

ПРАКТИЧЕСКАЯ РАБОТА № 4

ТЕМА: ОБЪЕКТ Math в JavaScript

Цель: Рассмотреть объект Math и его методы.

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 |

Пример. Предположим хотим написать сценарий, который будет вычислять площадь треугольника по трем его сторонам. Для этого нам потребуется использовать формулу Герона: S - площадь a, b, c - длины сторон треугольника. Итак, напомним 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 напомним 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? 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, как в примере .

| | | |
|-----------|----------------------|--|
| Сторона 1 | <input type="text"/> | <input type="button" value="Отменить"/> |
| Сторона 2 | <input type="text"/> | <input type="button" value="Вычислить"/> |
| Сторона 3 | <input type="text"/> | |
| Результат | <input type="text"/> | |

Замечание : 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.

Задания для самостоятельного выполнения

1. Исправьте конец функции и округлите результат до 2 знаков после запятой
2. Выведите случайное число между мин и максимальным числом.
3. Решите квадратное уравнение.