/javascript-operations.html#p3>

!!!Д/З: Вспомнить /ознакомиться с тегами Html: form, input, Input(type)

Лабораторная работа №4

(в журнале JS 2)

Функция и обработка события

Основным элементом языка JavaScript является функция. Описание функции имеет вид

```
function F (V) {S},
```

где F - идентификатор функции, задающий имя, по которому можно обращаться к функции; V - список параметров функции, разделяемых запятыми; S - тело функции, в нем задаются действия, которые нужно выполнить, чтобы получить результат. Необязательный оператор *return* определяет возвращаемое функцией значение. Обычно все определения и функции задаются в разделе `<head>` документа. Это обеспечивает интерпретацию и сохранение в памяти всех функций при загрузке документа в браузер.

Пример 1. Нахождение площади треугольника.

В предыдущих примерах пользователю не предоставлялась возможность вводить значения, и в зависимости от них получать результат. Интерактивные документы можно создавать, используя формы. Предположим, что мы хотим создать форму, в которой поля Основание и Высота служат для ввода соответствующих значений. Кроме того, в форме создадим кнопку Вычислить. При щелчке мышью по этой кнопке мы хотим получить значение площади треугольника. Действие пользователя (например, щелчок кнопкой мыши) вызывает событие. События в основном связаны с действиями, производимыми пользователем с элементами форм HTML. Обычно перехват и обработка события задается в параметрах элементов форм. Имя параметра обработки события начинается с приставки *on*, за которой следует имя самого события. Например, параметр обработки события *click* будет выглядеть как *onclick*.

Листинг 1. Реакция на событие Click.

```
<HTML>
<HEAD>
<title>Обработка значений из формы</title>
<script language="JavaScript">
<!--//
function care (a, h)
{
var s=(a*h)/2;
document.write ("Площадь прямоугольного треугольника равна ",s);
return s
}
//-->
</script>
</HEAD>
<BODY>
<P>Пример сценария со значениями из формы</P>
<FORM name="form1">
Основание: <input type="text" size=5 name="st1"><hr>
Высота: <input type="text" size=5 name="st2"><hr>
<input type="button" value=Вычислить
onClick="care(document.form1.st1.value, document.form1.st2.value)"> /*По клику мыши на кнопке в
функцию care передаются два параметра - содержимое полей ввода*/
</FORM>

</body>
</html>
```

При интерпретации HTML-страницы браузером создаются объекты JavaScript. Взаимосвязь объектов между собой представляет иерархическую структуру. На самом верхнем уровне иерархии находится объект *windows*, представляющий окно браузера. Объект *windows* является предком или "родителем" всех остальных объектов. Каждая страница кроме объекта *windows* имеет объект *document*. Свойства объекта *document* определяются содержанием самого документа: цвет фона, цвет шрифта и т. д. Для получения значения основания треугольника, введенного в первом поле формы, должна быть выполнена конструкция

```
document.form1.st1.value
```

т.е., говоря русским языком (при этом читаем с конца), используем данные *value* из поля ввода с именем *st1* находящегося на форме *form1* объекта *document*.

Пример 2. Вычисление площади квадрата.

Напишем сценарий, определяющий площадь квадрата по заданной стороне. Площадь должна вычисляться в тот момент, когда изменилось значение его стороны. Пусть форма содержит два текстовых поля: одно для длины стороны квадрата, другое для вычисленной площади. Кнопка Обновить очищает поля формы. Площадь квадрата вычисляется при возникновении события *change*, которое происходит в тот момент, когда значение элемента формы с именем *num1* изменилось, и элемент потерял фокус. HTML-код представлен в примере 2.

Листинг 2. Реакция на событие Change

```
<HTML>
<HEAD>
<title>Обработка события Change - изменение значения элемента</title>
<script>
function srec(obj)
{obj.res.value=obj.num1.value* obj.num1.value}
</script>
</HEAD>
<BODY>
<P>Вычисление площади квадрата</P>
<FORM name="form1">
Сторона: <input type="text" size=7 name="num1"
onChange="srec(form1)">
<hr>
Площадь: <input type="text" size=7 name="res"><hr>
<input type="reset" value=Обновить>
</FORM>

</body>
</HTML>
```

Событие *Focus* возникает в момент, когда пользователь переходит к элементу формы с помощью клавиши <Tab> или щелчка мыши. Событие "потеря фокуса" (*Blur*) происходит в тот момент, когда элемент формы теряет фокус. Событие *select* вызывается выбором части или всего текста в текстовом поле. Например, щелкнув дважды мышью по полю, мы выделим поле, наступит событие *select*, обработка которого приведет к вычислению требуемого значения. В [табл.3](#) представлены события и элементы документов HTML, в которых эти события

могут происходить. <http://learn.javascript.ru/focus-blur> , <http://webgyry.info/focus-javascript>

В языке JavaScript определены некоторые стандартные объекты и функции, пользоваться которыми можно без предварительного описания. Одним из стандартных объектов является объект *Math*. В свойствах упомянутого объекта хранятся основные математические константы, а его методы можно использовать для вызова основных математических функций. В [табл.4](#) приведены некоторые методы объекта *Math*. Выражение $y = \log x$ запишется $y = \text{Math.log}(x)$.

<http://www.site-do.ru/js/js6.php>

<http://javascript.ru/Math>

<http://myrusakov.ru/math-javascript.html>

Задания

1. Проверить примеры из лабораторной работы. (10)
2. Использовать событие *Focus*, т.е. изменять цвет фона и рамки активного элемента формы (номер задачи выбирать согласно Вашему номеру в списке группы, 1 вариант у 1, 6, 11, 16, 21, 26 студентов и т.д.) (15)
 - 2.1. На плоскости заданы координаты трех точек. Напишите сценарий, который вычисляет площадь треугольника.
 - 2.2. Вводится информация о доходах сотрудника за первый квартал текущего года. Требуется определить:
 - общую сумму дохода за квартал;
 - сумму подоходного налога (13%);
 - сумму, подлежащую выдаче на руки.
 - 2.3. Напишите сценарий, который определяет объем шара и площадь боковой поверхности, если известен радиус.
 - 2.4. Задан радиус окружности. Определите длину окружности и площадь соответствующего круга.
 - 2.5. Задана окружность (координатами центра и радиусом) и точка вне окружности. Определите длину касательной из заданной точки к окружности.
 - 2.n. Определите расстояние между двумя точками на плоскости, заданными своими координатами. (задание для отсутствующих на паре)
3. Использовать событие *Select* (15)
 - 3.1. Напишите сценарий, который для точки, заданной координатами на плоскости, определяет расстояние до начала координат (для нечетных номеров в списке группы).
 - 3.2. Напишите сценарий, который определяет площадь кольца по двум заданным радиусам (для четных номеров в списке группы)
4. Напишите сценарий, который обменивает местами значения двух введенных переменных (использовать событие *Blur*). (10)
5. Измените задачу из 1 лаб/раб (№2), используя форму для ввода значений, и любой из методов (*Change*, *Focus*, *Select*, *Blur*) (10)
6. Создать небольшой сайт: на главной странице тексты задач и ссылки на их выполнение. (30)

Все реализованные сценарии разместите в блоге (ссылку на архив), предварительно добавив логотип, в каждый файл (Web –страницу)

Если ссылки не работают: события и элементы документов HTML:

<http://www.puzzleweb.ru/html/eventattributes.php>

некоторые методы объекта *Math*.:

<http://javascript.ru/math>

Лабораторная работа № 5

(в журнале JS 3)

Организация ветвлений в программах

При составлении программы часто необходимо выполнение различных действий в зависимости от результатов проверки некоторых условий. Для организации ветвлений можно воспользоваться условным оператором, который имеет вид:

```
if B {S1} else {S2}
```

где B - выражение логического типа; $S1$ и $S2$ - операторы. Выполнение условного оператора осуществляется следующим образом. Вычисляется значение выражения B . Если оно истинно, то выполняются операторы $S1$, если ложно - операторы $S2$. Если последовательность операторов $S1$ или $S2$ состоит лишь из одного оператора, то фигурные скобки можно опустить. Возможна сокращенная форма условного оператора:

```
if B {S}
```

где B - выражение логического типа; S - последовательность операторов. Выполнение краткого условного оператора осуществляется так: вычисляется значение выражения B , если оно истинно, то выполняются операторы S .

Пример 1. Нахождение максимального значения

Для трех заданных значений a , b , c необходимо написать сценарий, определяющий максимальное значение. Поступим следующим образом. Сначала максимальным значением m будем считать значение a , далее значение b сравним с максимальным. Если окажется, что b больше m , то максимальным становится b . И, наконец, значение c сравнивается с максимальным значением из предыдущих значений a и b . Если c больше m , то максимальным становится c . Оператор присваивания

```
obj.res.value=m
```

обеспечивает запись вычисленного максимального значения в соответствующее поле формы. Функция *Number* (s) преобразует объект s , заданный в качестве параметра, в число. Полностью сценарий может быть записан так, как представлено в листинге 1.

Листинг 1. Вычисление максимального значения из трех заданных

```
<HTML>
<HEAD>
<TITLE>Вычисление максимального значения</TITLE>
<script language="JavaScript">
<!-- //
function maxval (obj )
{
var a = Number(obj.num1.value);
var b = Number(obj.num2.value);
var c = Number(obj.num3.value);
var m=a
if (b > m) m=b
if (c > m) m=c
obj.res.value=m }
//-->
</script>
</HEAD>
<BODY>
<H4>Вычисление максимального значения</H4>
<FORM name="form1">
```



```

Число 1: <input type="text" size=8 name="num1"><hr>
Число 2: <input type="text" size=8 name="num2"><hr>
Число 3: <input type="text" size=8 name="num3"><hr>
Максимальное значение равно
<input type="button" value=Определить onClick="maxval(form1)">
<input type="text" size=8 name="res"><hr>
<input type="reset">
</FORM>

```

```

</body>
</html>

```

Решим рассмотренную задачу другим способом. Вспомним, что стандартный объект *Math* имеет метод *max*, который определяет наибольшее значение двух аргументов. Опишем функцию *maxval1*, которая определяет максимальное значение из трех заданных значений и использует объект *Math*.

```

function maxval1 (obj )
{
  var a = Number(obj.num1.value);
  var b = Number(obj.num2.value);
  var c = Number(obj.num3.value);
  obj.res.value=Math.max(Math.max(a,b),c)
}

```

Если бы требовалось определить максимальное из четырех заданных значений *a, b, c, d*, то можно было бы воспользоваться формулой

```
Math.max(Math.max(a,b), Math.max(c,d)).
```

Задания

1. Проверьте примеры из лабораторной работы, попробуйте и второй вариант, используйте объект *Math*. (10)
2. Напишите программу, которая определяет, попадает ли точка на плоскости с координатами *X* и *Y* в заштрихованную область (рис. 2.1 - 2.6, номер задачи выбирать согласно Вашему номеру в списке группы, 1 вариант у 1, 6, 11, 16, 21, 26 студентов и т.д.). (15)

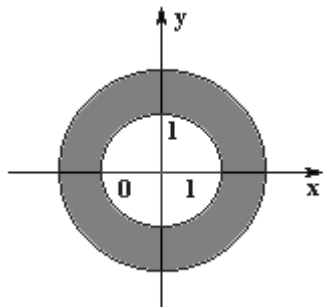


Рис 2.1. Точка попадает в область, образованную исключением двух кругов

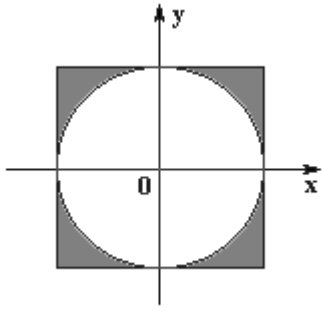


Рис 2.2. Точка попадает в область, образованную исключением квадрата и круга

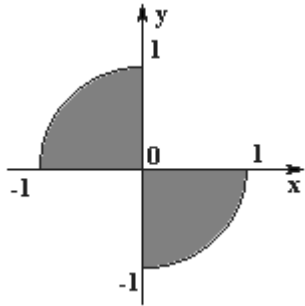


Рис 2.3. Точка попадает в область, образованную двумя секторами

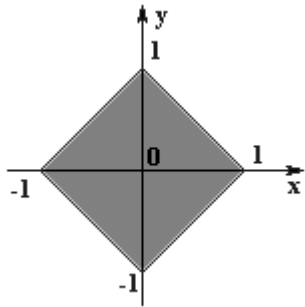


Рис 2.4. Точка попадает в область, образованную ромбом

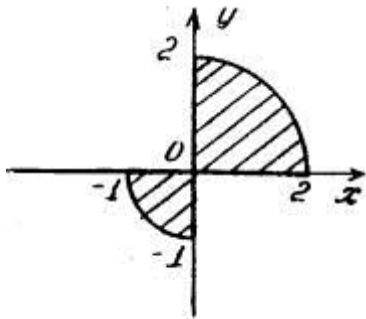


Рис 2.5. Точка попадает в область, образованную двумя секторами

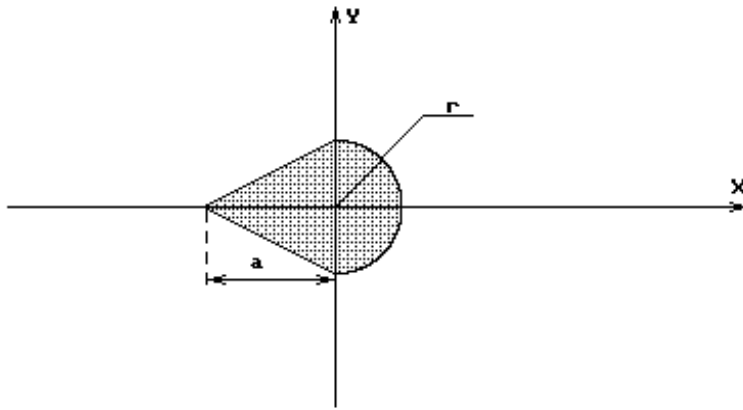


Рис 2.6. Точка попадает в область, образованную объединением треугольника и полукруга.
(задание для отсутствующих на паре)

3. Вводится последовательность из пяти чисел. Напишите сценарий, в котором определяется количество максимальных элементов, и их значение. (20)
4. Напишите программу, которая определяет, можно ли построить треугольник с заданными длинами сторон. (15)
5. Точка на плоскости задается своими координатами. Определите, какой из четвертей прямоугольной системы координат принадлежит заданная точка. (15)
6. Создать небольшой сайт – **Лаб_раб3_ФИО**: на главной странице тексты задач и ссылки на их выполнение. (25)
7. Все реализованные сценарии разместите в блоге (ссылку на архив), предварительно добавив логотип, в каждый файл (Web –страницу).

Лабораторная работа № 6

(в журнале JS 4)

Методы в JavaScript

Во время интерпретации HTML-документа браузером создаются объекты JavaScript. Свойства объектов определяются параметрами тегов языка HTML. Структура документа отражается в иерархической структуре объектов, соответствующих HTML-тегам. Родителем всех объектов является объект *windows*, расположенный на самом верхнем уровне иерархии, он представляет окно браузера и создается при запуске браузера. Для того чтобы открыть новое окно в сценарии JavaScript и отобразить в нем новый документ, применяется метод *open*, для закрытия окна можно воспользоваться методом *close*. Метод *alert* объекта *windows* отображает диалоговое окно с текстом, переданным методу в качестве параметра. Данный метод используется в случаях проверки правильности вводимых данных с помощью формы. Свойства объекта *windows* относятся ко всему окну, в котором отображается документ.

Подчиненными объектами (или объектами нижнего уровня) являются объекты *document*, *history*, *location*, *frame*. Свойства объекта *history* представляют адреса ранее загружаемых HTML-страниц. Свойства объекта *location* связаны с URL-адресом отображаемого документа, объекта *frame* - со специальным способом представления данных.

Свойства объекта *document* определяются содержимым самого документа: шрифт, цвет фона, формы, изображения и т. д. Объект *document* в зависимости от своего содержимого может иметь объекты, являющиеся для него подчиненными или дочерними. В частности подчиненными для объекта *document* являются объекты *form*, *image*, *link*, *area* и др. Иерархическая структура объектов представлена на [рис.1](#)

Для каждой страницы создается один объект *document*, некоторые его свойства соответствуют параметрам тега `<BODY>`: *bgColor*, *fgcolor*, *linkcolor*, *alinkcolor*, *vlinkColor*. Методы *write* и *writeln* записывают в документ текст, задаваемый параметром.

Если документ содержит изображения, то доступ к объекту, определяющему изображение, можно получить с помощью переменной, указанной в параметре *name* тега ``. Объект *image* имеет свойство *images*, которое содержит ссылки на все изображения, расположенные в документе. Ссылки перенумерованы, начиная с нуля. Доступ к первому изображению можно получить с помощью составной конструкции *document.images[0]*, ко второму - *document.images[1]*. Если на странице пять изображений, то доступ к последнему изображению можно получить, воспользовавшись ссылкой *document.images[4]*.

Рассмотрим примеры, в которых используются различные свойства объектов.

Для встраивания изображений в HTML-документ служит тег ``, имеющий обязательный параметр *src*, определяющий URL-адрес файла с изображением. Можно задавать размеры выводимого изображения. Значение параметра *width* определяет ширину изображения, значение параметра *height* - высоту изображения. Значения параметров ширины и высоты могут не совпадать с истинными размерами изображений, тогда при загрузке изображения автоматически выполняется перемасштабирование.

Изображение можно поместить в рамку. Для этого используется параметр *border*. Значением параметра должно быть число, определяющее толщину рамки в пикселях. По умолчанию рамка вокруг изображения отсутствует, если только изображение не является ссылкой.

Параметр *alt* определяет альтернативный текст. При наведении курсора мыши на изображение появляется комментарий.

Пример 1. Перестановка изображений

Напишем сценарий, который реализует обмен рисунков в документе. Пусть в документе расположено три изображения, пронумерованных от 1 до 3. В текстовых полях указываются номера рисунков, которые необходимо поменять местами. Требуется, чтобы после нажатия кнопки **Обменять** изображения переместились на нужные места.

Сначала проверим, правильно ли заданы номера изображений, если это не так, то выдадим сообщение. Переменная *z* служит для запоминания адреса первого графического изображения. Доступ к изображению с номером *r1* производится с помощью конструкции *document.images[r1-1]*. Для того чтобы на место изображения с номером *r1* поместить изображение с номером *r2*, требуется выполнить оператор присваивания:

```
document.images[r1-1].src=document.images[r2-1].src
```

И, наконец, на место изображения с номером *r2* помещается изображение, которое ранее было на месте с номером *r1*, и адрес которого запомнили в переменной *z*:

```
document.images[r2-1].src=z
```

Приведем полностью документ со сценарием (листинг 1).

Листинг 1. Перестановка изображений с помощью сценария

```
<HTML>
<HEAD>
<TITLE>Перестановка изображений</TITLE>
<script>
function chan(obj)
{ var r1=Number(obj.a1.value)
  var r2=Number(obj.a2.value)
  if ((r1<1)|| (r1>3)|| (r2<1)|| (r2>3))
    alert ("Неверно заданы номера рисунков!")
  else
    { var z=document.images[r1-1].src
      document.images[r1-1].src=document.images[r2-1].src;
      document.images[r2-1].src=z
    }
}
</script>
</HEAD>
<BODY>
<CENTER>
<H4>Галерея рисунков</H4><br>
<IMG src="p1.gif" width="90" name="pic1">
<IMG src="p2.gif" width="90" name="pic2">
<IMG src="p3.gif" width="90" name="pic3"> <br>
<FORM name=form1>
Рисунки с номерами<br>
```

```



```

```

<!-- copyright (t3) --><div align="center">Хостинг от <a href="http://www.ucoz.ru/" title="Создать
сайт бесплатно">uCoz</a><br /></div><!-- /copyright -->
</body>
</html>

```

Пример 2. Простое вертикальное меню

Напишем сценарий, реализующий вертикальное графическое меню. При наведении курсора мыши на пункт меню меняется цветовая палитра, соответствующая выделенному пункту меню.

Каждому пункту меню соответствует два изображения: первое изображение, когда пункт меню не выбран, второе - при выбранном пункте меню, цветовая палитра рисунка изменена. Графические изображения, соответствующие ситуации, когда пункты меню не выбраны, хранятся в файлах с именами *pch1.gif*, *pch2.gif*. Соответствующие им графические изображения с измененной палитрой хранятся в файлах с именами *wpch1.gif*, *wpch2.gif*.

Функция *img* имеет два параметра. Первый параметр задает выбор пункта меню, второй параметр - *n* - определяет номер пункта меню. От этого параметра зависит, какое изображение в документе требуется изменить *document.images[n-1].src* (вставить на этом месте рисунок "*wpch*" + *n* + ".gif" или *pch*" + *n* + ".gif"). Имя файла формируется динамически и представляет собой конкатенацию (слияние) строк, одна из составляющих которой - значение второго параметра. Если имена файлов не подчинены общему правилу, то в функции потребуется дополнительный анализ, какой файл подгрузить. Это сделать нетрудно, зная место в документе, из которого произошел вызов функции. Документ со сценарием, реализующий вертикальное графическое меню, представлен в листинге 2.

Листинг 2. Простое вертикальное меню

```

<HTML>
<HEAD>
<TITLE>Простое вертикальное меню</TITLE>
<script language="JavaScript">
function img(n, action)
{ if (action)
{ document.images[n-1].src="wpch"+n+".gif" }
else
{ document.images[n-1].src="pch"+n+".gif" }
}
</script>
</HEAD>
<BODY background="fon1.jpg">
<H2><FONT color="#0000ff">Содержание</FONT></H2>
<A href="tch1.htm" onmouseover="img(1,1)" onmouseout="img(1,0)">
<IMG src="pch1.gif" alt="форматирование текста" border="0" width="103"
height="35"></A><br>
<A href="tch2.htm" onmouseover="img(2,1)" onmouseout="img(2,0)">

```

```
<IMG src="pch2.gif" alt="форматирование текста" border="0" width="103"
height="35"></A><br>
<A href="tch3.htm" onmouseover="img(3,1)" onmouseout="img(3,0)">
<IMG src="pch3.gif" alt="форматирование текста" border="0" width="103"
height="35"></A><br>
```

```
<!-- copyright (t3) --><div align="center">Хостинг от <a href="http://www.ucoz.ru/" title="Создать
сайт бесплатно">uCoz</a><br /></div><!-- /copyright -->
</body>
</HTML>
```

При попадании курсора мыши в область изображения возникает событие *Mouseover*, параметр обработки события *onMouseOver* получает значение *img(p1,true)*.

Задания

1. Проверить примеры лабораторной работы. (15)
2. Написать сценарий выбора из трех изображений одного, которое вставляется ниже этих трех. Выбор осуществляется по клику на изображении. (15)
3. Написать сценарий картинки с "эффектом приближения", т.е. увеличения размеров как реакция на попадание курсора мыши в поле рисунка (использовать свойства *width* и *height*). (15)
4. Написать сценарий графического горизонтального меню с появляющейся стрелкой над пунктом, у которого находится курсор. (20)
5. Создать небольшой сайт – **Лаб_раб4_ФИО**: на главной странице тексты задач и ссылки на их выполнение. (25)

Все реализованные сценарии разместите в блоге (ссылку на архив), предварительно добавив логотип, в каждый файл (Web –страницу)

Лабораторная работа №7

(в журнале JS 5)

Переключатели

Данные удобно представлять с помощью элемента управления "переключатель" (или "радиокнопка") в том случае, когда из нескольких вариантов может быть выбран лишь один.

Пример 1. Вычисление площади фигуры.

Необходимо выбрать форму фигуры и определить ее площадь.

Пусть для выбора фигуры задана следующая форма:

```
<FORM name="form1">  
Введите значение  
<input type="text" name="data" size=10><hr>  
Укажите форму:<br>  
<input type="radio" name="fv" value=1>квадрат<br>  
<input type="radio" name="fv" value=2>Круг<br>  
<input type="radio" name="fv" value=3>треугольник<hr>  
<input type="reset" value="Отменить"><hr>  
Площадь: <input type="text" name="res" size=10>  
</FORM>
```

В этой форме шесть элементов. Первый элемент служит для ввода строки текста. Следующие три элемента образуют группу и являются переключателями. Пятый элемент создает кнопку сброса, нажатие которой отменяет все сделанные изменения. Шестой элемент является элементом для ввода строки.

Так как объект *forms* имеет свойство-массив *elements*, в котором содержатся ссылки на элементы формы в порядке их перечисления в теге `<FORM>`, то получить доступ к первому элементу формы можно либо с помощью значения параметра *name* этого элемента (*document.form1.data*), либо используя объектную модель JavaScript (*document.forms[0].elements[0]*). Вторым элементом рассматриваемой формы можно получить, если воспользоваться конструкцией *document.forms[0].elements[1]*. Это элемент-переключатель, определенный в составе группы элементов. В рассматриваемом примере группа элементов состоит из трех переключателей. В одну группу входят элементы с одинаковым значением параметра *name*. Доступ к следующим элементам группы может быть осуществлен так: *document.forms[0].elements[2]*, *document.forms[0].elements [3]*. Обязательный параметр *value* должен иметь уникальное значение для каждого элемента группы. Пользователь может выбрать только один вариант.

Напишем сценарий, в котором в зависимости от длины стороны или радиуса и формы выбранной фигуры вычисляется ее площадь. Для простоты будем считать, что фигура может иметь либо форму квадрата (задается его сторона), либо форму круга (задается радиус), либо форму равностороннего треугольника (задается его сторона).

Площадь рассматриваемых фигур считается по формуле ka^2 , где *k* - коэффициент, зависящий от формы выбранной фигуры; *a* - задаваемое пользователем значение. Вычисления будут проще, если коэффициент *k* указать в качестве значения параметра *value* соответствующего переключателя. Щелчок на элементе "переключатель" соответствует событию *click*, обработка которого заключается в вызове функции *test*. Функция имеет единственный параметр,

значение параметра - *value* переключателя, которое служит для вычисления площади фигуры.

HTML-код приведен в листинге 1.

Листинг 1. Вычисление площади выбранной с помощью переключателя фигуры

```
<HTML>
<HEAD>
<TITLE>Данные из формы типа "переключатель". Событие Click </TITLE>
<script language="JavaScript">
<!--//
function test (k)
{ var a= form1.data.value
if (a !="" )
form1.res.value= k*Math.pow(a,2)
else alert ("Введите значение")
}
//-->
</script>
</HEAD>
<BODY>
<FORM name="form1">
Введите значение
<input type="text" name="data" size=10>
<hr>
Укажите форму <br>
<input type="radio" name="fv" value=1
onClick="test(form1.elements[1].value)">квадрат<br>
<input type="radio" name="fv" value=3.14
onClick="test(form1.elements[2].value)">круг<br>
<input type="radio" name="fv" value=0.42
onClick="test(form1.elements[3].value)">треугольник<br>
<input type="reset" value="Отменить"><br>
Площадь: <input type="text" name="res" size=10>
</FORM>
</body>
</html>
```

Пример 2. Уникальные имена

Напишем сценарий, в котором определяется выбранный переключатель при щелчке мышью по определяемой в форме кнопке. Доступ к элементу требуется осуществить, используя уникальные имена (листинг 2).

Листинг 2. Уникальные имена

```
<HTML>
<HEAD>
<TITLE>Данные, представленные переключателем. Идентификаторы</TITLE>
<script language="JavaScript">
<!--//
function vid()
{ var d=document
var k=0;
if (d.all("i1").checked) k=1
```

```

else
if (d.all("i2").checked) k=2
else
if (d.all("i3").checked) k=3
else
if (d.all("i4").checked) k=4
d.all("resch").value=k
}
//-->
</script>
</HEAD>
<BODY>
<FORM name="form1">
Выберите форму витража<hr>
<input type="radio" name="fw" value=1 id="i1">прямоугольник<br>
<input type="radio" name="fw" value=2 id="i2">квадрат<br>
<input type="radio" name="fw" value=3 id="i3">треугольник<br>
<input type="radio" name="fw" value=4 id="i4">Круг<br>
<hr>
<input type="button" value="Выполнить" onClick="vid()">
<input type="reset" value="Отменить"><hr>
<input type="text" name="res" id="resch">
</FORM>
</BODY>
</HTML>

```

Пример 3. Выбор параметров обтекания изображения текстом

Напишем сценарий, который предоставляет возможность пользователю задавать значения параметров, определяющих, к какому полю окна (левому или правому) прижимается изображение, и, соответственно, с какой стороны текст его обтекает.

Если значение параметра *align* равно *Left*, то изображение прижимается к левому краю окна просмотра браузера, а текст или другие элементы документа "обтекают" изображение с правой стороны. Текст, размещаемый рядом с изображением, может занимать несколько строк. По умолчанию значение параметра *align* равно *Left*. При нажатии на кнопку **Обновить** для изображения и текста будут установлены значения параметров, принимаемых по умолчанию.

Пример HTML-кода, который содержит сценарий, обеспечивающий выполнение действий, задаваемых пользователем, приведен в листинге 3.

Листинг 3. Обтекание текстом изображения

```

<HTML>
<HEAD>
<TITLE>Изображение и текст. Обтекание</TITLE>
<script>
<!--
function chpict(obj)
{
if ((obj.elements[0]).checked)
document.mypict.align="Left"
else
document.mypict.align="Right"

```

```

}
function rset(obj)
{document.mypict.align="Left"}
//-->
</script>
</HEAD>
<BODY>
<CENTER>
<H4>Изображение и текст.
Обтекание изображения текстом</H4>
</CENTER>
<FORM name="form1">
Выберите значение параметра выравнивания нажмите кнопку <B>Просмотр</B>.<br>
<PRE>
<input type="radio" name="alg" checked value=ld>(left) изображение выравнивается по левому
краю
<input type="radio" name="alg" value=rd>(right) изображение выравнивается по правому краю
</PRE>
<input type="button" value= "Просмотр" onclick="chpict(form1)">
<input type="reset" value="Отменить" onclick="rset(form1)"> </FORM>
<TABLE bgcolor="F8F8FF">
<TR><TD>Иван Иванович Шишкин является одним из основоположников
русского национального пейзажа.
<IMG src=pl.jpg name=mypict align=left border=3 width=310>
В полотне "Рожь" Шишкин создал образ большой эпической силы
и подлинно монументального звучания. Могучая, полная
богатырских сил природа, богатый, привольный край. (Т. Юрова)
</TD></TR>
</TABLE>
</body>
</HTML>

```

Если изображение рассматривается как элемент строки, то значения параметров выравнивания задают расположение изображения относительно строки текста. Верхняя граница изображения может быть выровнена либо по самому высокому текстовому элементу текущей строки, либо по самому высокому элементу в строке (например, другому изображению). Базовой считается нижняя часть линии текста, которая проводится без учета нижней части некоторых символов. Середину изображения можно выровнять либо по базовой линии, либо по середине текущей строки. Нижнюю часть изображения можно выровнять по базовой линии, либо по нижней границе текущей строки.

Пример 4. Расположение текста и изображения в ячейке таблицы

Напишем сценарий, который позволяет продемонстрировать, как изменяется содержимое ячейки таблицы в зависимости от значений параметров горизонтального и вертикального выравнивания.

В документе слева находится список параметров, которые пользователь выбирает для горизонтального и вертикального выравнивания. В правой части в ячейке таблицы хранится изображение, положение которого будет меняться в зависимости от выбранных переключателей после нажатия кнопки **Просмотр**. (рис. 1.)

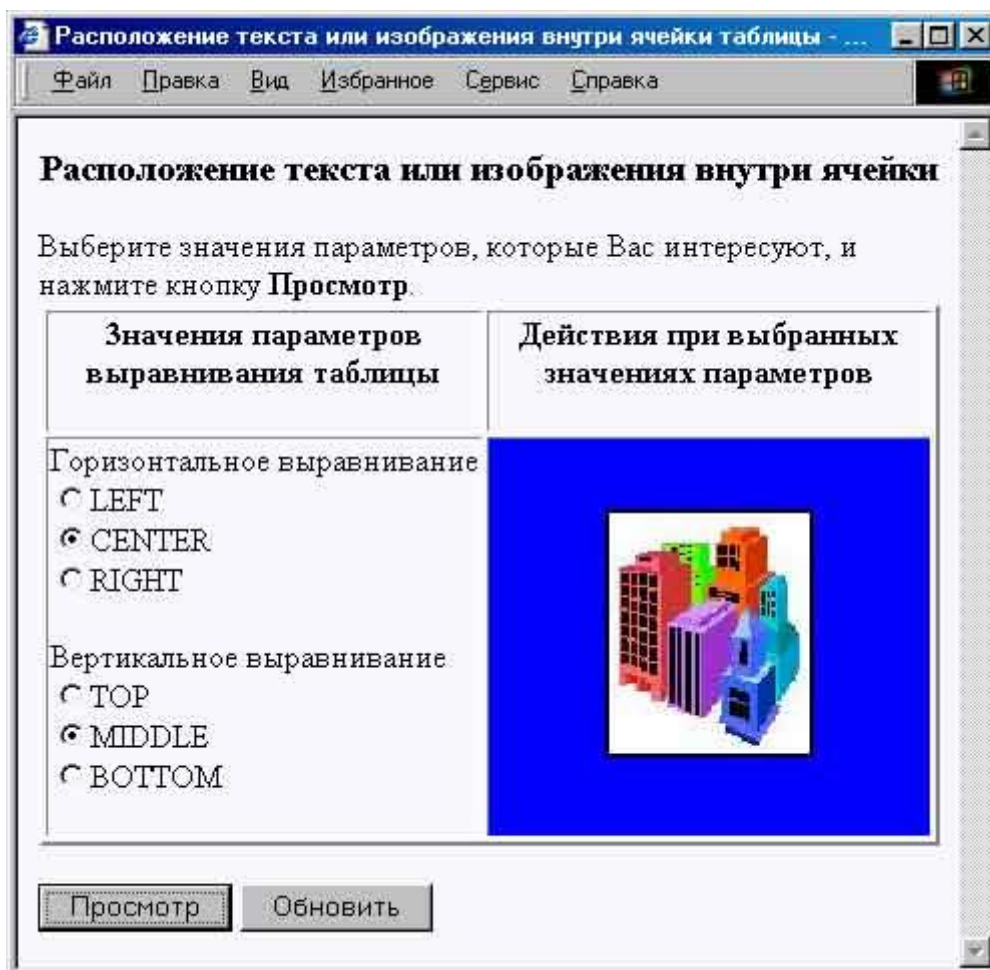


Рис. 1. Расположение изображения внутри ячейки

После нажатия кнопки **Просмотр** выполняется функция, анализирующая, какие параметры выбраны, и присваивает определенные значения атрибутам выравнивания содержимого ячейки. В правой части документа изображение занимает определенное параметрами выравнивания место в ячейке таблицы. Доступ к ячейке таблицы осуществляется по значению параметра `id`. При нажатии кнопки **Обновить** изображение требуется вернуть на место. По умолчанию горизонтальное выравнивание предполагает расположение содержимого по левому краю, вертикальное выравнивание — по верхней границе. HTML-код, содержащий сценарий, представлен в листинге 4.

Листинг 4. Расположение текста или изображения внутри ячейки таблицы

```
<HTML> <HEAD>
<TITLE>Расположение текста или изображения
внутри ячейки таблицы</TITLE>
<script>
<!--
function set(obj)
{ var d=document
if(obj.elements[0].checked) {d.all("itab").align="LEFT"}
else
if(obj.elements[1].checked) {d.all("itab").align="MIDDLE"}
else
if(obj.elements[2].checked) {d.all("itab").align="RIGHT"}
```

```

if(obj.elements[3].checked) {d.all("itab").vAlign="TOP"}
else
if(obj.elements[4].checked) {d.all("itab").vAlign="MIDDLE"}
else
if(obj.elements[5].checked) {d.all("itab").vAlign="BOTTOM"}
}
function rset(obj)
{ var d=document
d.all("itab").align="LEFT"
d.all("itab").vAlign="TOP"
}
//-->
</script>
<BODY bgcolor="F8F8FF">
<H3 align=center>Расположение текста или изображения
внутри ячейки</H3>
Выберите значения параметров, которые Вас интересуют,
и нажмите кнопку <B>Просмотр</B>.
<TABLE border=2>
<TR>
<TD><H4 align=center>Значения параметров выравнивания
таблицы</H4>
<TD><H4 align=center>Действия при выбранных значениях
параметров</H4>
</TR>
<TR>
<TD>
<FORM name="form1">
Горизонтальное выравнивание<br>
<input type="radio" name="al" value=0 checked>LEFT<br>
<input type="radio" name="al" value=1>CENTER<br>
<input type="radio" name="al" value=2>RIGHT<P>
Вертикальное выравнивание<br>
<input type="radio" name="vl" value=3 checked>TOP<br>
<input type="radio" name="vl" value=4>MIDDLE<br>
<input type="radio" name="vl" value=5>BOTTOM<br>
</FORM>
</TD>
<TD id="itab" bgcolor=blue background="firering.jpg" height=160
valign=TOP align=LEFT>
<IMG src="6rkranim.gif" border=2 height=120>
</TD>
</TR>
</TABLE><P>
<input type="button" value= "Просмотр" onclick="set(form1)">
<input type="reset" value="Обновить" onclick="rset()">
</BODY>
</HTML>

```

Задания

1. Проверить примеры из лабораторной работы. (4 з 20б)

2. Напишите сценарий, который позволяет продемонстрировать изменения размеров и положения на странице горизонтальной линии так, как показано на рис. 2. (30)

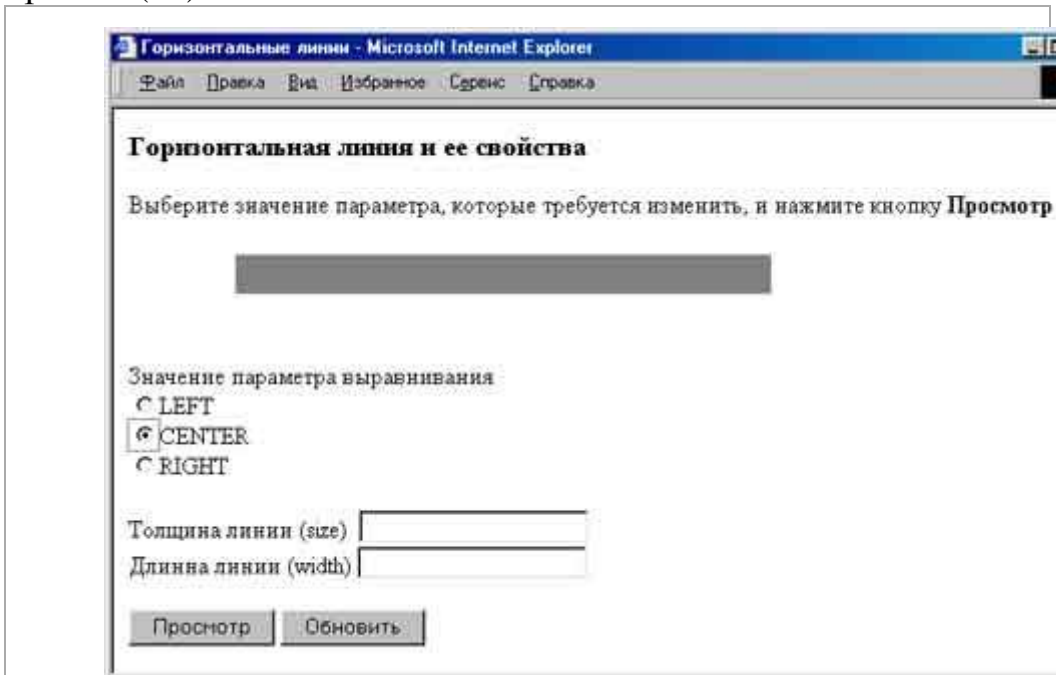


Рис. 2. Исследование свойств горизонтальной линии

Добавьте в сценарий изменение цвета линии (+5)

3. Разработайте анкету, определяющую характер человека. Пользователю предлагается ответить "да" или "нет" на следующие вопросы.

1. Считаете ли Вы, что у многих ваших знакомых хороший характер?
2. Раздражают ли Вас мелкие повседневные обязанности?
3. Верите ли Вы, что ваши друзья преданы Вам?
4. Неприятно ли Вам, когда незнакомый человек делает Вам замечание?
5. Способны ли Вы ударить собаку или кошку?
6. Часто ли Вы принимаете лекарства?
7. Часто ли Вы меняете магазин, в который ходите за продуктами?
8. Продолжаете ли Вы отстаивать свою точку зрения, поняв, что ошиблись?
9. Тяготят ли Вас общественные обязанности?
10. Способны ли Вы ждать более 5 минут, не проявляя беспокойства?
11. Часто ли Вам приходят в голову мысли о Вашей невезучести?
12. Сохранилась ли у Вас фигура по сравнению с прошлым?
13. Можете ли Вы с улыбкой воспринимать подтрунивание друзей?
14. Нравится ли Вам семейная жизнь?
15. Злопамятны ли Вы?
16. Находите ли Вы, что стоит погода, типичная для данного времени года?
17. Случается ли Вам с утра быть в плохом настроении?
18. Раздражает ли Вас современная живопись?

19. Надоедает ли Вам присутствие чужих детей в доме более одного часа?

Поставьте плюс, если вы ответили "да" в вопросах с номерами 3, 9, 10, 13, 14, 19 и "нет" в вопросах с номерами 1, 2, 4, 5, 7, 8, 11, 12, 14, 16, 17, 18. Посчитайте сумму набранных очков. Если она оказалась более 15, то у вас покладистый характер; если сумма в интервале от 8 до 15, то вы не лишены недостатков, но с вами можно ладить; если сумма менее 8 очков, то вашим друзьям можно посочувствовать.

Напишите сценарий обработки теста. Ответ на каждый из вопросов представьте с помощью переключателя. Для каждого из трех возможных результатов предусмотрите вывод соответствующего результату изображения. (40)

4. На последующих лабораторных работах вам предстоит создать сайт по следующим тематикам:

1. Едим дома
2. Дача и огород
3. Ремонт в квартире
4. Антивирусные программы
5. Охота
6. Рыбалка
7. Уход за собой
8. Авто
9. Услуги в интернете
10. Досуг и отдых
11. Мебель и интерьер
12. Бытовая техника и

электроника

13. Компьютерная и

цифровая техника

14. Эзотерика, религия
15. Диеты
16. Образование
17. Программное

обеспечение

Тему сайта вы можете выбрать самостоятельно, но на одну тему могут записаться не более 2 человек, и к сайтам «двойникам» будет особо пристальное внимание. (структура и содержание д.б. разное!!!! Сами следите, чтобы «конкурент» не воспользовался той же информацией.)

Чтобы записаться, сначала посмотрите в [таблице](#), какие темы уже заняты, потом заполните [форму](#).

К следующему занятию:

Исследуйте выбранную вами тему в интернете, накопите ссылки на сайты, с которых будете брать информацию.

Далее вам предстоит:

- Создание карты сайта
- Создание теста по выбранной теме
- Создание сайта

(подробные задания в последующих лабораторных работах)

Лабораторная работа № 8.

I Создание карты сайта

Идея взята с сайта <https://sites.google.com/site/inftech11/home/homub/for5>

Ознакомиться с содержанием материалов для [самостоятельного изучения](#)

Задача: Спроектировать структуру сайта (разделы, страницы... подробнее см. лабораторную работу №7) и изобразить её при помощи диаграммы связей Gliffy.net (30)

Карта сайта создается **по выбранной вами теме** (см. л/р5 последнее задание).

Диаграмма связей или интеллект-карта, (англ. Mind map) — способ изображения процесса общего системного мышления с помощью схем. Также может рассматриваться как удобная техника альтернативной записи. Диаграмма связей реализуется в виде древовидной схемы, на которой изображены слова, идеи, задачи или другие понятия, связанные ветвями, отходящими от центрального понятия или идеи.

В качестве примера рассмотрим структуру сайта inftech.spb.ru.

1. Главная страница – это информация, которая предстаёт перед пользователем при переходе его по адресу сайта. Другими словами, главная страница – это первое, с чем сталкивается посетитель, оказываясь на сайте. Главной страницей данного сайта является **доска объявлений (ИТ потоки)** для студентов.

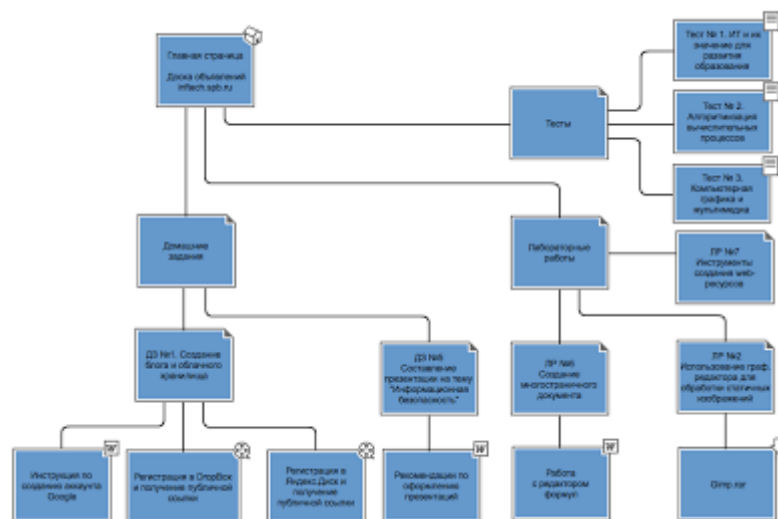
2. При помощи основного меню можно попасть в следующие **разделы:**

1. Лекции
2. Лабораторные работы
3. Домашние задания
4. Тесты
5. Литература
6. Результаты

3. Каждый из разделов, как правило, содержит ссылки на подстраницы. Например, в разделе "Домашние задания" содержится перечень ссылок на конкретные работы.

4. В некоторых лабораторных работах и домашних заданиях на странице прикреплены дополнительные материалы: текстовые документы, видеоролики или установочные файлы программ.

В ходе анализа любого сайта можно заметить, что все разделы и страницы строго подчинены иерархической (древовидной) структуре. Изобразим структуру данного сайта графически, спланируем возможные перемещения посетителя сайта, т.е. построим **карту сайта** (приведен лишь фрагмент карты, т.к. данный сайт содержит достаточно много страниц в каждом разделе):



(щелкните по изображению для увеличения)

В редакторе Gliffy.net для построение карты сайта предусмотрен специальный набор инструментов в меню слева – "Site map".

II Создание теста средствами google

Задание: Средствами **форм google** составить тест по материалам **вашего** сайта, выбранного вами в лабораторной работе 5.

В данном задании вам предстоит создать тест на основе форм Google Диск. Основное предназначение форм Google - проведение опросов, анкетирования и других подобных видов исследований, а также распределенный сбор данных, например: на собственном сайте вы можете создать подобие нашей регистрационной формы для получения адресов электронной почты и блогов (запись на курс). Также вы можете предложить участникам семинара или конференции пройти предварительную запись через формы Google, например для того, чтобы определить число посетителей и заранее выписать именные сертификаты об участии.

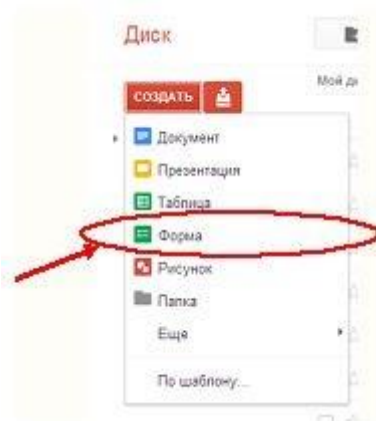
Проведение контроля знаний - нетипичный, но возможный вариант использования форм Google. Google не умеет проверять правильность ответов и выставлять отметку на основе заранее заданных критериев. Однако вы можете самостоятельно задать формулы для обработки ответов на форму, т.к. все результаты хранятся в электронной таблице (наподобие Microsoft Excel или Open Office Calc).

Этапы работы:

1. Создание формы (теста) (30б)
2. Настройка сохранения ответов в таблицу
3. Создание логических формул для проверки правильности ответов (+30б)
4. Интеграция теста в сайт Google
5. Публикация теста и таблицы в блог

Создание формы

Откройте Google Диск и создайте новый документ типа Форма (Создать – Форма).

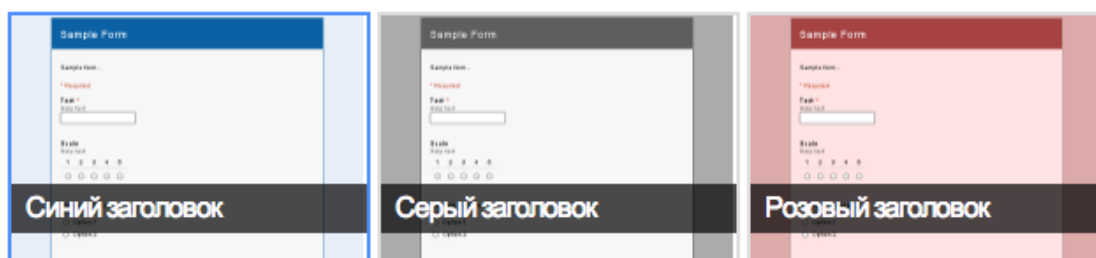


После создания документа автоматически откроется редактор форм и предложит ввести название формы и выбрать тему оформления:

Выберите название и тему

Заголовок

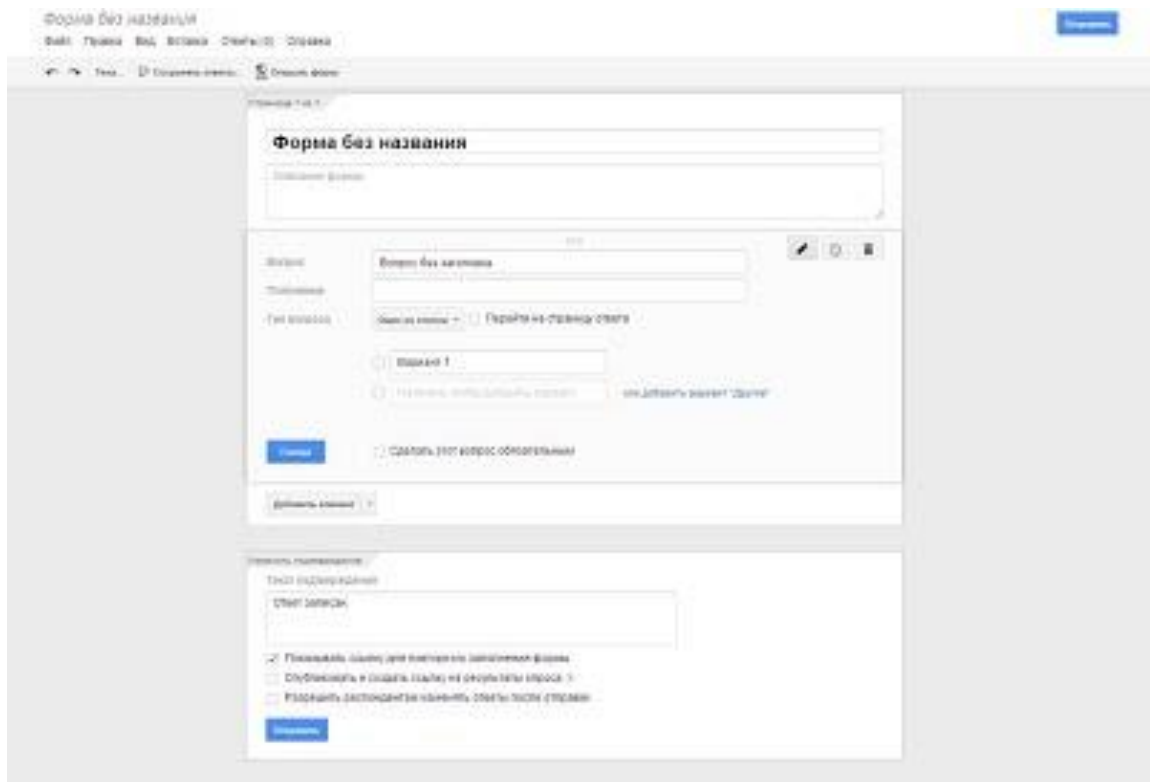
Тема: Синий заголовок



Показывать в новых формах

Если данное окно не появилось автоматически при создании формы, то тему изменить можно нажав кнопку «Тема» на панели инструментов.

После выбора темы в основном окне редактора вы можете также указать краткое пояснение в поле «Описание формы»:

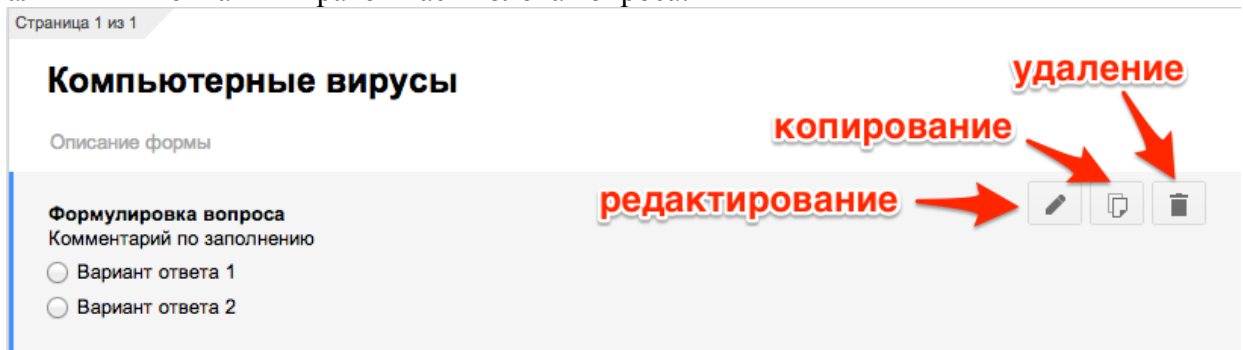


Далее необходимо последовательно добавить 10 вопросов (обратите внимание, что первый вопрос уже добавлен). В каждом вопросе необходимо вписать формулировку вопроса, выбрать тип вопроса и при необходимости варианты ответов и пояснение. Обратите внимание на галочку «Сделать этот вопрос обязательным»: не ответив на данный вопрос, учащийся не сможет завершить тестирование.

Для добавления новых вопросов используйте меню вставка или кнопку «Добавить элемент» в конце страницы.

! Грамотно сформулированное пояснение упрощает последующую обработку теста. Например, можно попросить участников тестирования набирать свою фамилию русскими буквами, чтобы после написания теста была возможность отсортировать строки ответы по фамилии в алфавитном порядке.

Для редактирования или удаления существующего вопроса воспользуйтесь специальными кнопками в правой части блока вопроса:



Типы вопросов

В зависимости от формулировки вопроса стоит выбирать соответствующий тип вопроса. В формах Google представлено несколько типов вопросов, но наиболее популярные из них относятся к простому типу: текст, один из списка, несколько из списка, выпадающий список. Данные типы вопросов проще создавать и

обрабатывать, но сложные вопросы позволяют получить более интересные тесты и результаты.

В рамках данной лабораторной работы рекомендуется использовать простые типы вопросов, т.к. это упростит обработку ответов, но вы можете попробовать использовать также какой-нибудь сложный тип вопроса.

Текст и Текст (абзац)

Данный тип предназначен для открытых вопросов, в которых тестируемый вводит ответ при помощи клавиатуры. Часто при помощи таких вопросов получают имя, фамилию, адрес электронной почты и другие сведения, которые невозможно предугадать заранее. Для типа текст (абзац) вместо обычного поля ввода будет использовано поле для многострочного ввода.

The screenshot shows a question editor interface. At the top right, there are three icons: a pencil (edit), a document (copy), and a trash can (delete). Below these is a text input field containing "Вопрос без заголовка". Underneath is another empty text input field labeled "Пояснение". Below that is a dropdown menu for "Тип вопроса" with "Текст" selected. A dashed box labeled "Ответ" is positioned below the dropdown. At the bottom left is a blue button labeled "Готово". To its right is a checkbox labeled "Сделать этот вопрос обязательным". At the very bottom is a button labeled "Добавить элемент" with a small downward arrow.

Тип вопроса: Один из списка

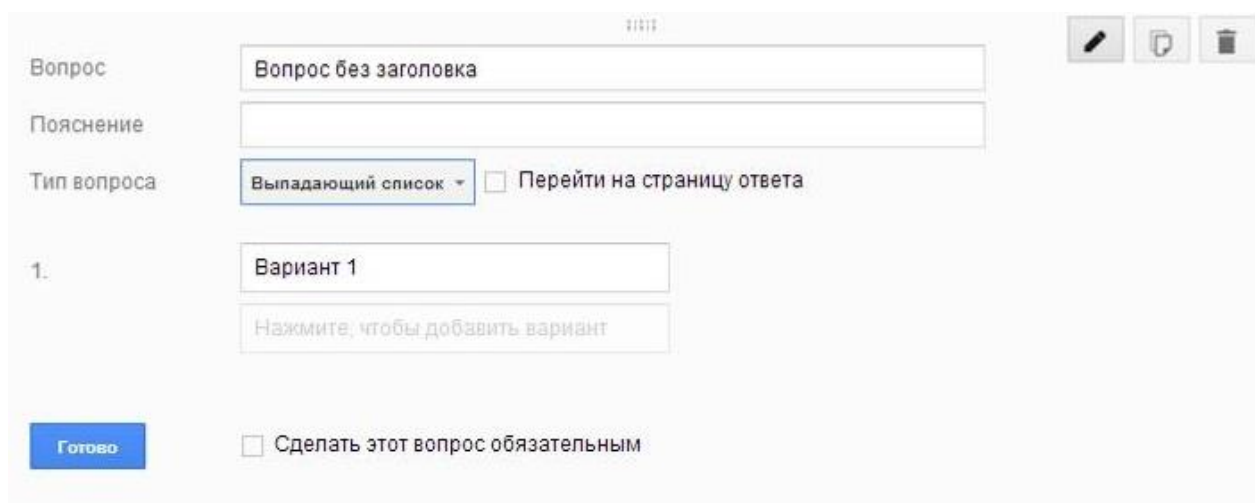
Обычно данный тип используется для закрытых вопросов, где тестируемый выбирает только один ответ из нескольких предложенных вариантов. Поле для первого варианта ответа доступно изначально, а для добавления дополнительных вариантов щелкните по полю «Нажмите, чтобы добавить вариант». Обратите внимание, что ставить точку или галочку напротив правильного ответа не нужно.

Отличить данный тип вопроса можно по характерному элементу – круглому полю для выбора варианта (радиокнопка).

The screenshot shows a question editor interface. At the top right, there are three icons: a pencil (edit), a document (copy), and a trash can (delete). Below these is a text input field containing "Вопрос без заголовка". Underneath is another empty text input field labeled "Пояснение". Below that is a dropdown menu for "Тип вопроса" with "Один из списка" selected. To the right of the dropdown is a checkbox labeled "Перейти на страницу ответа". Below the dropdown are two radio button options: "Вариант 1" and "Нажмите, чтобы добавить вариант". To the right of the second radio button is the text "или добавить вариант 'Другое'". At the bottom left is a blue button labeled "Готово". To its right is a checkbox labeled "Сделать этот вопрос обязательным". At the very bottom is a button labeled "Добавить элемент" with a small downward arrow.

Тип вопроса: Выпадающий список

Данный тип вопроса работает аналогично предыдущему (один из списка). Единственное отличие заключается в представлении вопроса во время прохождения теста: вместо списка вариантов с переключателями, ученик увидит более компактный выпадающий список, из которого он сможет выбрать только один вариант ответа.



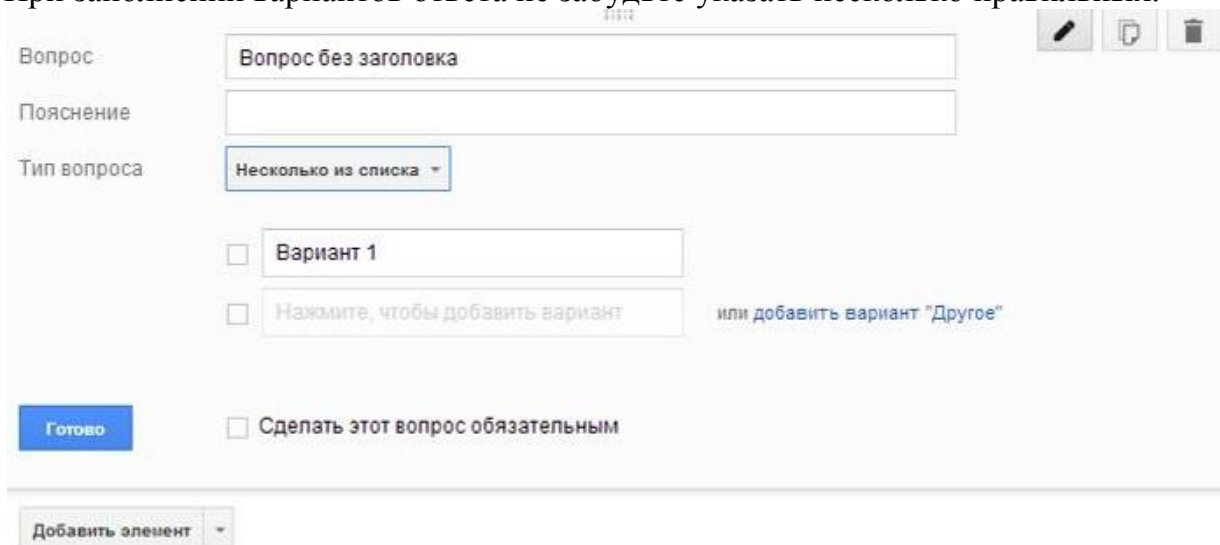
The screenshot shows a question editor interface. At the top, there are three icons: a pencil (edit), a document (copy), and a trash can (delete). Below these is a text input field for the question, containing the text "Вопрос без заголовка". Underneath is a text input field for the explanation. The "Тип вопроса" (Question type) is set to "Выпадающий список" (Dropdown list). There is a checkbox labeled "Перейти на страницу ответа" (Go to answer page) which is currently unchecked. Below this, there is a list of options starting with "1." and "Вариант 1". There is a text input field for adding a new option with the placeholder text "Нажмите, чтобы добавить вариант". At the bottom left is a blue button labeled "Готово" (Done). At the bottom right is a checkbox labeled "Сделать этот вопрос обязательным" (Make this question mandatory) which is unchecked.

Добавить элемент ▾

Тип вопроса: Несколько из списка

Данный тип вопроса предполагает выбор нескольких ответов из предложенных вариантов. Обычно подобные вопросы выглядят как список вариантов с элементами управления в виде поля с галочками (checkbox).

При заполнении вариантов ответа не забудьте указать несколько правильных.



The screenshot shows a question editor interface. At the top, there are three icons: a pencil (edit), a document (copy), and a trash can (delete). Below these is a text input field for the question, containing the text "Вопрос без заголовка". Underneath is a text input field for the explanation. The "Тип вопроса" (Question type) is set to "Несколько из списка" (Multiple choice). Below this, there is a list of options starting with "Вариант 1" and "Нажмите, чтобы добавить вариант". There is a checkbox next to each option. To the right of the second option, there is a link that says "или добавить вариант 'Другое'" (or add option 'Other'). At the bottom left is a blue button labeled "Готово" (Done). At the bottom right is a checkbox labeled "Сделать этот вопрос обязательным" (Make this question mandatory) which is unchecked.

Добавить элемент ▾

Также через меню «Вставка» можно добавить в форму изображения и видео. **Не забудьте добавить свой логотип!** Изображения и видео будут вставлены как отдельные элементы. Обратите внимание, что все элементы в форме можно менять местами просто перетаскивая их мышью.

Если вы собираетесь использовать в вопросе изображение или видео, то рационально будет поместить их непосредственно перед вопросом.

В нижней части окна, под вопросами вы можете настроить поведение формы и сопроводительную информацию:

Изменить подтверждение

Текст подтверждения

Ответ записан.

Показывать ссылку для повторного заполнения формы

Опубликовать и создать ссылку на результаты опроса ?

Разрешить респондентам изменять ответы после отправки

[Отправить](#)

В нашем случае форма представляет собой тест, т.е. включать данные опции не следует.

После ввода всех вопросов необходимо нажать кнопку «Отправить».

Сохранение ответов в таблицу

После каждого ответа на вашу форму ответы будут сохраняться непосредственно в форме, однако наиболее гибким и удобным вариантом является создание отдельной таблицы для ответов в Google Диск.

Нажмите кнопку «Сохранять ответы...» на панели инструментов и диалоговом окне укажите название для новой таблицы:

Сохранение ответов

Новая таблица

Форма без названия (Ответы)

Новый лист в существующей таблице...

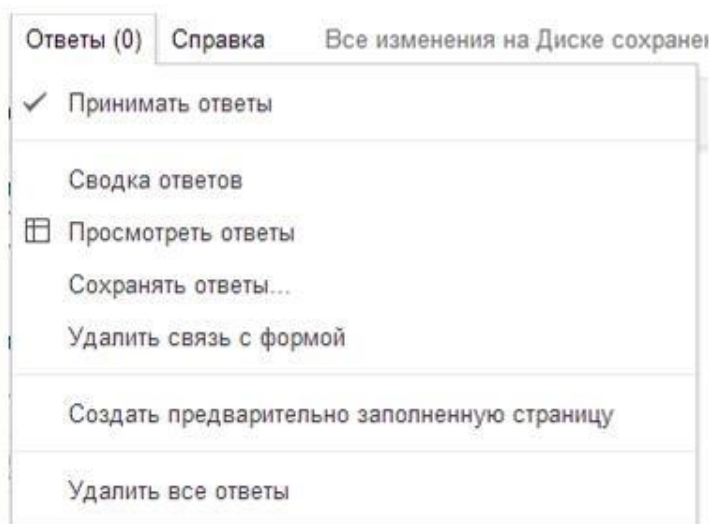
Всегда создавать таблицу ?

[Создать](#) [Сохранять ответы только в Формах](#) [Подробнее...](#)

ОТВЕТЫ, ПОЛУЧЕННЫЕ ЧЕРЕЗ ФОРМУ → ТАБЛИЦА

Изменение, упорядочивание и анализ данных не затронет исходные ответы, полученные через форму.

Просмотр ответов доступен в двух видах: через созданную таблицу (просмотреть ответы) или в виде сводки.



В таблице каждый новый ответ будет записан в отдельной строке в хронологическом порядке, где в первой ячейке будет указаны дата и время заполнения формы, а далее ответы на вопросы. Ответ на каждый из типов вопроса (кроме сетки) занимает ровно одну ячейку. В сводке ответов вы можете найти статистические данные по форме с диаграммами и графиками.

Добавление формул для проверки правильных ответов

В качестве примера рассмотрим тест из четырех вопросов:

1. Фамилия Имя
2. Открытый вопрос
3. Один из списка
4. Несколько из списка

Тогда тест будет выглядеть следующим образом:

Страница 1 из 1

Компьютерные вирусы

Описание формы

Фамилия Имя *

Как называется вредоносное программное обеспечение, способное создавать копии самого себя и внедряться в код других программ, а так же распространять свои копии по разнообразным каналам связи? *

На чем основано действие антивирусной программы? *

На ожидании начала вирусной атаки

На сравнении программных кодов с известными вирусами

На получении сигнала из центра обработки

Какие из перечисленных программ относятся к антивирусам? *

Kaspersky Internet Security

Windows Commander

Microsoft Security Essentials

GIMP

После заполнения формы данные соберутся в таблицу:

	A	B	C	D	E
1	Отметка времени	Фамилия Имя	Как называется вредоносное программное обеспечение...	На чем основано действие антивирусной программы?	Какие из перечисленных программ относятся к антивирусам?
2	27.11.2013 5:56:12	Иванов Иван	вирус	На сравнении программных кодов с известными вирусами	Kaspersky Internet Security, Microsoft Security Essentials
3	27.11.2013 5:56:28	Мария Петрова	антивирус	На ожидании начала вирусной атаки	Kaspersky Internet Security, Microsoft Security Essentials
4	27.11.2013 5:56:48	Петр Алексеев	вирус	На получении сигнала из центра обработки	Kaspersky Internet Security, Windows Commander, Microsoft Security Essentials
5	27.11.2013 5:57:34	Елизавета Скворцова	операционная система	На ожидании начала вирусной атаки	Microsoft Security Essentials, GIMP
6					
7					
8					

ответы через запятую



Для удобства к столбикам с вопросами можно применить условное форматирование из меню формат:

C	D	E	F	G
Как называется вредоносное программное обеспечение...	На чем основано действие антивирусной программы?	Какие из перечисленных программ относятся к антивирусам?		
вирус				
антивирус				
вирус				
операционная система				

Условное форматирование

Текст в точности Текст: Фон:

например, "готово" или "Джонатан"

[+ Добавить еще одно правило](#)

После получения нескольких пробных ответов на тест создайте второй лист для логических формул проверки правильности ответов:

Компьютерные вирусы (Ответы)

Файл Правка Вид Вставка Формат Данные Инструменты Форма Справка Комментарии Настройки доступа

fx `=IF(Ответы на форму!C2="вирус";1;0)` `=if(F2>=3; 3; if(F2=2; 2; if(F2=1; 1; 0)))`

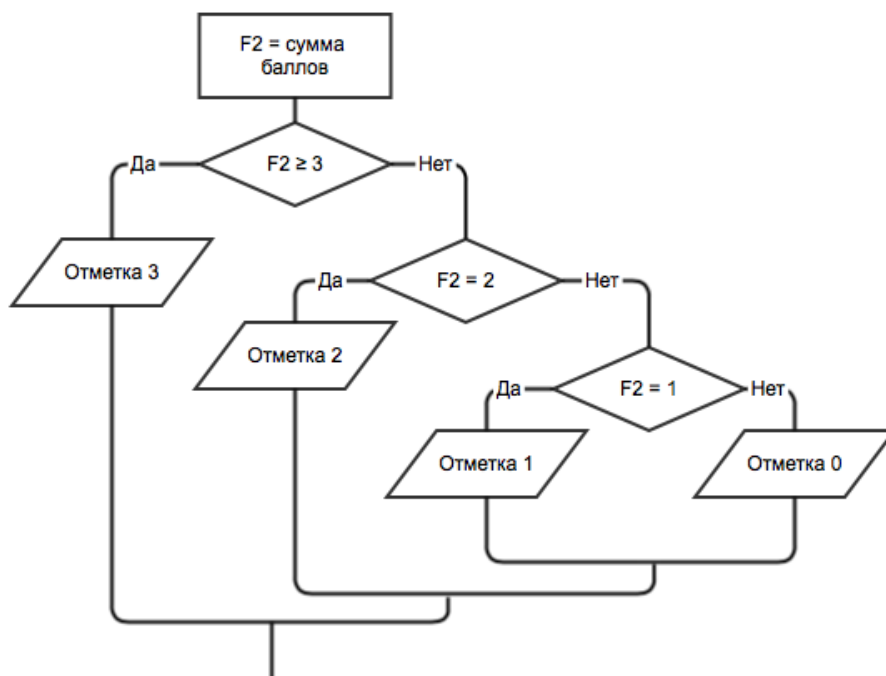
	A	B	C	D	E	F	G
1	Отметка времени	Фамилия Имя	Как называется вредоносное программное обеспечение...	На чем основано действие антивирусной программы?	Какие из перечисленных программ относятся к антивирусам?	Сумма баллов	Отметка
2	27.11.2013 5:56:12	Иванов Иван	1	1	1	3	3
3	27.11.2013 5:56:28	Мария Петрова	0	0	1	1	1
4	27.11.2013 5:56:48	Петр Алексеев	1	0	0	1	1
5	27.11.2013 5:57:34	Елизавета Скворцова	0	0	0	0	0
6							

`=SUM(C2:E2)`

`=IF(Ответы на форму!E2="Kaspersky Internet Security, Microsoft Security Essentials";1;0)`

Обратите внимание на формулу для вопроса с несколькими правильными вариантами ответа: варианты перечислены через запятую (точно так же, как на первом листе).

Вычисления отметки также основано на использовании формулы ЕСЛИ/IF, однако содержит несколько вложенных условий, которые можно было бы отобразить следующим образом при помощи блок-схем:



Измените формулы проверки правильности ответов, вычисления суммы баллов и отметки в соответствии с количеством вопросов и наименованием ячеек. К сожалению, при получении из формы новых ответов на листе

проверки формулы автоматически не создаются, поэтому после получения новых ответов **приходится копировать формулы** (например, выделив всю строчку, "протащить" маркер заполнения ячеек вниз).

Вставка формы на сайт/блог

Созданную форму можно интегрировать в сайт точно так же, как презентации из ЛР 7, через меню «Вставка-диск-форма». Формы, в отличие от презентаций, публикуются автоматически. В диалоговом окне добавления формы можно оставить поле «Ширина» пустым, тогда форма будет автоматически подстраиваться под ширину страницы.

! Для получения HTML-кода для встраивания на любой веб-ресурс воспользуйтесь командой «Файл - встроить в веб-страницу» в окне формы. Если вам необходимо получить только ссылку, например для распространения формы через социальные сети, то в окне с формой выполните команду «Файл - Отправить».

В качестве отчета о выполнении данной работы разместите в блоге ссылку на форму (вместо ссылки можно встроить HTML код), а также ссылку на таблицу с проверкой. Пожалуйста, не забудьте предоставить доступ к таблице (Настройки доступа - Уровни доступа, Изменить - Всем у кого есть ссылка).

Требования к содержанию теста

1. Тема теста совпадает с темой выбранного вами сайта, вопросы должны совпадать с содержанием вашего сайта.
2. В тесте должно быть не менее 10 вопросов, в закрытых вопросах не менее 3 вариантов ответа.
3. В тесте должны использоваться различные типы вопросов
4. Проверка правильности ответов должна осуществляться при помощи формул на втором листе
5. Тест должен быть встроен в одну из страниц сайта из Лабораторной работы №7. (это в будущем)

Лабораторная работа № 9

(в журнале JS 7).

Инструменты создания web-ресурсов.

Идея взята с сайта <https://sites.google.com/site/inftech11/home/labrab/lr7>

Цели: Ознакомиться с технологией создания веб-сайтов средствами Google, создать сайт по выбранной вами теме (см. л/р 5).

Задачи:

- на основе карты сайта из ЛР 6, создайте иерархическую структуру сайта (разделы и страницы)
- подберите и разместите материалы на страницах сайта

В ходе выполнения предыдущих лабораторных работ мы рассмотрели инструменты создания электронных ресурсов содержащих различные виды информации: неформатированного текста, статических изображений. Кроме того, в первых лабораторных работах рассмотрены два популярных инструмента тиражирования и доставки образовательной информации через Интернет: облачное хранилище и блог. Однако, пожалуй, самым распространенным способом обмена информацией в рамках образовательного процесса считается полноценный сайт (возможно в комбинации с облачным хранилищем) с удобной навигацией и четко структурированным материалом.

Содержание сайта

В рамках данной лабораторной работы вам предстоит разработать электронный образовательный ресурс по вашей теме. По карте сайта из Л/р №6 вам необходимо создать набор связанных друг с другом страниц (сайт) при помощи инструментов Google. Материалы для сайта можно найти в Интернете (презентации, видеоролики, изображения, текст...), однако ваша главная задача – правильно структурировать и отформатировать материал для сайта.

! Абсолютно не допускается бездумное копирование текста с различных сайтов без обработки.

Примерный план работы над сайтом может быть таким:

1. Продумайте содержание и структуру сайта (см. карту сайта)
2. Создайте новый сайт в среде Google.
3. Используя интернет-ресурсы, подберите материал и разместите его на страницах. (условия использования чужого материала: в конце фрагмента текста, таблицы, видео и т.д. вставляете ссылку с какого сайта взята информация)
4. Опубликуйте видеоролики и презентацию в блоге.

План работы:

- 1) Создание сайта
- 2) Редактирование страниц
- 3) Добавление страниц, создание иерархической структуры
- 4) Настройка

Этапы создания сайта средствами Google

Регистрация сайта

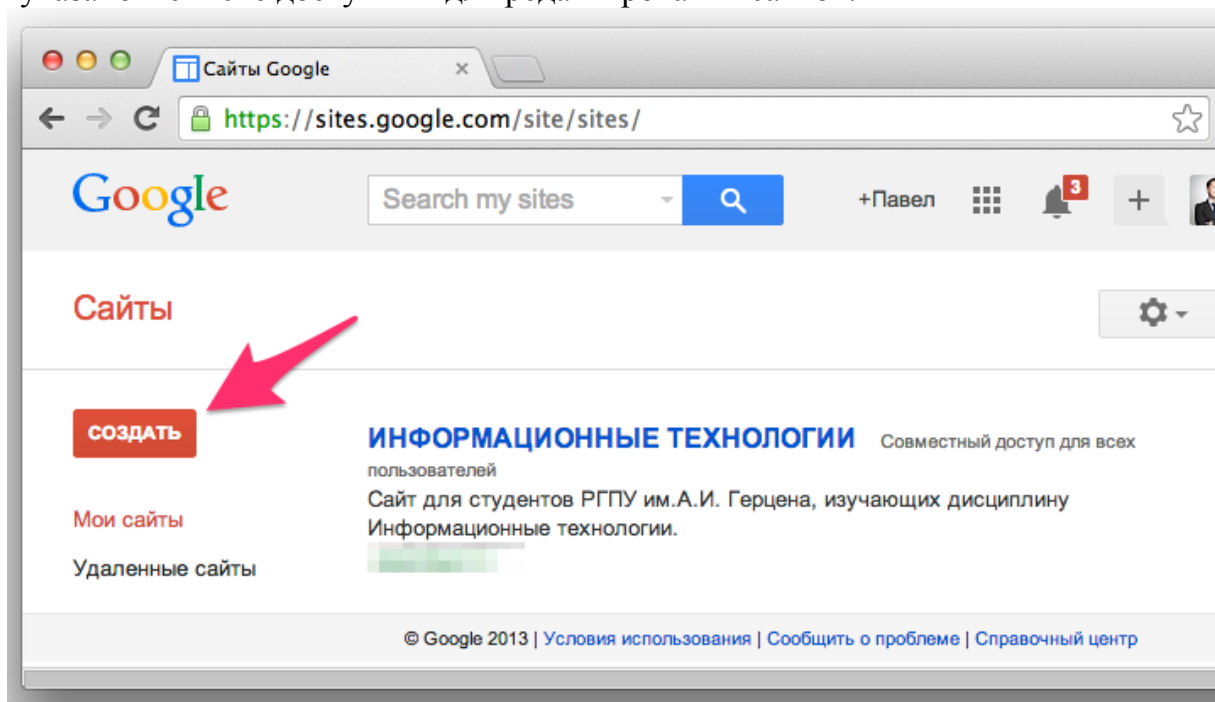
В данной лабораторной работе для создания сайта используется веб-приложение Google Сайты <https://sites.google.com/>. Перед началом работы необходимо войти в систему под учетной записью Google.

Видео:

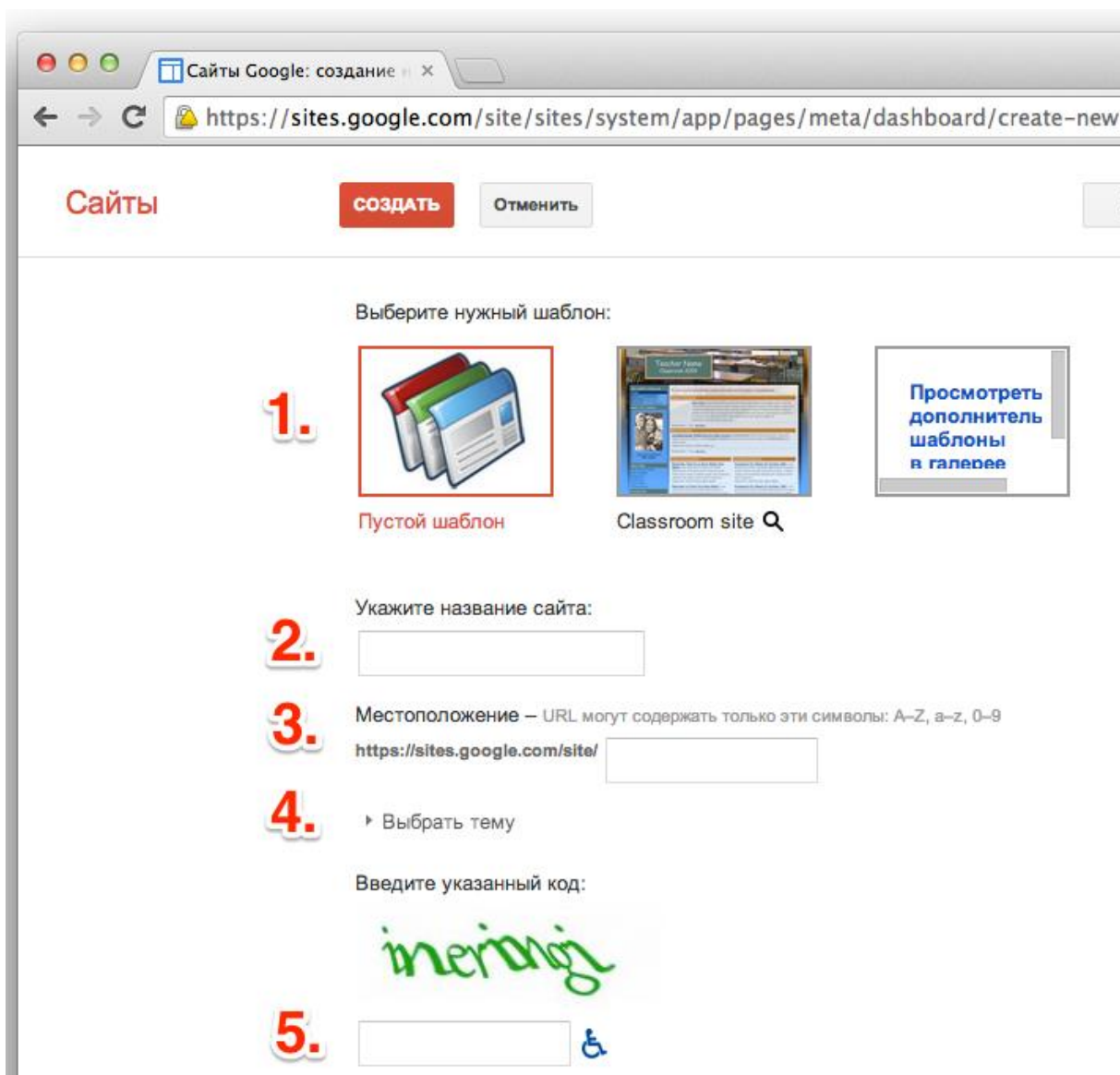
... нужно записать ... Приблизительный сценарий: открыть сайт, создать сайт, показать его в списке сайтов и обратить внимание на адресную строку.

Текстовая инструкция:

Для создания нового сайта на главной странице веб-приложения щелкните по кнопке «Создать». Обратите внимание на то, что после создания сайта, его название будет указано в списке доступных для редактирования сайтов:



В форме создания нового сайта заполните все необходимые поля:



Комментарии к заполнению:

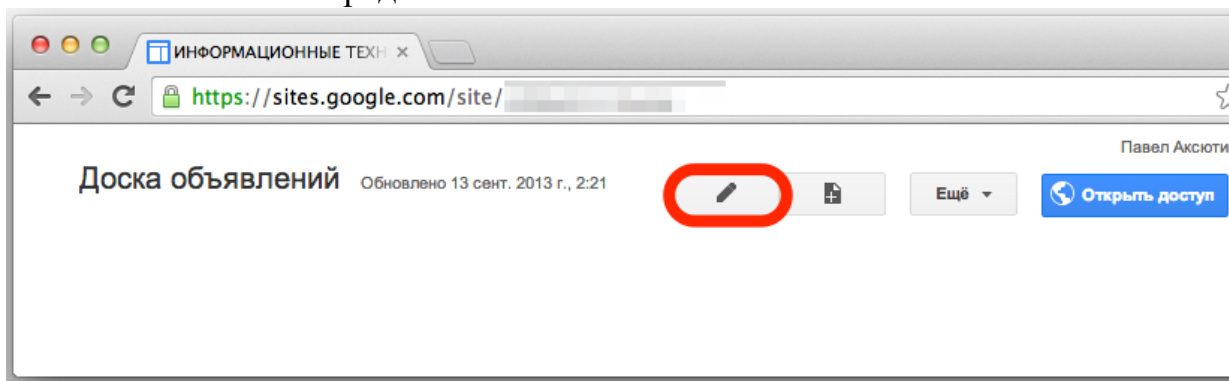
1. **Шаблон** (заданная структура и содержание сайта). Для ознакомления с различными вариантами шаблонов вы можете щелкнуть по ссылке «Просмотреть дополнительные шаблоны», однако в рамках данной работы выберите «**Пустой шаблон**».
2. **Название сайта.** Укажите краткое название сайта по-русски. Данный текст будет отображаться в верхней части сайта на всех страницах.
3. **Местоположение.** В данном поле необходимо указать последнюю (вариативную) часть адреса вашего будущего сайта. Будьте внимательны, скорее всего простые названия уже заняты, поэтому к адресу сайта нужно дописать фамилию и имя. Например: computervirus**ivanivanov**, тогда полный адрес сайта будет выглядеть следующим образом: `https://sites.google.com/site/computervirusivanivanov/`
4. Выберите любое подходящее к тематике сайта оформление из списка в разделе «**выберите тему**»
5. Введите код **САПТСНА**
После нажатия на кнопку «Создать», Google создаст новый сайт и откроет его главную страницу.

В случае неправильного ввода одного из полей или занятости выбранного местоположения Google выдаст ошибку и после устранения проблемы необходимо повторно нажать кнопку «Создать».

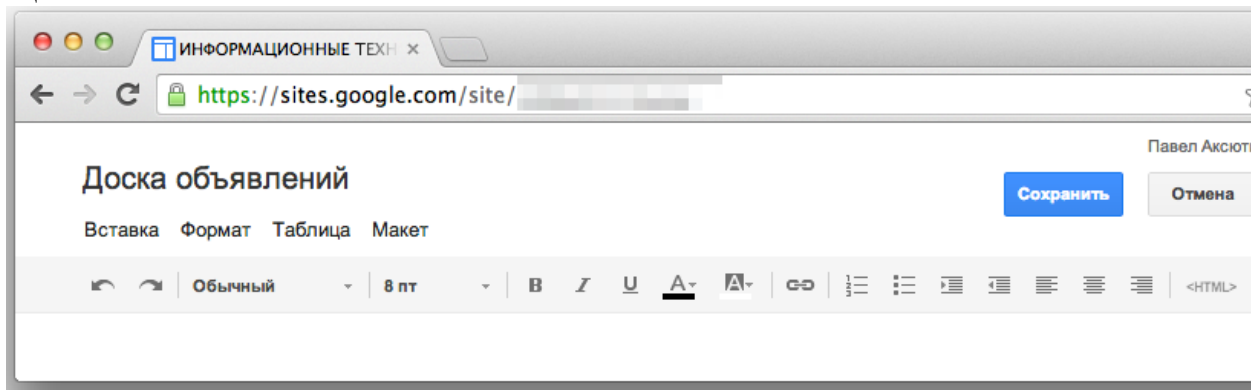
Редактирование страниц

Видеоролик про редактор (форматирование + макеты + изображения)

В отличие от традиционной технологии создания сайтов на основе языков разметки и программирования, данное веб-приложение работает в режиме WYSIWYG (What You See Is What You Get - «Что видишь, то и получаешь»). Для создания страниц сайта необязательно использовать язык HTML, т.к. Google создаст всю необходимую разметку в автоматическом режиме на основе заданного пользователем представления.



Для редактирования страницы необходимо нажать на кнопку «Изменить страницу» в правом верхнем углу страницы). Панель инструментов редактора содержит базовые команды форматирования абзацев и текста, а основное меню позволяет вставить дополнительные материалы на страницу, управлять таблицами и макетом сайта.

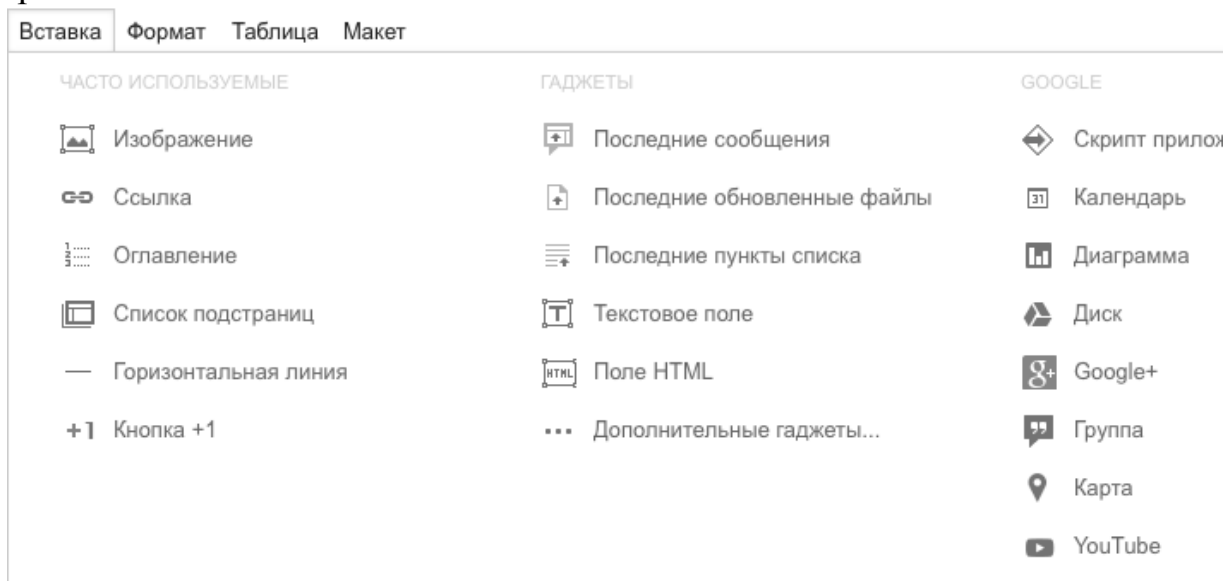


Изменение страницы в данном редакторе напоминает работу в текстовом редакторе, однако области для ввода текста строго подчиняются выбранному макету.

Для завершения работы с редактором и перехода к нормальному режиму сайта необходимо нажать кнопку «Сохранить». Google автоматически создает черновики страниц во время редактирования, поэтому потерять несохраненные изменения практически невозможно.

Вставка дополнительных материалов на страницу осуществляется через меню «Вставка».

Изображения и ссылки добавляются точно также, как и в блоге-портфолио Blogger.com. После выбора соответствующего пункта меню следуйте указаниям мастера.



Кроме того, в меню «вставка» доступны средства интеграции с другими сервисами Google: Календарь, карты Google Maps, Youtube и Google Диск. Интеграция с облачным хранилищем позволяет добавлять на страницы сайта текстовые документы, презентации, анкеты, электронные таблицы. В рамках данной лабораторной работы вам необходимо встроить презентацию и видеоролик. Создавать собственную презентацию и видеоролик не обязательно – можно найти готовые материалы в Интернете.

Пример встроенной презентации:

.... взять что-то готовое

Для того, чтобы презентация была доступна для вставки на страницу, она должна быть изначально создана в редакторе Google Drive, либо презентация PowerPoint загружена и преобразована в документ Google.



**Презентация
Google**



**Презентация
PowerPoint**

Видеоролик про вставку презентации

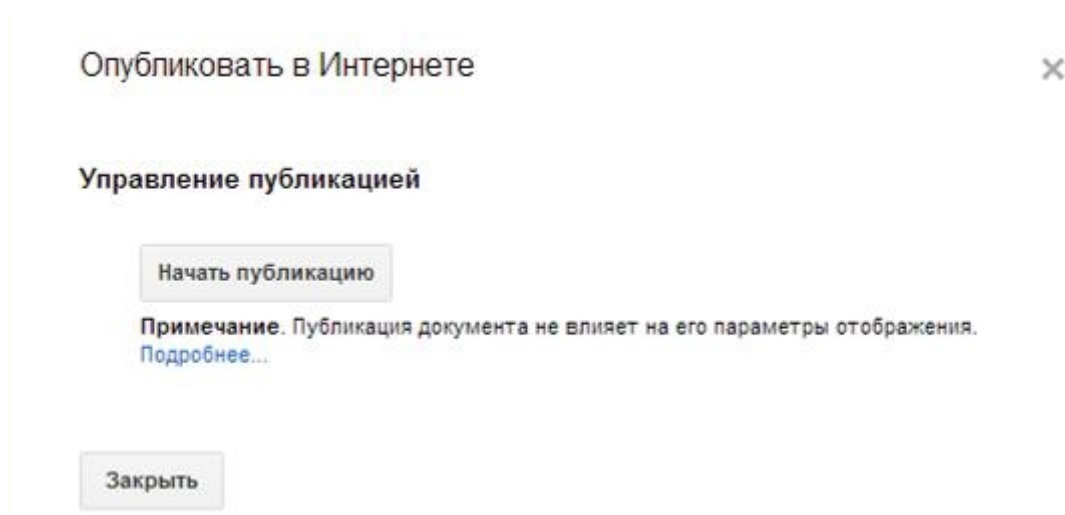
(вход в Google Диск + включение преобразования + загрузка презентации + встраивание в страницу)

Откройте Google Диск, убедитесь, что установлена галочка в меню «Настройки», «Настройки загрузки» напротив пункта «Преобразовывать загруженные файлы в формат Документов Google».

Загрузите файл в Google Диск, используя инструмент загрузки. Откройте презентацию и опубликуйте

! Обратите внимание, что Google не сможет преобразовать презентацию, если её объем более 4Мб.

Перед встраиванием в сайт презентацию необходимо опубликовать: в Google Диске выполните команду: Файл – Опубликовать в Интернете. В открывшемся диалоговом окне нажмите на кнопку «Начать публикацию».



! Получив «HTML-код для встраивания» вы можете использовать его для интеграции документа Google в любой веб-ресурс. Например, в блоге (blogger.com) вы можете перейти в режим редактирования HTML и, используя код `<iframe src="..."`, вставить документ непосредственно в сообщение блога.

Для вставки презентации на страницу сайта Google выполните команду **Вставка – Диск – Презентация**, где в списке документов выберите необходимую презентацию.

Добавление страниц, создание иерархической структуры

Современный веб-сайт - это набор связанных гиперссылками документов. Конечно, при более глубоком анализе поведения посетителей сайта можно заметить, что перемещения пользователя могут быть во многом случайны и более хаотичны чем это предусмотрено картой сайта: например, перемещение к разделу «Результаты» сайта inftech.spb.ru возможно не только через главную

страницу, но и с любой другой страницы через верхнее меню. Однако те страницы доступ к которым предоставляется из любой точки сайта через меню можно условно назвать «Верхний уровень».

После добавления новых страниц Google автоматически изменяет боковое меню на вашем сайте, а также создает ссылки на подстраницы. Выбор местоположения страницы осуществляется при создании новой страницы, либо в меню «Еще - Переместить страницу».



Для добавления новой страницы щелкните по кнопке «Создать страницу» и следуйте указаниям мастера.



Создание страницы на сайте Создание сайта (пример)

Название страницы:

URL страницы: /site/sozdaniesajtaprimer/ [изменить URL](#)

Выбрать шаблон ([подробнее...](#))

Веб-страница

Выберите местоположение:

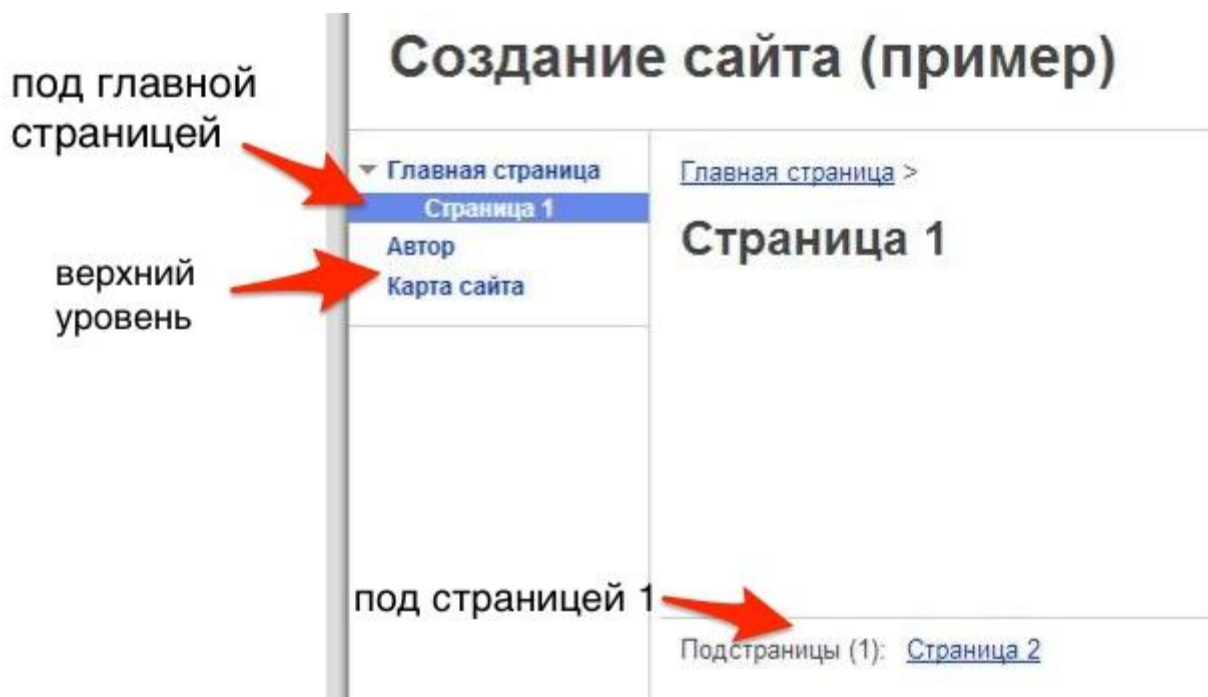
- Поместить страницу на верхний уровень
- Разместить под страницей "Главная страница"

» Новая страница

▸ [Выбрать другое местоположение](#)

1. В поле «Название страницы» введите название страницы. Выбранное название скорее всего попадет в меню, поэтому стоит воздержаться от использования длинных и непонятных названий.
2. В качестве шаблона выберите «Веб-страница».
3. В основном опираясь на карту сайта из домашнего задания №6, но помня об особенностях навигации на сайте, выберите местоположение страницы на сайте. Для наглядности ниже также представлены несколько иллюстраций:

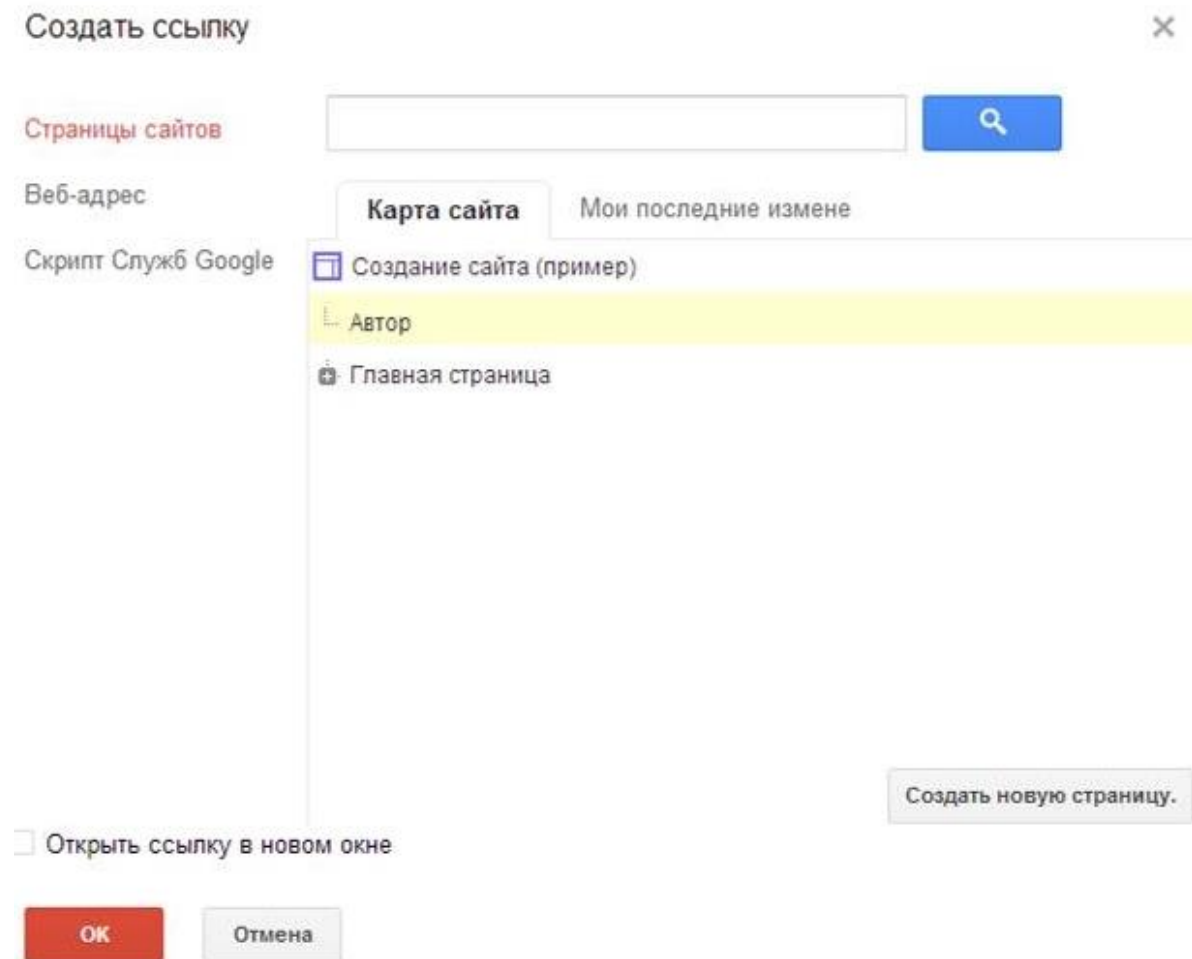
- «Поместить страницу на верхний уровень». Страница будет размещена на том же уровне, что и «Главная страница». В примере ниже это страница «Автор».
 - «Разместить под страницей «Главная страница»». Страница будет размещена вторым уровнем, под главной страницей. «Страница 1» в примере ниже.
 - «Выбрать другое расположение». Выбор данного пункта приведёт к открытию полной карты сайта с возможностью выбора любого местоположения в иерархической структуре сайта.
4. После нажатия кнопки «Создать» автоматически откроется редактор только что созданной страницы.



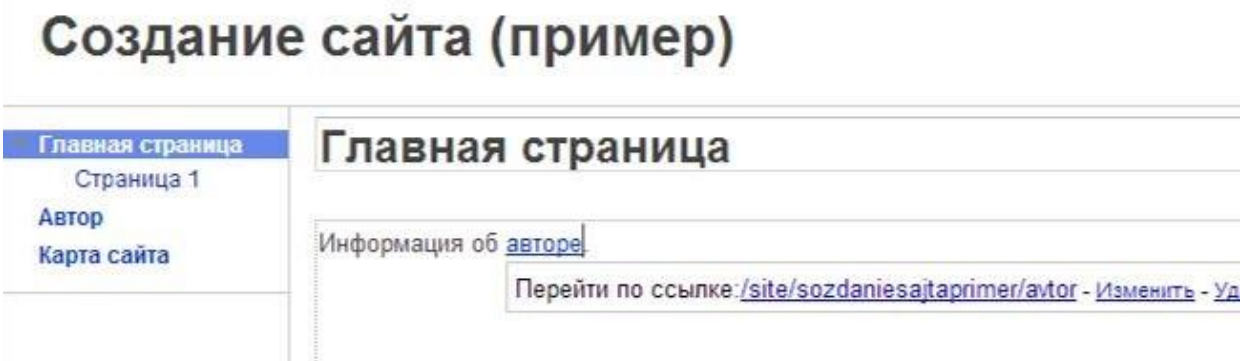


Вы также можете создать нелинейную навигацию на сайте, добавляя непосредственно в текст документа ссылки на другие страницы сайта или внешние ресурсы. Инструмент «Ссылка» работает точно также, как и в блоге, дополнительно позволяя выбрать внутренние страницы сайта из карты:

В примере ниже создается ссылка на страницу «Автор»:



Для посетителя ссылка будет работать в привычном режиме, но в редакторе она будет выглядеть следующим образом:



Расширенная версия руководства

Для ознакомления с инструментами «тонкой» настройки сайта Google перейдите к расширенной версии инструкции :

В инструкции вы сможете узнать о том, как:

- изменить местоположение страницы: например, если страница создана, но неправильно был выбран уровень ее размещения
- изменить название сайта
- изменить форматирование (фон) сайта
- изменить форматирование страниц сайта: цвет и размер заголовков страниц (в стандартном режиме редактирования страницы - форматировать

заголовок нельзя); автоматическое форматирование новых страниц сайта (например, какой размер/цвет текста будет на только что созданной странице) и так далее

- изменить параметры доступа к сайту: кто сможет видеть/читать (или редактировать) созданный сайт
- изменить настройки страницы: например, показ ссылок на подстраницы (в нижней части окна); показ приложений (файлов, прикрепленных к странице); настройка возможности оставлять комментарии
- изменить целевую страницу сайта: страницы, которая открывается первой (эта страница является главной страницей сайта)

Также в расширенной версии подробно рассмотрены базовые шаги создания сайта, описанные в краткой версии инструкции.

Требования к содержанию сайта

1. На главной странице сайта разместить ссылки на страницы с презентацией и видео-роликом и краткую информацию об авторе и назначении сайта.
2. На сайте обязательно должны быть встроены **рисунки, видео и презентация**.
3. Встроить в одну из страниц форму из л/р №6 ч II. На главную страницу сайта добавить ссылку, ведущую к странице с формой.
4. На странице "Об авторе" разместить логотип.

[Этапы создания сайта средствами Google.doc](#) (расширенная версия)

СПИСОК ЛИТЕРАТУРЫ

1. Васильев, Н.П. Технологии публикации расчетов web-обозревателей [Электронный ресурс] : учебное пособие / Н.П. Васильев, М.Л. Шилкина. — Электрон. дан. — Санкт-Петербург : СПбГЛТУ, 2008. — 48 с. — Режим доступа: <https://e.lanbook.com/book/45572> .
2. Громов, Г.Р. От гиперкниги к гипермозгу. Информационные технологии эпохи Интернета.: [текст]: Эссе, диалоги, очерки/ Г.Р. Громов. - М.: Радио и связь, 2004. - 208 с.
3. Евсеев, Д.А. Web - дизайн в примерах и задачах: [текст]: Учеб. пособие/ Д.А. Евсеев, В. В. Трофимов. - М.: Кнорус, 2010. - 272 с.
4. Евсеев, Д.А. Web - дизайн в примерах и задачах: [текст]: Учеб. пособие/ Д.А. Евсеев, В. В. Трофимов. - М.: Кнорус, 2010. - 272 с.
5. Маркин, А.В. Основы web-программирования на PHP : учебное пособие / А.В. Маркин, С.С. Шкарин. - М. : Диалог-МИФИ, 2012. - 252 с <http://biblioclub.ru/index.php?page=book&id=229742&sr=1> (ЭР)
6. Федорчук, А. Как создаются Web сайты: Кр. курс/ А. Федорчук. - СПб.: Питер, 2008. - 224 с.: ил. - Б.ц.,:
7. Федорчук, А. Как создаются Web сайты: Кр. курс/ А. Федорчук. - СПб.: Питер, 2000. - 224 с.: ил. - Б.ц.

Шульман Ирина Борисовна
ИНТЕРНЕТ ТЕХНОЛОГИИ
Методические указания
для студентов очной формы обучения
направления 09.03.01 Информатика и вычислительная техника

Редактор **Е.Ф. Изотова**

Подписано к печатиФормат 60x84 1/16.
Усл. печ. л.3,87. Тираж **20** экз. Заказ 191710. Рег. № 28.

Отпечатано в ИТО Рубцовского индустриального института
658207, Рубцовск, ул. Тракторная, 2/6.