

Язык программирования JavaScript



Оглавление

Понятие **JavaScript**

Окружение: **DOM, BOM, JS**

Объектная модель документа (**DOM**)

Объектная модель браузера (**BOM**)

Вставка скрипта

Типы данных

Целый и вещественный типы данных

Строковый и булевой типы данных

Переменные

Операторы **JavaScript**

Оператор арифметические операторы

Операторы сравнения и логические операторы

Специальные операторы

Оператор условия **if...else**

Операторы цикла

Поиск по **id**

Встроенные объекты **JavaScript**

Объект **Math**

Объект **Date**

Объект **Number**

Функции

Пример

Перечень вопросов по изученному материалу

Варианты заданий по **JavaScript**



Понятие JavaScript

Язык программирования JavaScript разработан фирмой Netscape для создания интерактивных HTML-документов.

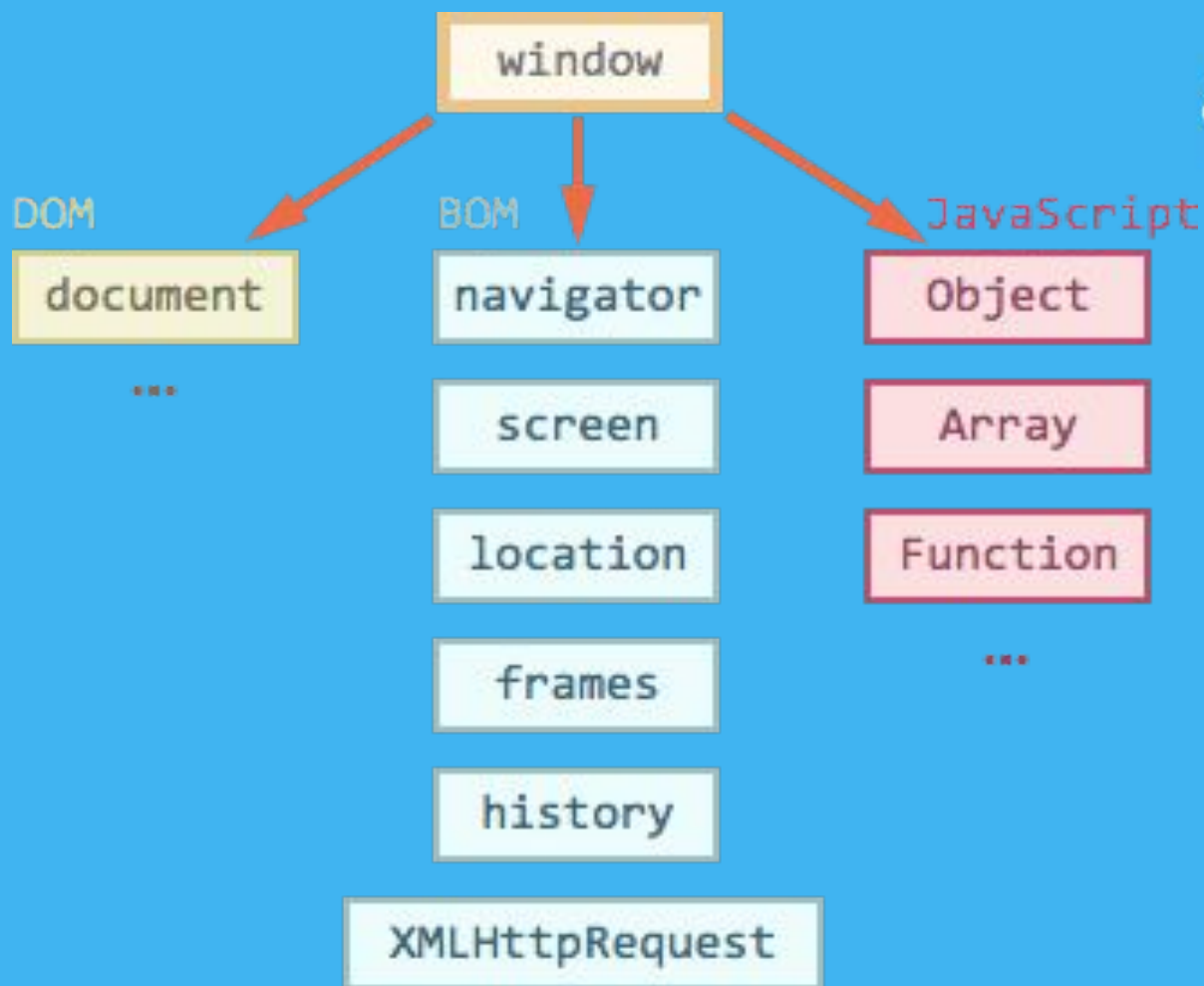
Основные области использования языка JavaScript при создании интерактивных HTML-страниц:

- Динамическое создание документа с помощью сценария;
- Оперативная проверка достоверности заполняемых пользователем полей форм HTML до передачи их на сервер;
- Создание динамических HTML-страниц совместно с каскадными таблицами стилей и объектной моделью документа;
- Взаимодействие с пользователем при решении “локальных” задач, решаемых приложением JavaScript, встроенном в HTML-страницу.



Окружение: DOM, BOM и JS

На рисунке ниже схематически отображена структура, которая получается если посмотреть на совокупность браузерных объектов с «высоты птичьего полёта».



Объектная модель документа(DOM)

Глобальный объект **document** даёт возможность взаимодействовать с содержимым страницы.

Пример использования:

```
document.body.style.background = 'red';
```

```
alert( 'Элемент BODY стал красным, а сейчас  
обратно вернётся');
```

```
document.body.style.background = '';
```



Оглавление

Объектная модель браузера (BOM)

BOM — это объекты для работы с чем угодно, кроме документа.

Например:

- Объект **navigator** содержит общую информацию о браузере и операционной системе.
- Объект **location** содержит информацию о текущем URL страницы и позволяет перенаправить посетителя на новый URL.
- Функции **alert/confirm/prompt** — тоже входят в BOM.



Оглавление

Вставка скрипта

JavaScript в HTML-файл можно подключить

2 способами:

1. В коде HTML-страницы

```
<SCRIPT Language="Javascript">
```

...

```
</SCRIPT>
```

2. Внешний файл.

```
<SCRIPT language=javascript src="имя  
файла"> </SCRIPT>
```



Типы данных

JavaScript использует переменные для хранения данных определенного типа. Реализация JavaScript является примером языка свободного использования типов. **В JavaScript не обязательно задавать тип переменной.** Ее тип зависит от типа хранимых в ней данных, причем при изменении типа данных меняется и тип переменной.

JavaScript поддерживает четыре простых типа данных:

- *Целый*
- *Вещественный*
- *Строковый*
- *Булевый, или логический*

Для присваивания переменным значений основных типов применяются литералы – буквенные значения данных соответствующих типов.



Целый

Целые литералы являются последовательностью цифр и представляют обычные целые числа со знаком или без знака:

- **123 // целое положительное число**
- **-123 // целое отрицательное число**
- **+123 // целое положительное число**

Вещественный

Для задания вещественных литералов используется синтаксис чисел с десятичной точкой, отделяющей дробную часть числа от целой, или запись вещественных чисел в научной нотации с указанием после символа “e” или “E” порядка числа. Пример правильных вещественных чисел:

1.25 0.125 e 01 12.5 E -1 0.0125 E +2



Строковый

Строковый литерал – последовательность алфавитно-цифровых символов, заключенная в одинарные (‘) или двойные кавычки (“), например: “Ира”, ‘ИРА’. При задании строковых переменных нельзя смешивать одинарные и двойные кавычки. Двойные кавычки – это один самостоятельный символ, а не последовательность двух символов одинарных кавычек. Если в строке нужно использовать символ кавычек, то строковый литерал необходимо заключать в кавычки противоположного вида.

Булевой

Булевые литералы имеют два значения: **true** и **false**, и используются для обработки ситуаций да/нет в операторах сравнения



Переменные

Каждая переменная имеет имя, которое должно начинаться с буквы латинского алфавита, либо символа подчеркивания “_”, за которым следует любая комбинация алфавитно-цифровых символов или символов подчеркивания.

Язык JavaScript чувствителен к регистру. Это означает, что строчные и прописные буквы алфавита считаются разными символами.

Определить переменную можно двумя способами:

- Оператором **var**
- Оператором присваивания (**=**)

Оператор **var** используется как для задания, так и для инициализации переменной и имеет синтаксис:

var имя_переменной [= начальное_значение];

Необязательный оператор присваивания задает данные, которые содержит переменная. Их тип определяет и тип переменной. Если при определении переменной ей не присвоено никакого значения, то ее тип не определен.



Операторы JavaScript

Арифметические операторы

| Оператор | Название |
|-----------|--|
| + | <i>Сложение</i> |
| - | <i>Вычитание</i> |
| * | <i>Умножение</i> |
| / | <i>Деление</i> |
| % | <i>Остаток от деления целых чисел</i> |
| ++ | <i>Увеличение значения переменной на единицу</i> |
| -- | <i>Уменьшение значения переменной на единицу</i> |



Операторы сравнения

| Оператор | Название |
|--------------|------------------|
| == | Равно |
| != | Не равно |
| >= | Больше или равно |
| <= | Меньше или равно |
| > | Строго больше |
| < | Строго меньше |

Логические операторы

| Оператор | Синтаксис | Описание |
|-------------------------------------|--------------------------------|---|
| && (логическое И) | <i>выраж1</i> && <i>выраж2</i> | Возвращает <i>выраж1</i> , если оно преобразуется или равно <code>false</code> , иначе <i>выраж2</i> |
| (логическое ИЛИ) | <i>выраж1</i> <i>выраж2</i> | Возвращает <i>выраж1</i> , если оно преобразуется или равно <code>true</code> , иначе <i>выраж2</i> |
| ! (логическое НЕ) | ! <i>выраж</i> | Если <i>выраж</i> равно <code>true</code> , возвращает <code>false</code> ; если <i>выраж</i> равно <code>false</code> , возвращает <code>true</code> . |



Оглавление

Специальные операторы

| | |
|---------------|--|
| ?: | <i>Позволяет Вам выполнить простое "if...then...else"</i> |
| delete | <i>Позволяет Вам удалять свойство объекта или элемент с определенным индексом в массиве.</i> |
| new | <i>Позволяет Вам создавать образец определяемого пользователем типа объекта или одного из встроенных типов объектов.</i> |
| this | <i>Ключевое слово, которое Вы можете использовать, чтобы обратиться к текущему объекту.</i> |
| void | <i>Оператор void определяет выражение, которое будет оценено без возврата значения.</i> |
| typeof | <i>Возвращает строку, указывающую тип неоцененного операнда.</i> |



Оператор условия **if...else**

Оператор условия **if** применяется, если необходимо вычислить некоторый блок операторов в зависимости от истинности заданного условия, и имеет следующий синтаксис:

```
if (условие) {  
  операторы1  
}  
[else {  
  операторы2  
}]
```

Первая группа операторов операторы1 выполняется при условии истинности выражения условие. Необязательный блок **else** задает группу операторов операторы2, которая будет выполнена в случае ложности условия, заданного в блоке **if**.

Внутри группы выполняемых операторов могут использоваться любые операторы **JavaScript**, в том числе и операторы условия.



Операторы цикла

Оператор цикла повторно выполняет последовательность операторов JavaScript, определенных в его теле, пока не выполнится некоторое заданное условие.

Оператор цикла **for** позволяет организовать выполнение блока операторов заданное число раз. Он определяет переменную, которая изменяет свое значение во время выполнения цикла. Условие завершения цикла зависит от значения этой переменной. Оператор имеет следующий синтаксис:

```
For([инициал_выражение];[условие];  
[изменяющее_выражение]) {  
[операторы]  
}
```

Цикл **while** выполняется пока истинно выражение, задающее условие выполнения цикла. Его синтаксис следующий:

```
while (условие) {  
[операторы]  
}
```



document.getElementById или просто id

Если элементу назначен специальный атрибут id, то можно получить его прямо по переменной с именем из значения id.

Правильной и общепринятой практикой является доступ к элементу вызовом

document.getElementById("идентификатор").

По стандарту значение id должно быть уникально, то есть в документе может быть только один элемент с данным id. И именно он будет возвращён.

Если в документе есть несколько элементов с уникальным id, то поведение неопределено. То есть, нет гарантии, что браузер вернёт именно первый или последний — вернёт случайным образом.



Встроенные объекты JavaScript

Встроенные объекты языка доступны в любом месте скрипта и в любой ситуации. В литературе эти объекты так же называют классами.

Рассмотрим некоторые встроенные объекты
JavaScript:



Оглавление

Объект *Math*

Предоставляет набор свойств и методов для работы с математическими константами и функциями. Для их использования нужна сначала ссылка на объект *Math*, затем — на требуемый метод:

```
var randomNumber = Math.random(),
```

где *randomNumber* – случайное число.

| | |
|-----------------|--|
| Abs (a) | Возвращает модуль числа <i>число</i> |
| acos (a) | Возвращает арккосинус числа <i>число</i> |
| asin (a) | Возвращает арксинус числа <i>число</i> |
| atan (a) | Возвращает арктангенс числа <i>число</i> |
| exp (a) | Возвращает экспоненту в степени число |



| | |
|--------------------|--|
| ceil (a) | Округляет число вверх до ближайшего целого |
| cos (a) | Возвращает косинус числа <i>число</i> |
| floor (a) | Округляет <i>число</i> вниз до ближайшего целого |
| log (a) | Возвращает натуральный логарифм числа <i>число</i> |
| max (a,a) | Возвращает большее из чисел <i>число1, число2</i> |
| min(a,b) | Возвращает меньшее из чисел <i>число1, число2</i> |
| pow (a,b) | Возвращает <i>число1</i> в степени <i>число2</i> |
| random () | Возвращает случайное число в диапазоне 0 и 1 |
| round (a) | Округляет <i>число</i> до ближайшего целого |
| sin (a) | Возвращает синус числа <i>число</i> |
| sqrt (a) | Возвращает квадратный корень из числа <i>число</i> |
| tan (a) | Возвращает тангенс числа <i>число</i> |



Объект Date

Объект Date поддерживает метод для работы с датами и временем в JavaScript. Новые экземпляры объектов Date создаются так:

`newDateObject = new Date([значение])`

Значение — необязательные установки даты и времени для нового объекта. Если они не заданы, используются текущие дата и время. Значение может иметь такой вид:

- миллисекунды — начиная с полуночи по Гринвичу 1 января 1970 года;
- год, месяц, день (например, 1997,0,27 — 27 января 1997);
- год, месяц, день, часы, минуты, секунды;
- месяц день, год часы: минуты: секунды (например, September 23,1997 08:25:30).

| | |
|-------------------------------|--|
| <code>getDate</code> | Возвращает число месяца как целое число от 1 до 31 |
| <code>getDay ()</code> | Возвращает день недели как целое число от 0 (воскресенье) до 6 (суббота) |



| | |
|----------------------------------|--|
| getHours () | Возвращает часы как целое от 0 до 23 |
| getMinutes () | Возвращает минуты как целое от 0 до 59 |
| getMonthO | Возвращает месяц как целое от 0 (январь) до 11 (декабрь) |
| getSeconds () | Возвращает секунды как целое от 0 до 59 |
| getFullYear () | Возвращает год без первых двух разрядов как целое число |
| setDate (значение) | Устанавливает день месяца с помощью целого числа значение от 1 до 31 |
| setHours (значение) | Устанавливает часы с помощью целого числа значение от 0 до 23 |
| setMinutes (значение) | Устанавливает минуты с помощью целого числа значение от 0 до 59 |
| setMonth (значение) | Устанавливает месяц с помощью целого числа значение от 0 до 11 |
| setSeconds (значение) | Устанавливает секунды с помощью целого числа значение от 0 до 59 |



Объект *Number*

Объект [Number](#) является объектом-обёрткой, позволяющей вам работать с числовыми значениями. Объект Number создаётся через конструктор `Number()`.

| Свойство | Описание |
|--------------------------------------|---|
| <code>MAX_VALUE</code> | Максимальное численное значение в JavaScript (~1,79E+308) |
| <code>MIN_VALUE</code> | Минимальное численное значение в JavaScript (~2,22E-308) |
| <code>MIN_VALUE</code> | Значение, не являющееся числом |
| <code>NEGATIVE_INFINITY</code> | Минус бесконечность |
| <code>POSITIVE_INFINITY</code> | Плюс бесконечность |
| Метод | Описание |
| <code>toString([основание])</code> | Преобразует число в десятичную систему счисления (или в число в системе по основанию основание,) и возвращает в виде строки |
| <code>valueOf ()</code> | Возвращает двоичное численное значение числа |



Функции

Функция – это именованная последовательность операторов, которая инициализируется и выполняется простой ссылкой на имя функции.

Процедура задается оператором **function**, имеющим следующий синтаксис:

```
function имя_функции ([параметры] {  
  [операторы]  
}
```

где имя функции – любое правильное имя языка JavaScript, **параметры** – список передаваемых в процедуру параметров, элементы которого отделяются запятыми.

Оператор function только определяет процедуру, но не выполняет ее. Для вызова процедуры достаточно указать ее имя с заданными в скобках параметрами.

Процедура может возвращать некоторое вычисляемое в ней значение. В этом случае в операторах, определяющих последовательность выполняемых ею действий, обязательно должен присутствовать оператор **return**, задающий возвращаемое функцией значение.



Пример

Напишите программу на языке JavaScript, которая будет рассчитывать работу силы по формуле: $A = F \cdot S \cdot \cos(\alpha)$. Данные должны вводиться заранее с клавиатуры. В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести белым цветом, на зеленом фоне с выравниванием по центру и размером шрифта – 25px. Добавить кнопку, которая будет перезапускать программу и ссылку, которая будет подставлять случайные значения в формулу и выводить на экран.

Работа с JavaScript не обходится без таких языков программирования как HTML, который классифицирует части документа и обеспечивает его правильное отображение в браузере, и CSS, который задает стили документа.

Для лучшего понимания, мы добавили к объектам программы комментарии, которые объясняют работу того или иного участка программы.



```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>Пример</title>
```

```
  <!-- Добавляем стили -->
```

```
  <style>
```

```
    /* стили для фона */
```

```
    body {
```

```
      width: 100%; /* устанавливаем ширину для тега body 100%*/
```

```
      height: 100%; /* устанавливаем высоту для тега body 100%*/
```

```
      text-align: center; /* выравниваем по центру все что в теге body */
```

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
<!--
```

Добавляем <div></div> элемент с атрибутом id="rabota" куда будем выводить полученный результат.

Атрибут id должен быть уникальным и мы будем использовать значение id для поиска этого элемента.

```
-->
```

```
<div id="rabota"></div>
```



<!--

Добавляем элемент кнопки, прописывая тег `<button></button>` с атрибутами и текстом кнопки.

Атрибуты:

`type`: указывает тип кнопки

`onclick`: обработчик события, которое возникает при щелчке по кнопке.

В данном примере при щелчке по кнопке срабатывает событие `onclick` и идет вызов функции с именем `start()`, которая запускает программу.

-->

```
<button type="button" onclick="start()">Перезапустить</button>
```

<!--

Добавляем ссылку, которая описывается тегом `<a>`

Атрибуты:

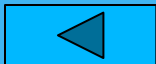
`href`: это обязательный атрибут, где указывается ссылка на другую страницу, в нашем примере при нажатии на ссылку, мы не будем переходить на другую страницу, поэтому поставим символ `#`.

`onclick`: обработчик события, которое возникает при щелчке по кнопке.

В данном примере при щелчке по кнопке срабатывает событие `onclick` и идет вызов функции с именем `randomResult()`, которая вычисляет значение по формуле со случайными данными.

-->

```
<a href="#" onclick="randomResult()">Добавить результат со  
случайными данными</a>
```





```
<!-- Добавляем Javascript -->
<script type="text/javascript">
// начало
start();
// основная функция программы, мы назвали ее start()
function start() {
// ввод F, вызывая стандартную функцию prompt
var f = window.prompt("Введите F:");
// проверка на валидность введенных данных, вызывая нашу функцию validation
if (validation(f)) {
// ввод S
var s = window.prompt("Введите S:");
if (validation(s)) {
// ввод a
var a = window.prompt("Введите a:");
if (validation(a)) {
// находим элемент по атрибуту id, в нашем примере он равен rabota
var infoBlock = document.getElementById("rabota");
//вывод на экран
var result = A(f, s, a);
infoBlock.innerHTML = "<div> A=" + result + "</div>"; // выводим результат
```

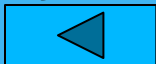


```
// применение стилей для фона
document.body.style.background = "green";
// применение стилей для блока с информацией
infoBlock.style.textAlign = "center"; // выравниваем текст
infoBlock.style.color = "white"; // изменяем цвет текста
infoBlock.style.fontSize = 25+ "px"; // изменяем шрифт текста
```

```
    }
  }
}
// функция для вычисления работы
function A(F, S, a) {
  return F * S * Math.cos(a); // высчитываем значение по формуле
}
```

```
// функция для вывода результата со случайными значениями
```

```
function randomResult() {
// находим элемент по атрибуту id, в нашем примере он равен rabota
var infoBlock = document.getElementById("rabota");
var f = Math.random()*100; // случайное значение для F от 0 до 100
var s = Math.random()*100;
var a = Math.random()*100;
var newResult = A(f, s, a); // высчитываем новый результат по формуле со случайными
значениями вызывая функцию
infoBlock.innerHTML = infoBlock.innerHTML + "<div> A=" + newResult + "</div>"; // к
существующему результату добавляем новый
}
```



// функция для проверки значения на валидность

```
function validation(value) {
```

```
// проверяем, значение переменной value на пустоту или  
неопределенность
```

```
if (!value) {
```

```
// если значение пустое или неопределено,
```

```
// показываем сообщение об ошибке вызывая стандартную функцию alert
```

```
window.alert("Значение не может быть пустым!");
```

```
// выходим из функции возвращая значение false (типа boolean)
```

```
return false;
```

```
}
```

```
// проверяем, является ли значение переменной value числом
```

```
if (isNaN(value)) {
```

```
  window.alert("Вы ввели не число!");
```

```
return false;
```

```
}
```

```
return true;
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

[Запустить пример задачи в браузере](#)



Оглавление

Перечень вопросов по изученному материалу

1. Назовите основные области использования языка JavaScript?
2. Что такое объектная модель документа(DOM)?
3. Что такое объектная модель браузера(BOM)?
4. Что содержит объект Navigation?
5. Что содержит объект location?
6. Назовите 2 способа подключения JavaScript в HTML-файл.
7. Какие типы данных поддерживает JavaScript?
8. От чего зависит тип переменной в JavaScript?
9. Какими способами можно определить переменную в JavaScript?



10. Приведите примеры операторов равенства JavaScript.
11. Назовите логические операторы JavaScript.
12. Назовите специальные операторы JavaScript.
13. В каком случае применяется оператор условия `if...else`?
14. Какие существуют операторы цикла и в чем их отличие?
15. С какой целью присваивается атрибут `id` в JavaScript?
16. Какие вы знаете встроенные объекты JavaScript?
17. Для чего используется Объект *Math*?
18. Для чего используется Объект *Date*?
19. Для чего используется Объект *Number*?
20. Что такое функция в JavaScript?
21. Какой синтаксис имеет оператор `function`?



Варианты заданий по JavaScript

Вариант 1

Напишите программу на языке JavaScript, которая будет рассчитывать потенциальную энергию по формуле: $E_p = mgh$. Данные должны вводиться заранее с клавиатуры. Необходимо учитывать, что g – ускорение свободного падения (постоянная величина). В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести на зеленом фоне с выравниванием по левому краю. Добавить кнопку, которая будет перезапускать программу и ссылку, которая будет изменять цвет фона на красный.

Вариант 2

Напишите программу на языке JavaScript, которая будет рассчитывать мощность по формуле: $N = A/t$. Данные должны вводиться заранее с клавиатуры. В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести на красном фоне с выравниванием по центру. Добавить кнопку, которая будет позволять изменить параметр t , и 2 ссылки, при нажатии на которые, будет изменяться цвет фона на зеленый и фиолетовый.



Вариант 3

Напишите программу на языке JavaScript, которая будет рассчитывать кинетическую энергию по формуле: $E = (m \cdot v^2) / 2$. Данные должны вводиться заранее с клавиатуры. В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести на желтом фоне с выравниванием по центру и курсивом. Добавить кнопку, которая будет перезапускать программу и ссылку, которая будет увеличивать шрифт на 5px (начальный размер шрифта взять произвольно).

Вариант 4

Напишите программу на языке JavaScript, которая будет рассчитывать количество теплоты по формуле: $Q = c \cdot m \cdot (t_2 - t_1)$. Данные должны вводиться заранее с клавиатуры. В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести на голубом фоне с подчеркиванием. Добавить 2 кнопки, которые будут позволять изменить переменные c и m , и ссылку, которая будет изменять цвет фона на зеленый.



Вариант 5

Напишите программу на языке JavaScript, которая будет рассчитывать электрическое сопротивление по формуле: $R=U/I$. Данные должны вводиться заранее с клавиатуры. В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести на красном фоне с выравниванием по центру. Добавить 2 кнопки, одна из которых будет уменьшать значение I в 2 раза, а вторая – увеличивать I в 2 раза.

Вариант 6

Напишите программу на языке JavaScript, которая будет рассчитывать удаленное электрического сопротивление по формуле: $=(R*S)/I$. Данные должны вводиться заранее с клавиатуры. В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести курсивом, с подчеркиванием. Добавить 3 кнопки, одна из которых будет придавать фону коричневый цвет, вторая – голубой, третья – зеленый цвет.



Вариант 7

Напишите программу на языке JavaScript, которая будет рассчитывать силу тока по формуле: $I=q/t$. Данные должны вводиться заранее с клавиатуры. В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести на фиолетовом фоне, курсивом. Добавить 2 кнопки, одна из которых будет показывать текущую полную дату (дата, время и часовой пояс), а вторая – удалять выведенный результат даты с экрана.

Вариант 8

Напишите программу на языке JavaScript, которая будет рассчитывать электрическое напряжение по формуле: $U=A/q$. Данные должны вводиться заранее с клавиатуры. В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести на голубом фоне с выравниванием по центру, шрифтом 18px. Добавить 2 ссылки, одна из которых будет уменьшать шрифт на 2px, а вторая – увеличивать на 2px.



Вариант 9

Напишите программу на языке JavaScript, которая будет рассчитывать силу центростремительного ускорения по формуле: $E=(m*v*v)/2$. Данные должны вводиться заранее с клавиатуры. В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести на оранжевом фоне с выравниванием по центру и размером шрифта – 20px. Добавить кнопку, которая будет перезапускать программу и ссылку, которая будет показывать текущую дату и время.

Вариант 10

Напишите программу на языке JavaScript, которая будет рассчитывать индукцию магнитного поля по формуле: $B=F/(l*I)$. Данные должны вводиться заранее с клавиатуры. В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести на зеленом фоне с выравниванием по левому краю и размером шрифта – 36px. Добавить кнопку, которая будет перезапускать программу и ссылку, которая будет изменять цвет фона на фиолетовый.



Вариант 11

Напишите программу на языке JavaScript, которая будет рассчитывать поглощенную дозу излучения по формуле: $D=E/m$. Данные должны вводиться заранее с клавиатуры. В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести с выравниванием по центру, подчеркнутым и размером шрифта – 15px. Добавить кнопки, которые будут позволять изменить любую из переменных и ссылку, которая будет увеличивать выведенный результат на 5 px.

Вариант 12

Напишите программу на языке JavaScript, которая будет рассчитывать угловую скорость при равномерном движении по формуле: $w=2 \cdot \pi \cdot v$. Необходимо учитывать, что π – постоянная величина. Данные должны вводиться заранее с клавиатуры. В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести на красном фоне с подчеркиванием и размером шрифта – 20px. Добавить кнопку, которая будет показывать текущую дату и время и ссылку, которая будет перезапускать программу.



Вариант 13

Напишите программу на языке JavaScript, которая будет рассчитывать основное уравнение газа по формуле: $p=1/3^{**}n^{**}$. Данные должны вводиться заранее с клавиатуры. В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести с выравниванием по левому краю, на голубом фоне. Добавить 2 кнопки, одна из которых будет уменьшать полученное значение в 3 раза, а вторая – увеличивать результат в 3 раза.

Вариант 14

Напишите программу на языке JavaScript, которая будет рассчитывать основное уравнение газа по формуле: $p=F/S$. Данные должны вводиться заранее с клавиатуры. В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести на сером фоне, курсивом. Добавить 2 кнопки, при нажатии на первую цвет фона будет изменяться на красный, на вторую – цвет изменяется на голубой. Добавить ссылку, которая будет уменьшать шрифт на 2px.



Вариант 15

Напишите программу на языке JavaScript, которая будет рассчитывать количество вещества по формуле: $V=m/M$. Данные должны вводиться заранее с клавиатуры. В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести с выравниванием по центру, подчеркнутым и размером шрифта – 24px. Добавить кнопку, которая будет подставлять случайные значения в формулу и дополнительно выводить полученное значение и ссылку, которая будет перезапускать программу.

Вариант 16

Напишите программу на языке JavaScript, которая будет рассчитывать эквивалентную дозу по формуле: $H=D*K$. Данные должны вводиться заранее с клавиатуры. В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести с выравниванием по левому краю, подчеркнутым и размером шрифта – 28px. Добавить кнопки, которые будут позволять изменить переменные величины и кнопку перезапуска программы.



Вариант 17

Напишите программу на языке JavaScript, которая будет рассчитывать изменение энергии связи по формуле: $E = m \cdot c^2$. Данные должны вводиться заранее с клавиатуры. В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести на фиолетовом фоне с выравниванием по центру, курсивом и размером шрифта – 19px. Добавить 2 кнопки, одна из которых будет выводить текущую дату и время, а вторая – перезапускать программу.

Вариант 18

Напишите программу на языке JavaScript, которая будет рассчитывать скорость по формуле: $v = S/t$. Данные должны вводиться заранее с клавиатуры. В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести на желтом фоне с выравниванием по левому краю, подчеркнутым и размером шрифта – 30px. Добавить кнопку, которая будет менять цвет фона на красный. Добавить 2 ссылки, одна из которых будет уменьшать шрифт на 5px, а вторая – увеличивать на 5px.



Вариант 19

Напишите программу на языке JavaScript, которая будет рассчитывать вес тела по формуле: $P=g*m$. Данные должны вводиться заранее с клавиатуры. Необходимо учитывать, что g – ускорение свободного падения (постоянная величина). В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести с выравниванием по левому краю, курсивом и размером шрифта – 20px. Добавить кнопку, которая будет перезапускать программу, а также ссылку, которая будет выравнивать результат по центру.

Вариант 20

Напишите программу на языке JavaScript, которая будет рассчитывать работу по формуле: $A=F*S$. Данные должны вводиться заранее с клавиатуры. В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести на коричневом фоне с выравниванием по центру, подчеркнутым и размером шрифта – 16px. Добавить кнопку, которая будет копировать полученный результат на свободное место страницы и ссылку, которая будет выводить текущую дату и время.



Вариант 21

Напишите программу на языке **JavaScript**, которая будет рассчитывать давление по формуле: $P=F/S$. Данные должны вводиться заранее с клавиатуры. В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести на красном фоне, подчеркнутым и размером шрифта – 18px. Добавить 2 кнопки, одна из которых будет менять цвет фона на зелёный, а вторая – увеличивать шрифт на 5px. Добавить ссылку, которая будет перезапускать программу.

Вариант 22

Напишите программу на языке **JavaScript**, которая будет рассчитывать момент силы по формуле: $M=F*l$. Данные должны вводиться заранее с клавиатуры. В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести на розовом фоне с выравниванием по центру, подчеркнутым и размером шрифта – 15px. Добавить кнопку, которая будет подставлять случайные значения в формулу и дополнительно выводить полученное значение, а также ссылку, которая будет перезапускать программу.



Вариант 23

Напишите программу на языке **JavaScript**, которая будет рассчитывать потенциальную энергию по формуле: $E_p = (k^*)/2$. Данные должны вводиться заранее с клавиатуры. В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести на зеленом фоне с выравниванием по центру, курсивом, и размером шрифта – **24px**. Добавить кнопки, которые будут позволять изменить переменные данные, и кнопку, которая будет выравнивать текст по левому краю.

Вариант 24

Напишите программу на языке **JavaScript**, которая будет рассчитывать количество теплоты по формуле: $Q = L * m$. Данные должны вводиться заранее с клавиатуры. В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести с выравниванием по центру, курсивом и размером шрифта – **22px**. Добавить кнопку, которая будет выводить текущую дату и время, и ссылку, которая будет изменять цвет фона на фиолетовый.

Вариант 25

Напишите программу на языке **JavaScript**, которая будет рассчитывать количество теплоты по формуле: $Q = *R * t$. Данные должны вводиться заранее с клавиатуры. В случае не ввода данных или при вводе не числовых значений, должны выводиться сообщения об ошибке. Результат вывести на красном фоне с выравниванием по левому краю, и размером шрифта – **20px**. Добавить 2 кнопки, которые будут позволять изменить переменные **R** и **t**, а также ссылку, которая будет перезапускать программу.





СПАСИБО ЗА ВНИМАНИЕ



Оглавление