

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Кислова Наталья Николаевна

Должность: Проректор по учебно-методической работе и качеству образования

Дата подписания: 11.09.2023

Уникальный программный ключ:

52802513f5b14a975b3e9b13008093d5726b159bf6064f865ae65b96a966c035

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

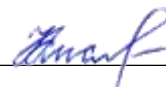
Федеральное государственное бюджетное образовательное учреждение высшего образования

«Самарский государственный социально-педагогический университет»

Кафедра информационно-коммуникационных технологий в образовании

Утверждаю

Проректор по учебно-методической  
работе и качеству образования



Н.Н. Кислова

Брыксина Ольга Федоровна  
Калинкина Марина Викторовна

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

для проведения промежуточной аттестации по дисциплине  
«Web-программирование»

Направление подготовки:

44.03.05 Педагогическое образование (с двумя профилями подготовки)

Направленность (профиль): «Экономика» и «Информатика»

Квалификация выпускника  
Бакалавр

Рассмотрено

Протокол № 3 от 25.10.2023

Заседания кафедры информационно-коммуникационных технологий в образовании

Одобрено

Начальник Управления  
образовательных программ



Н.А. Доманина

Пояснительная записка

Фонд оценочных средств (далее – ФОС) для промежуточной аттестации по дисциплине «Web-программирование» разработан в соответствии федеральным государственным образовательным стандартом высшего образования – бакалавриат по направлению подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки). Утвержден приказом Министерства образования и науки Российской Федерации от 22 февраля 2018 г. № 125, основной профессиональной образовательной программой высшего образования по направлению подготовки: 44.03.05 Педагогическое образование (с двумя профилями подготовки), направленность (профиль): «Экономика» и «Информатика», с учетом требований профессионального стандарта «Педагог (педагогическая деятельность в сфере дошкольного, начального общего, основного общего, среднего общего образования) (воспитатель, учитель)», утвержденный приказом Министерства труда и социальной защиты Российской Федерации от 18 октября 2013 г. № 544н (зарегистрирован Министерством юстиции Российской Федерации 6 декабря 2013 г., регистрационный № 30550), с изменениями, внесенными приказами Министерства труда и социальной защиты Российской Федерации от 25 декабря 2014 г. № 1115н (зарегистрирован Министерством юстиции Российской Федерации 19 февраля 2015 г., регистрационный № 36091) и от 5 августа 2016 г. № 422н (зарегистрирован Министерством юстиции Российской Федерации 23 августа 2016 г., регистрационный № 43326).

Цель ФОС для промежуточной аттестации – установление уровня сформированности компетенций (их частей):

ОПК-8 Способен осуществлять педагогическую деятельность на основе специальных научных знаний.

Задачи ФОС для промежуточной аттестации – контроль качества и уровня достижения образовательных результатов по формируемым в соответствии с учебным планом индикаторов компетенции:

ОПК-8.1. Знает: историю, теорию, закономерности и принципы построения и функционирования образовательного процесса, роль и место образования в жизни человека и общества, современное состояние научной области, соответствующей преподаваемому предмету; прикладное значение науки; специфические методы научного познания в объеме, обеспечивающем преподавание учебных предметов

Требования к процедуре оценки:

Помещение: лаборатория информационно-коммуникационных технологий

Оборудование: ноутбук с выходом в сеть Интернет

Доступ к дополнительным справочным материалам:

- Ресурсное обеспечение лабораторий кафедры ИКТ в образовании. [Электронный ресурс] - Режим доступа: <https://lms.sgspu.ru/mod/page/view.php?id=13150>

Нормы времени: на выполнение каждой лабораторной работы и итоговой работы дается 30 минут.

ФОС предоставляется обучающимся для ознакомления в начале изучения дисциплины.

## Комплект оценочных средств для проведения промежуточной аттестации

**Тема 1: Основы языка HTML****Проверяемая компетенция:**

Общепрофессиональная компетенция

ОПК-8 Способен осуществлять педагогическую деятельность на основе специальных научных знаний

**Проверяемый индикатор:**

ОПК-8.1. Знает: историю, теорию, закономерности и принципы построения и функционирования образовательного процесса, роль и место образования в жизни человека и общества, современное состояние научной области, соответствующей преподаваемому предмету; прикладное значение науки; специфические методы научного познания в объеме, обеспечивающем преподавание учебных предметов

**Проверяемые образовательные результаты:**

Знает:

эволюцию моделей гипертекста и гипермедиа от зарождения технологии до современного состояния, их возможности и ограничения, тенденции развития;

основы языка разметки гипертекста (HTML): теги, их параметры, возможности HTML по работе с мультимедиа; способы организации связей между ресурсами с помощью гипертекстовых ссылок;

**Лабораторная работа 1. Форматирование текста при помощи HTML. HTML-списки**

Общий вид Web – документов представлен приведенной ниже программой:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Документ без названия</title>

  </head>

  <body>
</body>
</html>

```

Создадим шаблон документа. Сохраним его как Web – документ и при выполнении дальнейших лабораторных работ, будем открывать его, набирать текст в контейнере тела и сохранять под другим именем.

1. Откройте “Блокнот”;

2. Наберите или скопируйте первую часть: **<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">**

3. Создайте контейнер **HTML**, для этого наберите открывающийся тег **<HTML>** и закрывающийся тег **</HTML>** (не забывайте, при этом, поставить “слэш” /). Учтите, этот контейнер содержит внутри себя несколько контейнеров, поэтому разнесите их по документу.

4. Создайте контейнер заголовка, для этого наберите теги контейнера **<head>** и **</head>**, внутри которых наберите:

1. начальный тег заголовка документа **<title>**, назовите документ и вставьте конечный тег заголовка **</title>**; укажите тип кодировки. Для этого вставьте тег **<meta>** (*закрывающегося тега нет!*) и наберите в нем фрагмент программы задания кодировки всего документа: **<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">** кодировка windows, либо **<meta http-equiv="Content-Type" content="text/html; charset=utf-8">**- кодировка utf-8;

5. Закройте контейнер **</head>**;6. Организуйте контейнер текста, для этого напишите тег **<body>** (далее идет текст документа);7. Закройте контейнеры **</body>** и **</html>**;

8. Сохраните документ. Для этого:

а. в программе “Блокнот”, кликните левой кнопкой мыши на иконке “Файл” и в открывшемся окне нажмите “Сохранить как”;

б. в открывшемся окне выберете папку, в которой будите сохранять документ, и в нижнем окне напишите название вашего файла, желательно на английском языке.

**Задание:**

1. Откройте шаблон документа (который создали в предыдущей лабораторной работе) в браузере Internet Explorer;

2. Кликните левой кнопкой мыши на иконке “Вид” в главном меню, а затем кликните на строку “Источник”. После этого откроется “Блокнот” с шаблоном документа.

3. В открывшийся “Блокнот” вставляйте по частям и форматируйте предложенный ниже текст (если есть свой текст, то лучше вставить его);



```
<hr><font color=#330000 size=1><strong> Design by ND
</strong></font>
</html>
```

### Лабораторная работа 3. «Графика и таблицы в HTML»

#### Задание:

#### Создание простейших таблиц.

Рассмотрим простейшую таблицу, которую можно представить следующим образом:

<b>Создание простейших таблиц</b>	HTML–конструкция документа может содержать множество различных тегов и параметров.
-----------------------------------	------------------------------------------------------------------------------------

Эта таблица имеет рамку, состоит из двух ячеек. В первой ячейки представлен заголовок, а во второй текст. Коды этой таблицы приведены ниже.

Пример 1.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<title>Table1</title>
</head>
<body>
<table border="1" width="100%">
<tr>
<th width="50%">Создание простейших таблиц</th>
<td width="50%">HTML–конструкция документа может содержать множество
различных тегов и параметров.</td>
</tr>
</table>
</body>
</html>
```

#### Ход работы

- Откройте шаблон документа (который создали лабораторной работе 1) в браузере Internet Explorer;
- Кликните левой кнопкой мыши на иконке “Вид” в главном меню,
- Кликните на строку “Источник”. После этого откроется “Блокнот” с шаблоном документа (document’s form.htm) .
- Между тегами <title>...</title> напишите название таблицы;
- В шаблоне документа между тегами <body> .... </body> наберите предложенную выше программу создания простейших таблиц.
- Сохраните программу под именем table.htm . При этом учтите, что всегда имена папок должны:
  - называться с использованием английского алфавита и арабских цифр;
  - не должны содержать следующие символы:
    - пробел;
    - табуляцию;
    - новая строка;
    - возврат каретки;
    - символы.
- Изменяя толщину рамки ( border =”1”) от 0 до 3, сохраните Web-страницу, в зависимости от толщины внешней рамки: table0.htm; table1.html; table2.htm; table3.htm.

#### Задание (создание сложных таблиц)

Создать таблицу следующего вида:

ячейка 1	ячейка 2	ячейка 3
ячейка 4	ячейка 5	ячейка 6
ячейка 7	ячейка 8	ячейка 9

Для этого:

- Откройте шаблон документа (document’s form.htm) в браузере Internet Explorer;
- Кликните левой кнопкой мыши на иконке “Вид” в главном меню,
- Кликните на строку “Источник”. После этого откроется “Блокнот” с шаблоном документа.
- Между тегами <title>...</title> напишите название таблицы;
- В шаблоне документа между тегами <body> .... </body> наберите предложенную ниже программу создания сложной таблицы.
- В комментарии <!-- Далее продолжить составление таблицы самостоятельно--!> предлагается самостоятельно написать теги для создания следующих столбцов таблицы.
- Сохраните программу под именем table6.htm .

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title> Создание сложных таблиц</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
</head>
<table border="1" width=900 cellspacing="0" cellpadding="0" align="center">
<tr>
<th width=300> ячейка 1</th>
<td width=300> ячейка 2</td>
<td width=300> ячейка 3</td>
</tr>
<!-- Дальше продолжить составление таблицы самостоятельно--!>
</table>
</ body >
</html>

```

### Задание 3.3. Параметры внешних и внутренних границ

1. Откройте шаблон документа (document's form.htm ) в браузере Internet Explorer;
2. Кликните левой кнопкой мыши на иконке “Вид” в главном меню,
3. Кликните на строку “Источник”. После этого откроется “Блокнот” с шаблоном документа.
4. Между тегами <title>...</title> напишите название таблицы;
5. В шаблоне документа между тегами <body> .... </body> наберите предложенную ниже программу создания таблицы с различными параметрами **внешних и внутренних границ**.

Обратите внимание на общее построение программы.

Программа создания таблицы с различными параметрами **внешних и внутренних границ**

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title> FRAME</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
</head>
<body>

<table border=0 width=40%>
<caption align="left">
<strong>Возможные значения атрибута frame</strong>
</caption><br> <br>
<tr>
<td><table border=4 frame="border">
<tr><td width="49">frame1</td>
<td width="9">=</td>
<td width="55">border</td>
</tr></table><br>

<table align="left" border="1" cellspacing="0" cellpadding="4" width="14%" frame="box" rules="groups">
<tr><td width="7%">frame2</td>
<td width="3%">=</td>
<td width="90%">box</td>
</tr></table><br>
<br>
<table align="left" border="1" cellspacing="0" cellpadding="4" width="14%" frame="above"
rules="all"><tr><td>frame3</td><td>=</td><td>above</td>
</tr></table> <br>
<br>

<table align="left" border="1" cellspacing="0" cellpadding="4" width="169" frame="void" rules="none">
<tr><td width="49">frame4</td>
<td width="9" >=</td>
<td width="79">void</td>
</tr></table><br>
<br>

```

```

<table align="left" border="1" cellspacing="0" cellpadding="4" width="14%" frame="hsides"
rules="all"><tr><td>frame5</td></tr></table><br> <br>
<table align="left" border="1" cellspacing="0" cellpadding="4" width="14%" frame="vsides"
rules="all"><tr><td>frame6</td><td class="eq"></td></tr></table><br><br>
<table align="left" border="1" cellspacing="0" cellpadding="4" width="14%" frame="lhs"
rules="all"><tr><td>frame7</td><td class="eq"></td></tr></table><br> <br>
<table align="left" border="1" cellspacing="0" cellpadding="4" width="14%" frame="rhs"
rules="all"><tr><td>frame8</td><td class="eq"></td></tr></table>
</body>
</HTML>

```

6. Сохраните программу под именем table7.htm

Критерии оценивания лабораторных работ:  
5 баллов, если:

- отчёт полностью отражает основные положения использования тегов гипертекстовой разметки;
- обучающийся знает основные теги гипертекстовой разметки и умеет их применять.

**Итоговое задание (разработка проекта сайта произвольной тематики)**

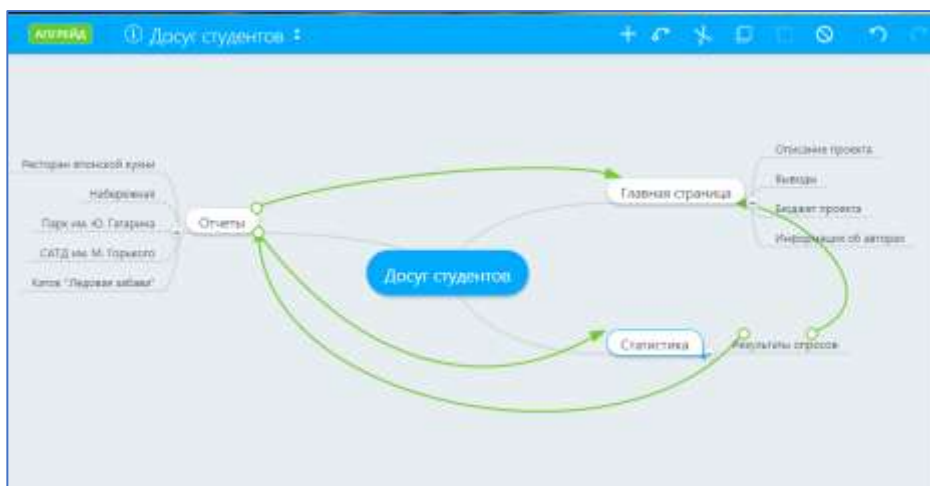
Данная работа включает в себя три задания:

- разработка иерархической структуры сайта по выбранной тематике (4 балла);
- разработка контента по выбранной тематике для наполнения сайт (4 балла);
- подбор соответствующих иллюстраций для сформированного ранее контента сайта (2 балла).

Критерии оценивания:

- структура сайта содержит, как минимум, 4 страницы;
- все страницы сайта должны быть связаны гиперссылками;
- на главной странице расположено меню навигации;
- организован переход с любой страницы сайта на главную;
- контент соответствует выбранной тематике;
- материалы сайта в полной мере раскрывают тему проекта;
- на страницах сайта представлен контент различного типа (текст, статические и динамические изображения);
- представленный контент характеризуется высоким качеством;
- изображения соответствуют содержанию страниц сайта;
- параметры графических файлов соответствуют требованиям размещения изображений на Web-страницах.

Результат разработки иерархической структуры сайта представляется в виде блок-схемы или ментальной карты с указанием связей и переходов. Пример работы:



Контент и иллюстрации, соответствующие тематике проекта размещаются на Web-страницах, между которыми организуются переходы по гиперссылкам.

## Тема 2. Каскадные таблицы стилей (CSS). Принципы веб-дизайна

### Проверяемая компетенция:

Общепрофессиональная компетенция

ОПК-8 Способен осуществлять педагогическую деятельность на основе специальных научных знаний

### Проверяемый индикатор:

ОПК-8.1. Знает: историю, теорию, закономерности и принципы построения и функционирования образовательного процесса, роль и место образования в жизни человека и общества, современное состояние научной области, соответствующей преподаваемому предмету; прикладное значение науки; специфические методы научного познания в объеме, обеспечивающем преподавание учебных предметов

### Проверяемые образовательные результаты:

Знает:

основы языка разметки гипертекста (HTML): теги, их параметры, возможности HTML по работе с мультимедиа; способы организации связей между ресурсами с помощью гипертекстовых ссылок;

основные понятия и принципы веб-дизайна;

основы синтаксиса языка JavaScript, каскадные таблицы стилей CSS, способы создания JavaScript-мультипликации и графических меню; интерактивных клиент-серверных Web-приложений;

различные компромиссные решения в отношении производительности и безопасности Web-приложений;

### Практическая работа 1. Создание заголовков разного уровня

#### Порядок выполнения работы

1. Откройте **Paint**.

2. В **Paint** создайте рисунок размером 100x100 точек и сохраните его в папку *D:\Users\...\Web\Lab10\* в формате *JPG* под именем *bgpic*.

3. В **Paint** создайте второй рисунок размером 200x400 точек и сохраните его в папку *D:\Users\...\Web\Lab10\* в формате *JPG* под именем *risunok*.

4. Откройте **Блокнот**.

5. Напишите следующий код:

```
body {background-color: lightsteelblue;
background-image: url(bgpic.jpg)}
P {font: 14pt "Times Cyr", serif}
P {text-indent: 20}
H1 {color: red}
H2 {color: blue}
```

Сохраните файл в папку *D:\Users\...\Web\Lab10\* с именем *mycss.css*.

6. Создайте *HTML*-документ, в нём создайте:

- заголовок первого уровня "Краткая информация";
- абзац, в котором напишите несколько строк о себе;
- горизонтальную линию;
- заголовок второго уровня "Учёба";
- абзац, в котором напишите название факультета, направление подготовки, группу;
- горизонтальную линию;
- надпись "Контактная информация", оформите её как гиперссылку на документ *kontakt.html*.

Сохраните файл в папку *D:\Users\...\Web\Lab10\* с именем *index10.html*.

7. Подключите к *HTML*-документу созданную таблицу стилей (*mycss.css*).

8. Используя внедрённую таблицу стилей, измените положение и повторяемость фонового рисунка, расстояние между буквами в заголовке второго уровня, интервал между строками в абзаце, выравнивание линий.

9. Создайте в **Блокноте** новый *HTML*-документ, в нём:

- создайте заголовок первого уровня "Контактная информация";
- создайте список, содержащий *e-mail*, адрес, телефон;
- вставьте рисунок *risunok*, сделайте чтобы текст обтекал его с левой стороны;
- подключите созданную таблицу стилей;
- используя внедрённую таблицу стилей измените цвет текста в абзаце.

Сохраните файл в папку *D:\Users\...\Web\Lab10\* с именем *kontakt.html*.

### Практическая работа 2. Разработка индивидуального стиля для сайта с применением каскадных таблиц стилей

Критерии:

- создан единый файл CSS-стиля для всех Web-страниц сайта;
- в файле стиля прописано форматирование для всех видов текстовых блоков;
- в файле стиле прописано форматирование для вкладок сайта;



Результатом выполнения данного задания является создание файла стиля для страниц сайта, разработанного в рамках изучения темы № 1 (Язык HTML как средство создания информационных ресурсов Интернет).

Пример выполненной работы (содержание стилевого файла):

```
H1,H2,H3,H4
{
  text-align:left;
  color:maroon;
  font-family:Times New Roman;
}

P {
  text-indent:20px;
  text-align:justify;
  font-family:Times New Roman;}

td
{font-family:Times New Roman;
text-align:center;
bordercolor:white;}

LI {
text-align:justify;}
IMG {border:0;}

table
{bordercolor:white;
background-color:white;
font-family:Times New Roman;
border:0;
}

a
{color:black;
text-decoration:none;
font-family:Times New Roman;
}

img
{width:255;
height=:191;}

img.title
{width:500;
height=:400;}

td.activ
{background-color:white;
width="267"}

td.notactiv
{background-color:#FCE2B4;
width="267"}
```

Критерии:

- отчёт полностью отражает основные положения работы с CSS (2 балла);
- обучающийся чётко и ясно объясняет способы связывания документов с таблицами стилей, умеет их применять (2 балла);
- создан единый файл CSS-стиля для всех Web-страниц сайта (2 балла);
- в файле стиля прописано форматирование для всех видов текстовых блоков (2 балла);
- в файле стиле прописано форматирование для вкладок сайта (2 балла).

### Тема 3. Язык JavaScript как средство создания интерактивных ресурсов

#### Проверяемая компетенция:

Общепрофессиональная компетенция

ОПК-8 Способен осуществлять педагогическую деятельность на основе специальных научных знаний

#### Проверяемый индикатор:

ОПК-8.1. Знает: историю, теорию, закономерности и принципы построения и функционирования образовательного процесса, роль и место образования в жизни человека и общества, современное состояние научной области, соответствующей преподаваемому предмету; прикладное значение науки; специфические методы научного познания в объеме, обеспечивающем преподавание учебных предметов

#### Проверяемые образовательные результаты:

Знает:

основы синтаксиса языка JavaScript, каскадные таблицы стилей CSS, способы создания JavaScript-мультипликации и графических меню; интерактивных клиент-серверных Web-приложений;

различные компромиссные решения в отношении производительности и безопасности Web-приложений;

место и значимость курса в плане формирования технологической компетентности школьников и профессиональной ориентации.

#### Лабораторная работа 1. Основы языка JavaScript

**Задание 1:** Создать приведенный пример документа в любом редакторе. Удобнее использовать редактор Visual Studio или любой другой, поддерживающий языки разметки. Сохранить файл с названием Пример1\_1.html и открыть (запустить) его в любом браузере.

Пример кода в составе страницы HTML:

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p>
```

JavaScript может написать прямо в HTML выходной поток – в теле документа

```
</p>
```

```
<script>
```

```
document.write("<h1>Это тег для заголовка</h1>");
```

```
document.write("<p>Это тег для обозначения параграфа</p>");
```

```
</script>
```

```
<p>
```

Вы можете использовать метод `<strong> document.write </strong>` в теле выходном HTML.

Если вы используете этот метод после загрузки документа (например, в функции), весь документ будет перезаписан.

```
<!--тег <strong> делает выделения текста на выходе страницы -->
```

```
</p>
```

```
</body>
```

```
</html>
```

**Задание 2.** Создать приведенный пример документа в любом редакторе. Удобнее использовать редактор Visual Studio или любой другой, поддерживающий языки разметки. Сохранить файл с названием Пример1\_2.html и открыть (запустить) его в любом браузере.

Для успешного решения широкого круга задач требуется многократно повторить некоторую последовательность действий, записанную в программе один раз. В том случае, когда число повторений последовательности действий нам неизвестно, либо число повторений зависит от некоторых условий, можно воспользоваться оператором цикла вида:

```
while (B) {s}
```

где B - выражение логического типа; s - операторы, называемые телом цикла. Операторы s в фигурных скобках выполняются до тех пор, пока условие B не станет ложным.

Пример 1. Нахождение общего делителя

Напишем программу, которая для двух заданных чисел определяет наибольший общий делитель.

При решении задачи воспользуемся алгоритмом Евклида. Если значение m равно нулю, то наибольший общий делитель чисел n и m равен n:

НОД(n, 0) = n.

В остальных случаях верно следующее соотношение:

$\text{НОД}(n, m) = \text{НОД}(m, n \% m)$ .

В функции `nod` переменная `p` используется для получения остатка от деления чисел `n` и `m` (листинг 1). Выполнение цикла продолжается до тех пор, пока значение `p` не станет равным нулю. Последнее вычисленное значение `m` равно наибольшему общему делителю.

**Листинг 1:**

```
<HTML>
<HEAD>
<TITLE>Наибольший общий делитель двух чисел</TITLE>
<script language="JavaScript">
<!-- //
function nod(obj)
{ var n=obj.num1.value
var m=obj.num2.value
var p = n%m
while (p!=0)
{ n=m
m=p
p=n%m
}
obj.res.value=m
}
//-->
</script>
</HEAD>
<BODY>
Наибольший общий делитель двух заданных чисел
<FORM name="form1">
Введите число <input type="text" name="num1" size="8"><br>
Введите число <input type="text" name="num2" size="8"><br>
<input type="button" value="Вычислить" onClick="nod(form1)"><br>
Наибольший общий делитель <input type="text" name="res"
size="8"><hr>
<input type="reset" value="Отменить">
</FORM>

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

Если число повторений заранее известно, то можно воспользоваться следующим оператором цикла, который часто называют оператором цикла арифметического типа. Синтаксис этого оператора таков:

```
for (A; B; I){S}
```

Выражение `A` служит для инициализации параметра цикла и вычисляется один раз в начале выполнения цикла. Выражение `B` (условие продолжения) управляет работой цикла. Если значение выражения ложно, то выполнение цикла завершается, если истинно, то выполняется оператор `S`, составляющий тело цикла. Выражение `I` служит для изменения значения параметра цикла. После выполнения тела цикла `S` вычисляется значение выражения `I`, затем опять вычисляется значение выражения `B` и т.д. Цикл может прекратить свою работу в результате выполнения оператора `break` в теле цикла.

**Задания:**

1. Напишите программу, которая "переворачивает" заданное натуральное число.
2. Напишите сценарий, в котором определяется количество "счастливых" шестизначных автобусных билетов, т. е. таких, в номерах которых сумма первых трех цифр равна сумме трех последних.
3. Напишите программу, определяющую все делители заданного натурального числа.

**Лабораторная работа 2. Функции в JavaScript**

Объявление функций.

1. С помощью текстового редактора "Блокнот" (Notepad) создайте страницу `testjs.html` и поместите в начало страницы следующий код:

```
<html>
<head>
<title>Script test</title>
</head>
<body>
```

```

<script language="JavaScript">
document.write("Добро пожаловать на мою страницу!<br>");
document.write("Это JavaScript!<br>");
document.write("Добро пожаловать на мою страницу!<br>");
document.write("Это JavaScript!<br>");
document.write("Добро пожаловать на мою страницу!<br>");
document.write("Это JavaScript!<br>");
</script>
</body>
</html>

```

2. Сохраните страницу и просмотрите ее с помощью браузера. Такой скрипт выведет на экран следующий текст:

```

Добро пожаловать на мою страницу!
    Это JavaScript!

```

три раза. Наверняка вы заметили, что для получения необходимого результата определенная часть его кода была повторена три раза. Это неэффективно, поэтому попробуем оптимизировать код.

3. Определите функцию `welcome_func()`, которая будет выводить на экран строки:

```

Добро пожаловать на мою страницу!
    Это JavaScript!

```

4. Затем вызовите функцию несколько раз. Для этого вы можете использовать цикл:

```

for(i=0; i<3; i++){
welcome_func();
}

```

5. Функции могут также использоваться в совместно с процедурами обработки событий. Рассмотрим пример. Создайте страницу `testjs2.html`, введите следующий код:

```

<html>
<head>
<script language="JavaScript">
function calculation() {
var x = 12;
var y = 5;
var result = x + y;
window.status = result;
}

```

```

</script>

```

```

</head>

```

```

<body>

```

```

<form>

```

```

<input type="text" onFocus="calculation()">

```

```

</form>

```

```

</body>

```

```

</html>

```

В приведенном примере интерес представляют следующие нововведения:

`var x = 12;` — так в JavaScript объявляются переменные, тип переменной не дифференцируется — одинаково объявляются переменные типа `double`, `integer`, `String` и прочие;

`window.status = result;` — свойству `window.status` присваивается новое значение, в нашем случае это строка 17, эта строка будет отображена в статус-панели браузера.

`onFocus` — это команда, которая вызывает действие, когда пользователь "фокусируется" на элементе страницы, в нашем случае — помещает курсор в текстовое поле.

6. Сохраните файл.

Для просмотра созданной Web-страницы используйте браузер Microsoft Internet Explorer.

Вы должны увидеть Web-страницу с определенным в ее верхней части окном редактирования, установка курсора в этом окне приведет к появлению строки 17 в статус-панели браузера.

На этой же странице создайте функцию, которая будет вызываться при потере фокуса полем редактирования и помещать в статус-панель результат, полученный от разности чисел `x` и `y`. Обработчику события "потеря фокуса" в JavaScript соответствует параметр `onBlur`.

7. Продемонстрируйте полученный документ преподавателю.

### Лабораторная работа 3. Объекты в JavaScript. Реализация форм

С помощью текстового редактора "Блокнот" (Notepad) создайте страницу `testform.html` и поместите на странице следующий код:

```

<head>
<title>Управление объектами формы</title>
</head>
<body bgcolor=#ffffff>



<p>

<form name="myForm">
Ваше имя:
<br>
<input type="text" name="name" value=""><br>
e-mail:
<br>
<input type="text" name="email" value=""><br><br>
<br>
<input type="button" value="Нажмите!" name="myButton">
</form>

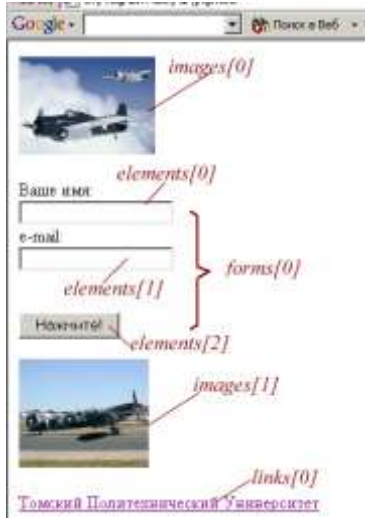
<p>


<p>
<a href="http://www.tpu.ru/">Томский Политехнический Университет</a>

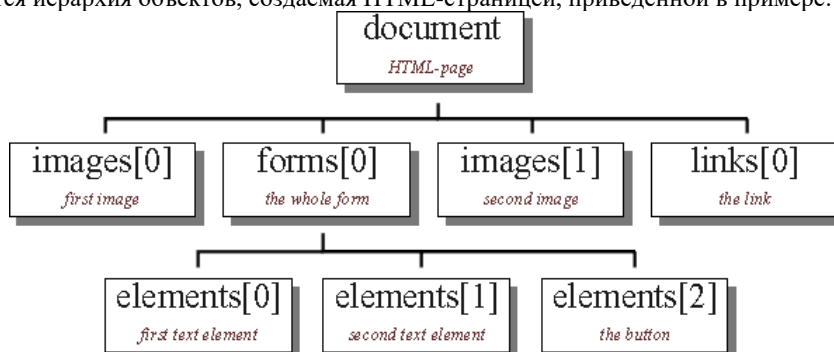
</body>
</html>
    
```

Для самопроверки:

Вот так должна выглядеть Ваша страница:



На странице размещено два рисунка, одна ссылка и форма с двумя полями для ввода текста и одной кнопкой. Как уже отмечалось выше, с точки зрения языка JavaScript, загруженная в настоящий момент html-страница, представляет собой объект *document*. Через этот объект можно получить доступ ко всем элементам, размещенным на странице. На следующем рисунке иллюстрируется иерархия объектов, создаваемая HTML-страницей, приведенной в примере:



Создадим скрипт, который позволит заполнять текстовые поля формы predetermined значениями.

В разделе <head> ... </head> разместите следующий код:

```
<script>
function fillIn(){
document.forms[0].elements[0].value = "Евгений";
document.forms[0].elements[1].value = "vest@list.ru";
}
</script>
```

Привяжите запуск функции к событию *onClick* кнопки "Нажмите!".

Сохраните страницу и просмотрите ее в окне браузера. При правильной работе скрипта по нажатию кнопки "Нажмите!" текстовые поля должны заполняться значениями.

Так как все элементы на странице имеют свои уникальные имена (обратите внимание на параметр *name* в каждом элементе), в скрипте их можно вызывать не по индексам, а по именам в соответствующих массивах:

```
<script>
function fillIn(){
document.forms['myForm'].elements['name'].value = "Евгений";
document.forms['myForm'].elements['email'].value = "vest@list.ru";
}
</script>
```

или даже так:

```
<script>
function fillIn(){
document.myForm.name.value = "Евгений";
document.myForm.email.value = "vest@list.ru";
}
</script>
```

Продемонстрируйте полученный документ преподавателю.

Критерии оценивания выполнения лабораторных работ:

5 баллов, если:

- отчёт полностью отражает основные положения работы с JavaScript;
- обучающийся демонстрирует на конкретном примере знание функций в JavaScript;

### Итоговое задание

#### Проверяемая компетенция:

Общепрофессиональная компетенция

ОПК-8 Способен осуществлять педагогическую деятельность на основе специальных научных знаний

#### Проверяемый индикатор:

ОПК-8.1. Знает: историю, теорию, закономерности и принципы построения и функционирования образовательного процесса, роль и место образования в жизни человека и общества, современное состояние научной области, соответствующей преподаваемому предмету; прикладное значение науки; специфические методы научного познания в объеме, обеспечивающем преподавание учебных предметов

Знает:

место и значимость курса в плане формирования технологической компетентности школьников и профессиональной ориентации.

Приведите примеры тем для организации учебно-исследовательской деятельности обучающихся в рамках тем данного курса:

<i>Тема</i>	<i>Проблемы для проведения учебного исследования</i>	<i>Прогнозируемые продукты и виды деятельности</i>	<i>Интернет-ресурсы для сопровождения учебно-исследовательской деятельности</i>

#### Оценочный лист к заданию 3.

Каждое требование оценивается по шкале:

3 – задание выполнено правильно полностью;

2 – задание выполнено с незначительными ошибками;

1 – задание выполнено большей частью с ошибками, недочетами);

0 – задание не выполнено или выполнено с существенными ошибками.

Критерий	Количество баллов
содержательный отбор проблем способствует повышению мотивации включения обучающихся в учебно-исследовательскую деятельность;	
рекомендуемые темы и виды деятельности способствуют формированию интереса обучающихся к организации научного исследования, расширения кругозора и т.п.;	
организация учебно-исследовательской деятельности сопровождается созданием информационного продукта;	
выбраны адекватные продукту инструменты деятельности (средства и сервисы ИКТ);	
проведен качественный отбор Интернет-ресурсов для содержательного изучения проблемы в ходе исследования;	
ресурсы содержат материалы, доступные по восприятию для целевой аудитории и способствуют профессиональному самоопределению школьников;	
показана роль средств ИКТ, мобильных приложений для обеспечения поисковой, исследовательской и др. познавательной деятельности;	
задания ориентированы на включение обучающихся в исследовательскую деятельность;	

Максимальное количество баллов: 24

## Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации

Код контролируемой компетенции (или ее части)	Наименование оценочного средства	Максимальное количество баллов	Уровень освоения компетенцией (в баллах)		
			Пороговый (56-70%)	Продвинутый (71-85%)	Высокий (86-100%)
ОПК-8.1.	Тема 1. Лабораторные работы 1-3	15	8-10	11-13	14-15
	Тема 1. Итоговое задание	10	5-6	7-8	9-10
	Тема 2. Практические работы 1, 2	10	5-6	7-8	9-10
	Тема 3. Лабораторные работы 1-3	15	8-10	11-13	14-15
	Итоговое задание	24	13-17	18-21	22-24
<b>Итого за семестр:</b>		74	42-52	53-63	64-74