

Лабораторная работа № 3 Работа с JavaScript. Размещение JavaScript на HTML странице

Тема: Работа с JavaScript. Размещение JavaScript на HTML странице

Цель занятия: Ознакомить с основами языка JavaScript

Задания:

- 1 Ознакомьтесь с теоретическими аспектами темы.
- 2 Создайте простую веб-страницу с использованием JavaScript согласно методическим указаниям.
- 3 Создайте веб-страницу с формой и кнопкой на основе JavaScript согласно методическим указаниям.
- 4 Создайте интерактивную страницу с формой и диалогом, производящую расчёты для определения стоимости или физического объёма ваших товаров(услуг) согласно таблице вариантов и подключите её по ссылкам к созданным ранее страницам.
- 5 Составьте отчёт о выполненной работе со скриншотами этапов работы и пояснениями. Письменно ответьте в отчёте на контрольные вопросы.
- 6 Сдайте на проверку отчёт и zip – архив с созданными html страницами.

Необходимые приборы: ПК, текстовый редактор Блокнот, браузер

Методические рекомендации к практическому занятию

1. Введение

Язык программирования JavaScript был разработан Бренданом Эйком (Brendan Eich) в Netscape Communications как язык сценариев для обозревателей Netscape Navigator, начиная с версии 2.0. В дальнейшем к развитию этого языка подключилась корпорация Microsoft, чьи обозреватели Internet Explorer поддерживают JavaScript, начиная с версии 3.0. Версия Microsoft получила название JScript, поскольку JavaScript является зарегистрированной маркой фирмы Netscape.

Код скрипта JavaScript размещается непосредственно на HTML-странице. Все, что стоит между тэгами `<script>` и `</script>`, интерпретируется как код на языке JavaScript. Инструкция `document.write()` - одна из наиболее важных команд, используемых при программировании на языке JavaScript. Команда `document.write()` используется, когда необходимо что-либо написать в текущем документе (в данном случае таким является наш HTML-документ).

События и обработчики событий являются очень важной частью для программирования на языке JavaScript. События, главным образом, инициируются теми или иными действиями пользователя. Если он щелкает по некоторой кнопке, происходит событие *"Click"*. Если указатель мыши пересекает какую-либо ссылку гипертекста - происходит событие *MouseOver*. Существует несколько различных типов событий. Мы можем заставить нашу JavaScript-программу реагировать на некоторые из них. И это может быть выполнено с помощью специальных программ обработки событий. Так, в результате щелчка по кнопке может создаваться выпадающее окно. Это означает, что создание окна должно быть реакцией на событие щелчка - *Click*. Программа - обработчик событий, которую мы должны использовать в данном случае, называется *onClick*. И она сообщает компьютеру, что нужно делать, если произойдет данное событие.

Вы можете использовать в скрипте множество различных типов функций обработки событий. В большинстве случаев функции представляют собой лишь способ связать вместе нескольких команд. Функции могут также использоваться совместно с процедурами обработки событий.

JavaScript - это объектно-ориентированный язык программирования (ООП), основан не на обработке команд кода, а на присвоении отдельным элементам программы конкретных событий и выполнении их, если данное событие имело место. Например, событие нажатие на кнопку приводит к изменению содержимого текстового поля:

Пример №1

Введите свое имя и нажмите кнопку

Основными понятиями любого объектно-ориентированного языка являются объекты, классы, методы и свойства. Разберём основные понятия на конкретных примерах:

```
<script type="text/javascript">
document.write("Введите свое имя и нажмите кнопку")
</script>
```

В результате при просмотре данной страницы в браузере появится текст: "Введите свое имя и нажмите кнопку".

ЗАДАНИЕ 1

1. Запустите Notepad++ . Установите кодировку UTF-8, Синтаксисы – Н – HTML .

2. Введите текст

```
<html>
```

```
<body>
```

```
<br>
```

Это обычный HTML документ.

```
<br>
```

```
<script language="JavaScript">
```

```
document.write("А это JavaScript!")
```

```
</script>
```

```
<br>
```

Вновь документ HTML.

```
</body>
```

```
</html>
```

3. Сохраните документ в формате html

4. Запустите страницу в окне браузера.

Результат выполнения файла в случае, если используемый браузер поддерживает JavaScript:

Это обычный HTML документ.
А это JavaScript!

Если браузер не поддерживает JavaScript, то он проигнорирует тег <script>. В этом случае измените исходный текст:

```
<html>
```

```
<body>
```

```
<br>
```

Это обычный HTML документ.

```
<br>
```

```
<script language="JavaScript">
```

```
<!-- hide from old browsers
```

```

    document.write("Аэто JavaScript!")
// -->
</script><br>
Вновь документ HTML.
</body>
</html>

```

В этом случае использован тег комментария из HTML - `<!-- -->`. В результате новый вариант нашего исходного кода будет выглядеть как:

```

    Это обычный HTML документ.
    Вновь документ HTML.

```

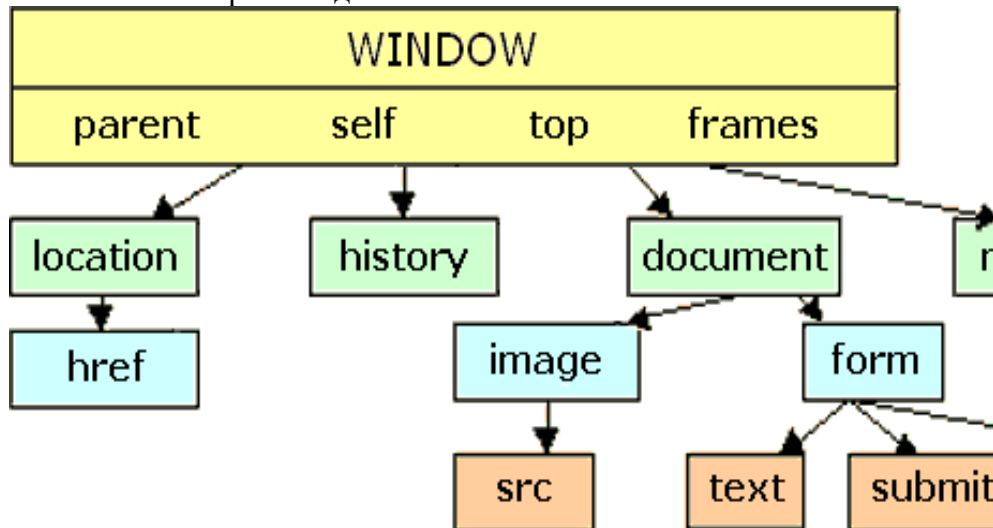
2. Основные понятия JavaScript: объект, метод, свойства, события.

ОБЪЕКТ (*объект*) - это то, с чем производится действие, событие. Это может быть документ, открываемый в окне браузера или само окно браузера, или какая-то часть документа, теги. Объект должен иметь уникальное имя (ID), чтобы к нему можно было обратиться.

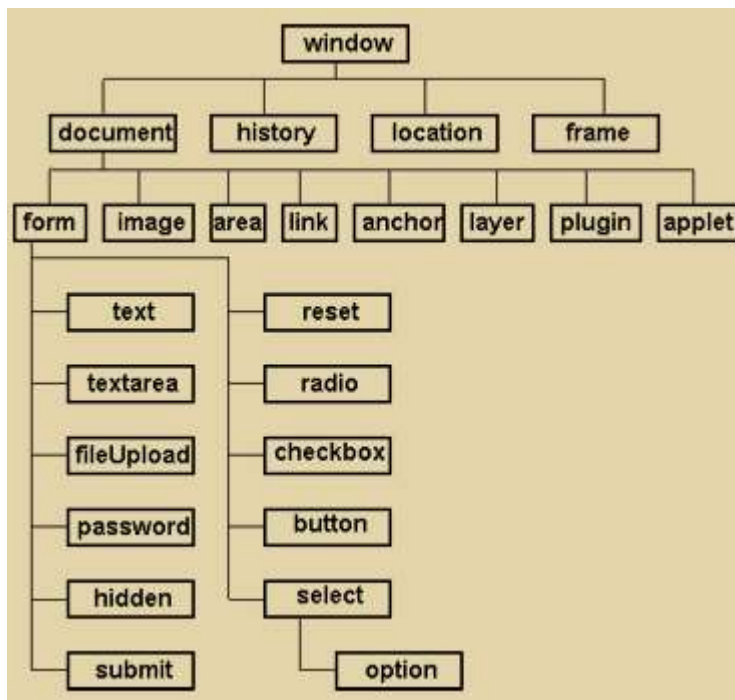
В нашем случае объектом является документ HTML и к нему можно просто обратиться по имени: **document**.

Иерархия объектов в JavaScript

В языке JavaScript определены объекты, которые называются объектами браузера. Каждый объект соответствует некоторому элементу Web-страницы: окну, документу, изображению, ссылке и т.п. Управлять частями документа можно с помощью методов браузера. Объекты браузера имеют иерархическую структуру. На самом верхнем уровне располагается объект window. Он является родителем остальных объектов. Объект window представляет окно браузера. Свойство window.status можно использовать для изменения вида строки состояния. Для отображения диалоговых сообщений объект window имеет три метода.



Иерархическая структура объектов



другие объекты JavaScript

Каждый объект обладает своими методами.

METHOD (*метод объекта*) - это действия, которые можно выполнять над объектом такого типа, или которые сам объект может выполнять.

Синтаксис кода: между именем объект и методом обязательно ставят разделительный оператор точка, после метода в скобках параметры метода.

Объект . Метод ("параметры
метода")

Параметры метода относятся к типу данных - строки символов. Строки символов нужно обязательно взять в кавычки либо в одинарные, либо в двойные.

Каждый объект обладает своими свойствами.

PROPERTY (свойство) - каждый объект имеет свои свойства. Один и тот же объект может обладать многими свойствами. Часто эти свойства необходимо изменить, при возникновении некоторого события.

Для изменения свойства объекта необходимо соблюдать следующий синтаксис:

Объект . свойство = "новое значение
объекта свойства"

Например, для изменения фонового цвета документа HTML (имя данного свойства bgColor) следует написать следующее:

```

<script type="text/javascript">
document.bgColor='gold'
</script>
  
```

И при просмотре в окне браузера фоновый цвет HTML документа будет золотым. Коды цветов можно посмотреть в [Википедии](#)

Обратите внимание на то, что значение свойства gold пишется в кавычках (одинарных или двойных), т.к. значение свойства относится к типу данных строки символов.

Задание №2

Добавьте в созданный ранее файл страницы скрипт, задающий свойство документа - фон красного цвета

Разумеется, нас будет интересовать возможность изменения свойства при возникновении какого-либо события.

EVENT (событие) - это все, что случилось: открытие окна, загрузка в него документа, клик клавишей мышки или просто перемещение курсора по экрану, нажатие клавиши на клавиатуре - это все события, и они могут инициировать запуск больших и маленьких программ.

Стандартные события в HTML

имя события	происходит
onclick	при щелчке кнопки мыши на элементе
ondblclick	при двойном щелчке кнопки мыши на элементе
onmousedown	при нажатии кнопки мыши на элементе
onmouseup	при отпускании кнопки мыши на элементе
onmouseover	при попадании курсора мыши на элемент
onmousemove	при движении курсора мыши по элементу
onmouseout	при попадании курсора мыши за пределы элемента
onkeypress	при нажатии и отпускании клавиши на элементе
onkeydown	при нажатии клавиши на элементе
onkeyup	при отпускании клавиши на элементе

Здесь следует пояснить, что события (event) и обработчики событий (event handler) относятся к JavaScript, но они скорее «встроены» в HTML-код. Они входят в структуру документа HTML и не требуют тегов `<script>` и `</script>`. Среди разнообразных обработчиков событий для начала мы выберем один, самый популярный, — **onmouseover** (навести мышь).

Пример №2

...пример смотрите здесь - [в другом окне](#)

Код выглядит следующим образом:

```
<p onmouseover="document.bgColor ='red'">Наведи мышь на этот текст .... </p>
```

Как уже говорилось для написания этого кода не требуются теги `<script>` `</script>`. Событие встраивается в HTML код, т.е является описанием, атрибутом тега (в данном случае тега `<p></p>`) при выполнении данного события - наведении мышкой на текст данного абзаца - изменяется свойство объекта - фон документа HTML.

И здесь есть еще одна важная особенность: **document.bgColor ='red'** нужно также записать в кавычках - одинарных или двойных. Вы можете использовать любой тип кавычек. Однако если Вы вынуждены как в данном случае ставить кавычки дважды, то можно использовать только вложенные кавычки. Не имеет значения, в каком порядке Вы использовали кавычки - сначала двойные, а затем одинарные или наоборот.

МОЖНО:

```
onmouseover="document.bgColor      ='red'           "           или
onmouseover='document.bgColor ="red" '
```

Но НЕЛЬЗЯ:

```
onmouseover="document.bgColor      ="red"           "
onmouseover='document.bgColor      ='red'           '
onmouseover="document.bgColor ='red" '
```

Если значение HTML-атрибута обработчика события состоит из нескольких JavaScript-инструкций, они должны отделяться точками с запятой либо значение атрибута должно располагаться в нескольких строках.

Задание №3

Измените скрипт, созданный ранее так, чтобы - фон документа при наведении курсором на какой-то текст менял цвет, а при уходе с него курсора возвращался к первоначальному.

А если мы хотим изменить не свойство всего документа, а только свойство какого-то абзаца? Как в данном случае мы можем изменить свойство данного объекта? Есть несколько способов.

ПРИМЕР

`<p style="color:blue" onmouseover="this.style.color='red'">Этот абзац меняет цвет при наведении на него мышкой с синего на красный!</p>`

ВАЖНО! Код должен быть записан в одну строчку

Разберем код.

1. `style="color:blue"` определяется стиль текста в данном абзаце
2. `onmouseover=` событие которое может произойти с этим абзацем, в кавычках надо указать что при этом делать
3. `this.style.color='red'` изменить стиль абзаца: цвет текста на красный:
 - слово `this` используется для доступа к элементу (объекту), вызвавшему событие (к данному абзацу),
 - через точку указывается его свойство `style`, дающее доступ к стилям,
 - ещё через точку указывается **конкретное свойство**, значение которого мы хотим изменить (`color` - цвет текста)
 - затем идет знак присвоить `=`,
 - и затем значение свойства "цвет текста" - красный (`'red'`).

Рассмотрим еще один пример. Изменение цвета фона текста:

`<p style="background-color:blue" onmouseover="this.style.backgroundColor='red' " onmouseout="this.style.backgroundColor='blue'">Цвет фона текста меняется на красный при наведении мышкой на него!</p>`

Разберем код.

1. `style="background-color:blue"` определяется стиль текста в данном абзаце (цвет фона)
2. `onmouseover=` (навести мышь) и `onmouseout=` (увести мышь) события которые могут произойти с этим абзацем, в кавычках надо указать что при этом делать
3. `this.style.backgroundColor='red'` изменить стиль абзаца: цвет фона на красный.

Задание №4

- 1) Измените созданный ранее файл следующим образом.
- 2) Создайте три абзаца
- 3) Напишите скрипт, изменяющий цвет текста первого абзаца при наведении мышкой
- 4) Напишите скрипт, изменяющий цвет фона текста второго абзаца при наведении мышкой
- 5) Напишите скрипт, изменяющий цвет фона текста третьего абзаца только при наведении мышкой

Метод *alert*

Метод `window.alert` отображает диалоговое окно, в которое помещается сообщение для пользователя. Этот метод часто используется при обработке полей формы. Если в какое-либо из полей формы введено неверное значение, то пользователю посылается сообщение, которое изображено на рис. 1.

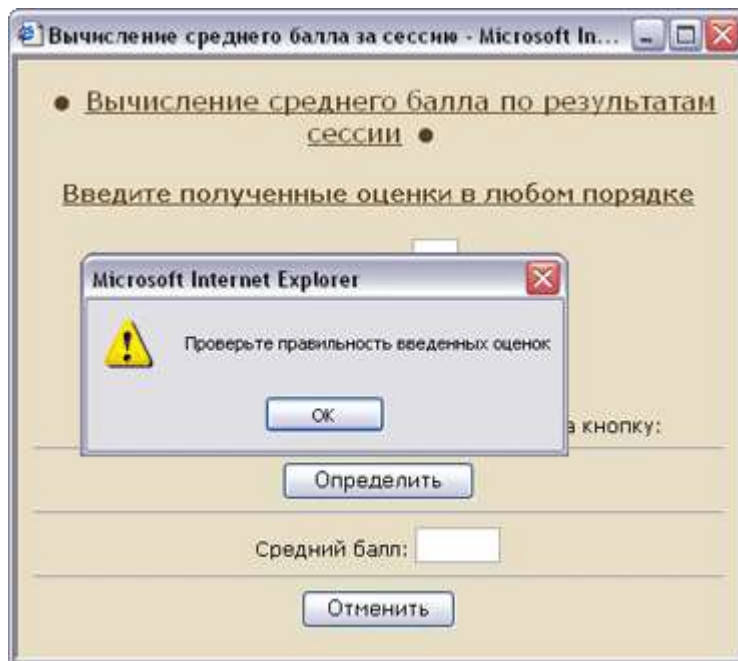


Рис. 1. Метод alert при анализе введенных данных

Документ, HTML- код которого представлен в листинге 1, содержит сценарий, при работе которого проверяется правильность введенных данных.

Листинг 1. Метод alert для проверки введенных данных

```
<HTML>
<HEAD>
<TITLE>Вычисление среднего балла за сессию</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!-- //
function msgot()
{ window.status = "неверно введены данные";
  alert ('Проверьте правильность введенных оценок')
  window.status = ""}
function st (obj )
{ var a = Number(obj.num1.value);
  var b = Number(obj.num2.value);
  var c = Number(obj.num3.value);
  var d = Number(obj.num4.value);
  if ((a < 2 ) || ( a > 5) || (b < 2 ) || ( b > 5)||
      (c < 2 ) || ( c > 5) || (d < 2 ) || ( d > 5))
    {msgot()}
  else
    obj.ball.value=(a+b+c+d)/4
}
//-->
</SCRIPT>
</HEAD>
<BODY>
<h4>Вычисление среднего балла по результатам сессии</h4>
<h4>Введите полученные оценки в любом порядке</h4>
<FORM name="form1">
Оценка 1: <INPUT type="text" size=1 name="num1"><BR>
```

```

Оценка 2: <INPUT type="text" size=1 name="num2"><BR>
Оценка 3: <INPUT type="text" size=1 name="num3"><BR>
Оценка 4: <INPUT type="text" size=1 name="num4"><BR>
Для определения среднего балла нажмите на кнопку:<HR>
< INPUT type="button" value=Определить onClick="st(form1)"><HR>
Средний балл: <INPUT type="text" size=5 name="ball"><HR>
<INPUT type="reset" value=Отменить>
< /FORM></BODY></HTML>

```

Задание №5

- 1) Отладьте страницу вычисления среднего балла.
- 2) Измените скрипт таким образом, чтобы при среднем балле менее 3 выдавалось сообщение: «Стыдно, вы двоечник!»

Метод *confirm*

Метод *confirm* отображает диалоговое окно подтверждения выполнения операции. Оно содержит две кнопки **ОК** и **Cancel (Отмена)**, позволяющие выбрать один из вариантов. Если пользователь щелкнул по кнопке **ОК**, то возвращается значение *true*, после щелчка по кнопке **Отмена** возвращается значение *false*. В функции *quest()* листинга 2 анализируется возвращаемое методом *confirm* значение и выполняются соответствующие действия.

Листинг 2. Метод *confirm* объекта *window*

```

<HTML>
< HEAD>
< TITLE>Отгадка задуманного числа</TITLE>
< SCRIPT LANGUAGE="JavaScript">
<!-- //
function test (obj)
{ var i=Number(obj.num1.value)
  var j=Number(obj.num2.value)
  var k
  var s=" "
  var p
  while ( i < j )
  { if ((i+j)%2== 0)
    k=(i+j)/2
    else
    k=(i+j-1)/2
    s="Задуманное число меньше или равно "+k+"?"
    p=confirm(s)
    if (p)
      j=k
    else
      i=k+1
  }
  obj.res.value=i
}
function cont (obj)
{ if (obj.res.value != obj.num.value )
  alert ("Кто-то обманывает")
}

```

```

else
    alert ("Все хорошо!")
}
//-->
</SCRIPT>
</HEAD>
<BODY>
<h4>Задумайте число в интервале</h4>
<FORM name="form1">
<input type="text" name="num1" size=4> и
<input type="text" name="num2" size=4><br>
Для проверки введите задуманное число
<input type="password" name="num" size=4><br>
<HR>
Ответьте на вопросы после нажатия кнопки <b>Отгадай</b><br>
<input type="button" value="Отгадай" onclick="test(form1)"><br>
Задуманное число: <input type="text" name="res" size=10><br>
<input type="button" value="Проверка" onclick="cont(form1)">
<input type="reset" value="Очистить">
</FORM></BODY></HTML>

```

На рис. 2 документ, HTML-код которого хранится в листинге 2.

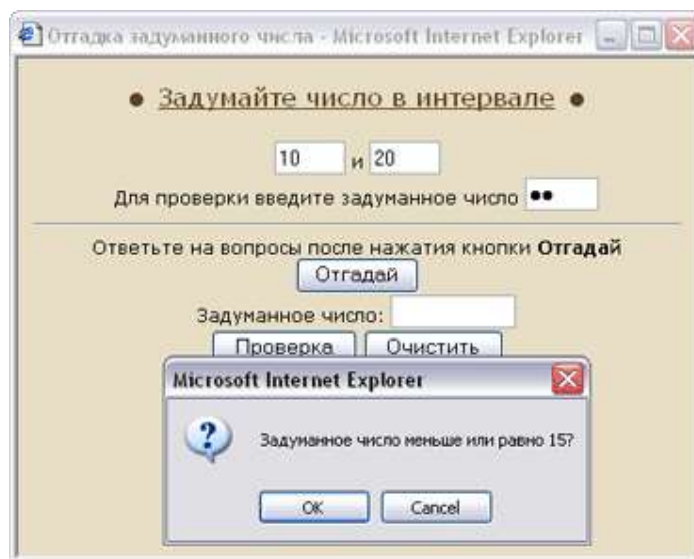


Рис. 2. Диалог с пользователем с помощью метода confirm.

Задание №6

- 1) Отладьте страницу из листинга 2

Метод *prompt*

Метод `prompt` используется для вывода диалогового окна запроса данных. При щелчке по кнопке **ОК** введенные пользователем в текстовое поле данные отображаются в документе. Если выбрано значение `Cancel`, то возвращается значение `Null`. Метод `prompt` имеет второй параметр, с помощью которого задается значение по умолчанию.

Листинг 3. Диалоговое окно запроса данных

```
<HTML>
< HEAD>
< TITLE>Метод prompt объекта window</TITLE>
< SCRIPT>
function reg (obj)
{ var s= window.prompt ("Введите Ваше имя", "")
  obj.regname.value = s
}
< /SCRIPT>
< /HEAD>
< BODY>
< form name="form1">
< input type=text name=regname>
< input type="button" value="Регистрация" onClick= "reg(form1)">
< /form></BODY></HTML>
```

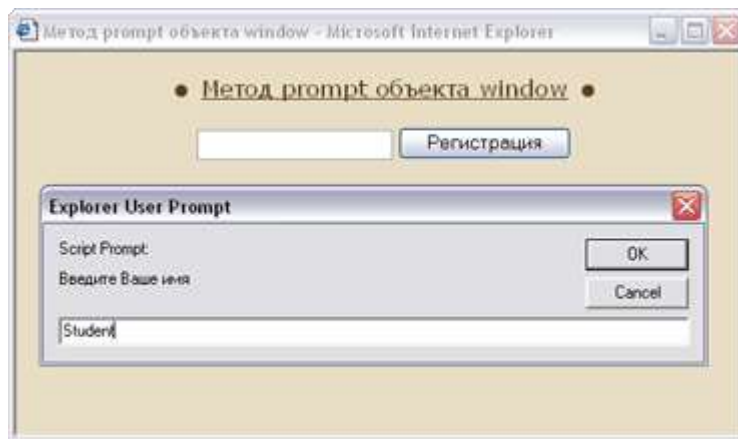


Рис. 3. Диалоговое окно запроса данных

Задание №7

- 1) Отладьте страницу из листинга 3

Задание для самостоятельной работы

Создайте интерактивную страницу с формой и диалогом, производящую расчёты для определения стоимости или физического объёма ваших товаров(услуг) согласно таблице вариантов и подключите её по ссылкам к созданным ранее страницам.

Таблица вариантов

№ варианта	Задание для расчёта
1	Объём бетона для заливки фундаментной плиты
2	Объём бетона для заливки ленточного фундамента
3	Количество кирпичей для кладки глухой стены заданной толщины
4	Объём и стоимость топлива для заданного пробега автомобиля с заданным удельным расходом и стоимостью
5	Количество мешков штукатурной смеси для оштукатуривания заданной площади
6	Сечение провода для передачи электроэнергии для заданного тока и плотности тока в проводнике
7	Урожай зерновых с поля заданных размеров и урожайности
8	Количество фундаментных блоков для устройства ленточного фундамента
9	Расстояние прямой видимости для антенны заданной высоты по глади моря
10	Объём видеопамати для хранения изображения заданного размера в рх и глубины цвета
11	Время передачи сообщения заданного объёма в байтах по каналу с заданной пропускной способностью
12	Время прохождения сигнала до космического объекта, находящегося от передатчика на расстоянии в астрономических единицах
13	Дина транспортной ленты для транспортёра, находящегося на заданном расстоянии и высоте от объекта
14	Время полёта космического корабля в годах, месяцах и днях до объекта на расстоянии в световых годах со скоростью в км/с
15	Величина налогового платежа НДФЛ для годового дохода по новой ставке 13% и 15% для доходов выше 5 000 000 руб.

16	Переплата по кредиту заданной величины и процентной ставки для заданного срока
17	Доход по депозиту заданного объема, ставки, срока с учётом ежемесячной капитализации процентов
18	Количество краски в банках заданного объема и расхода для окраски заданной площади
19	Количество рулонов обоев заданной длины для оклейки комнаты заданных размеров и высоты оклейки
20	Длина кабеля тёплого пола для обогрева комнаты заданных размеров при укладке змейкой с заданным шагом
21	Количество листов бумаги А4 для распечатки файла с заданным числом символов шрифтом Times New Romans 14 полуторным интервалом и полями по умолчанию

Контрольные вопросы

1. Где могут быть размещены выражения *JavaScript* ?
2. Перечислите основные узлы дерева HTML документа.
3. Как применять программный интерфейс HTML DOM?
4. Как изменить свойство узлов?
5. Как применять регулярные выражения в *JavaScript* ?
6. В чем сущность объектной модели браузера?
7. Для чего применяется библиотека *jQuery* ?
8. Что такое DOM (Document Object Model)?
9. Что такое куки?
10. Как получить доступ к информации о текущей сессии?
11. Как записать информацию в куки?
12. Что такое Ajax – приложение?