**Частное учреждение образования**

**«Гродненский колледж бизнеса и права»**

**Лабораторная работа № 17**

**по дисциплине**

**«Программные средства создания Internet - приложений»**

**Тема:** «Использование стандартных объектов JavaScript: Date, Array, Math»

для учащихся 4 курса специальности

2-40 01 01 «Программное обеспечение информационных технологий»

**Тема:** Использование стандартных объектов JavaScript: Date, Array, Math.

**Цель:** сформировать умения использовать стандартные объекты JavaScript

**Вид занятия:** Практическая работа.

**Тип занятия:** Формирование навыков.

**Время выполнения:** 2 часа

**Оборудование и методическое обеспечение:** IBM PC, приложение Блокнот, MS Internet Explorer карточки с заданием.

**Литература:**

1. Java Script.. Г. В. Гудман СПб.: Питер, 2004.

**План работы:**

1. Организационный этап занятия
2. Проверка домашнего задания
3. Информационно-подготовительный этап занятий
4. Исполнительский этап занятий
5. Постановка домашнего задания
6. Оценочно-рефлексивный этап занятия

**Ход занятия**

1. **Организационный этап занятия**

Проверка готовности учащихся к занятию, отметка отсутствующих, объявление темы и цели урока.

1. **Проверка домашнего задания**

Устный опрос учащихся по следующим вопросам:

1. Для чего предназначен объект Math?
2. Для чего предназначен объект Date?
3. Для чего предназначен объект Array?
4. **Информационно-подготовительный** **этап занятия**

Объект math работает с математическими функциями, а его методы можно использовать для вызова этих функций. Ниже представлены некоторые методы объекта math:

|  |  |
| --- | --- |
| **метод** | **описание** |
| abs | абсолютное значение |
| sin, cos, tan | тригонометрические функции |
| log | натуральный логарифм |
| exp | экспонента |
| pow | показательная функция |
| sqrt | квадратный корень |
| min | наименьшее значение |
| max | наибольшее значение |

Какие-то методы вы будете использовать часто, какие-то редко, а некоторые вам и вовсе не пригодятся. Рассмотрим пример использования объекта math.

Предположим мы хотим написать сценарий, который будет вычислять площадь треугольника по трем его сторонам. Для этого нам потребуется использовать формулу Герона:

формула ГеронаS – площадь, a, b, c - длины сторон треугольника

Итак, напишем в html-странице код формы:

<html>

<head>

<title>math javascript</title>

<link rel="stylesheet" type="text/css" href="style1.css">

<script type="text/javascript" src="script.js"></script>

</head>

<body>

<form name="forma3">

Сторона 1 <input type="text" size="8" maxlength="8" name="st1">

Сторона 2 <input type="text" size="8" maxlength="8" name="st2">

Сторона 3 <input type="text" size="8" maxlength="8" name="st3"><br><br>

<input type="button" value="Вычислить" onClick="areaOfTriangle(forma3);">

<input type="reset" value="Отменить">

Результат <input type="text" size="8" maxlength="8" name="res">

</form>

</body>

</html>

Теперь на странице script.js напишем код функции:

function areaOfTriangle(obj){

var a=1\*obj.st1.value;

var b=1\*obj.st2.value;

var c=1\*obj.st3.value;

var p=(a+b+c)/2;

var s=Math.sqrt(p\*(p-a)\*(p-b)\*(p-c));

obj.res.value=s;

}

Здесь мы использовали объект *Math* и его метод *sqrt* для извлечения квадратного корня. Выражение, из которого извлекается корень, должно быть взято в скобки.

Также обратите внимание на первые три строчки функции, они начинаются с *1\**, т.е. наши переменные a, b, c мы умножили на единицу. Зачем? Помните, в первом уроке мы говорили о том, что у каждой переменной есть тип, который определяется автоматически. Так как наши переменные приходят из текстового поля, то и тип они имеют *string*, т.е. они воспринимаются функцией не как числа, а как буквы.

Если бы мы перемножали эти переменные, то их тип автоматически переопределился бы к *number*, но мы их складываем, а знак операции *+* расценивается в данном случае, как конкатенация строк. Чтобы этого не происходило, мы и умножили наши переменные на единицу, преобразовав их таким образом к типу *number*. Ради эксперимента уберите умножение на единицу в этих трех строках и посмотрите, что "насчитает" сценарий. Затем верните правильный вариант и убедитесь, что все работает правильно.

Вроде все хорошо, но есть один нюанс: если извлекаемый корень является дробным числом, то дробная часть может быть бесконечно длинной. Для точных вычислений это необходимо, но в большинстве случаев достаточно двух знаков после запятой.

Чтобы округлить результат до N знаков после запятой можно воспользоваться методом *toFixed* объекта *Number*. Синтаксис записи следующий:

(x).toFixed(N)

где x - число, которое надо округлить, а N - число знаков после запятой.

Давайте исправим конец нашей функции и округлим результат до 2 знаков после запятой:

function areaOfTriangle(obj){

var a=1\*obj.st1.value;

var b=1\*obj.st2.value;

var c=1\*obj.st3.value;

var p=(a+b+c)/2;

var s=Math.sqrt(p\*(p-a)\*(p-b)\*(p-c));

s=s.toFixed(2);

obj.res.value=s;

}

для тренировки поэкспериментируйте с другими методами объекта math.

Объект ***Array***

***Пример*** Создание массивов.

<script> var a1=new Array();

a1[0]="А"; a1[1]="Б"; a1[2]="В"; a1[3]="Г";

a2=new Array("Пн", "Вт", "Ср", "Чт", "Пт","Сб", "Вс"); </script>

***Пример 2.***

<script> var a=new Array(10);

for (i=0; i<10; ++i) {

a[i]=prompt("Введите какое-нибудь значение");}; </script>

***ПРимер 3.***Заполниние массива десятью случайными числами из диапазона (0; 1) и вывод массива на экран:

[<script> a=new Array(10);](javascript:a=new%20Array(10);for%20(i=0;%20i%3c10;%20++i)%20%7ba%5bi%5d=Math.random();%7dvoid(0);)

[for (i=0; i<10; ++i) {](javascript:a=new%20Array(10);for%20(i=0;%20i%3c10;%20++i)%20%7ba%5bi%5d=Math.random();%7dvoid(0);)

[a[i]=Math.random();}](javascript:a=new%20Array(10);for%20(i=0;%20i%3c10;%20++i)%20%7ba%5bi%5d=Math.random();%7dvoid(0);)

[for (i=0; i<10; ++i) {](javascript:a=new%20Array(10);for%20(i=0;%20i%3c10;%20++i)%20%7ba%5bi%5d=Math.random();%7dvoid(0);)

[alert(a[i]); </script>](javascript:a=new%20Array(10);for%20(i=0;%20i%3c10;%20++i)%20%7ba%5bi%5d=Math.random();%7dvoid(0);)

1. **Исполнительский этап занятия**

*Самостоятельно*

* 1. Пользователь вводит номер текущего дня, а программа выдает ему словесное название дня. Например: 1 - "Понедельник"; 4 - "Четверг".
  2. Дано три массива размером 7 элементов: первый вводит пользователь; второй заполняется случайными числами от 1 до 10 000; третий является суммой двух предыдущих (первый элемент третьего массива равен сумме первого элемента первого массива и первого элемента второго массива)
  3. Заполнить массив из тридцати элементов случайными числами от 0 до 100. Попросить пользователя ввести любое число от 0 до 100 и сообщить имеется ли такое число в массиве.
  4. Math.random() дает псевдослучайное число в диапазоне от 0 до 1. Подумайте как составить генератор псевдослучайных чисел в диапазоне от 1 до 100.
  5. Пусть пользователь сначала скажет, сколько чисел он хочет ввести, а затем только введет их все. Например, если пользователь закажет 6 чисел, то среднее арифметическое надо будет вывести для 6 чисел
  6. Программа должна выводить факториал введенного пользователем числа. Вспомним, что факториал - это произведение последовательных натуральных чисел. Т.е. n!=1·2·3·...·n; 5!=1·2·3·4·5=120.
  7. Пользователь вводит номер квартиры, программа выводит на каком этаже и в каком подъезде находится эта квартира. Дополнительные условия: в доме 14 подъездов, 9 этажей, на каждом этаже - 4 квартиры.
  8. "Арифметическая прогрессия". Пользователь вводит первый член арифметической прогрессии, ее разность и количество элементов. Задача программы вывести такую прогрессию на экран.
  9. "Пифагоровы тройки". Пифагоровыми тройки называются, когда сумма квадратов двух натуральных чисел равна квадрату третьего натурального числа. Например, 3; 4; 5 (32 + 42 = 52). Задача программы вывести всевозможные такие комбинации в диапазоне от 1 до n, n вводит пользователь Программа, окажется немного сложнее, если также проверять можно ли составить треугольник из полученных чисел.
  10. "Счастливый" билетик. Пользователь вводит число, а программа проверяет "счастливое" это число или нет. Ответ будет положительным, если сумма цифр первой половины равна сумме цифр второй половины. Например, билет с серийным номером 263506 - "счастливый", так как 2 + 6 + 3 = 5 + 0 + 6 = 11. В том случае, если количество цифр нечетно, среднюю цифру учитывать не будем.
  11. Запросить количество чисел в 4 массивах, если длины массивов совпадают, то сложить эти массивы.

1. **Постановка домашнего задания**

Из литературы [2], с.39-42

1. **Оценочно-рефлексивный этап занятия**

Учащимся необходимо ответить на поставленные вопросы:

1. Что нового Вы узнали на занятии?
2. Что вызвало затруднение при выполнении задания?
3. Было ли Вам интересно на занятии?