В данном уроке мы будем изучать стандартные функции JavaScript.

Изучите методы и функции по следующим ссылкам, постарайтесь запомнить и законспектировать полученную информацию: Math.abs, Math.round, Math.ceil, Math.floor, Math.min, Math.max, Math.sqrt,Math.pow, Math.random, isNaN, isFinite, parseInt, parseFloat, toFixed.

Обязательно изучите каждую статью до конца - примеры в конце статьи специально подобраны так, чтобы подготовить вас к решению зада

http://code.mu/books/javascript/base/rabota-s-matematicheskimi-funkciyami-v-javascript.html

***Функция Math.abs***

Функция Math.abs возвращает модуль числа, то есть из отрицательного числа делает положительное.

Синтаксис

Math.abs(число);

В данном примере будет выведен модуль числа -3:

document.write(Math.abs(-3));

Результат выполнения кода: 3

Пример

В данном примере будет выведен модуль числа 3:

document.write(Math.abs(3));

Результат выполнения кода: 3

***Функция Math.round***

Функция Math.round выполняет округление до ближайшего целого числа по правилам математического округления.

См. функции Math.ceil и Math.floor, которые также выполняют округление числа (ceil всегда в большую сторону, а floor - всегда в меньшую).

См. методы toFixed и toPrecision, которые также выполняют округление числа.

Синтаксис

Math.round(число);

Округлим до целых число 6.4:

document.write(Math.round(6.4));

Результат выполнения кода:

6

Округлим до целых число 6.6:

document.write(Math.round(6.6));

Результат выполнения кода: 7

Пример

Округлим до целых число 6.5:

document.write(Math.round(6.5));

Результат выполнения кода: 7

Пример

Округлим до целых число 6.49999:

document.write(Math.round(6.49999));

Результат выполнения кода: 6

***Функция Math.ceil***

Функция Math.ceil производит округление числа до целых всегда в большую сторону.

См. функции Math.round и Math.floor, которые также выполняют округление числа.

См. методы toFixed и toPrecision, которые также выполняют округление числа.

Синтаксис

Math.ceil(число);

Округлим число 3.00001 в большую сторону:

document.write(Math.ceil(3.00001));

Результат выполнения кода: 4

***Функция Math.floor***

Функция Math.floor производит округление числа до целых всегда в меньшую сторону.

См. функции Math.round и Math.ceil, которые также выполняют округление числа.

См. методы toFixed и toPrecision, которые также выполняют округление числа.

Синтаксис

Math.floor(число);

Округлим число 2.9999 в меньшую сторону:

document.write(Math.floor(2.9999));

Результат выполнения кода: 2

***Функция Math.min***

Функция Math.min возвращает минимальное число из группы чисел, переданных в функцию.

Если в функцию ничего не передано, то будет возращено Infinity.

По умолчанию функция не работает с массивами, однако с помощью хитрого приема можно найти минимальное значение массива: Math.min.apply(null, arr), где arr - произвольный массив.

См. также функцию Math.max, которая возвращает максимальное число из группы чисел.

Синтаксис

Math.min(число, число, число...);

В данном примере будет выведено минимальное число из группы чисел 40, 20, 42, 100, 67:

document.write(Math.min(40, 20, 42, 100, 67));

Результат выполнения кода: 20

В данном примере будет выведено минимальное число из группы чисел -1, -100, -30, -25, 40:

document.write(Math.min(-1, -100, -30, -25, 40));

Результат выполнения кода: -100

В данном примере будет выведено -Infinity, так как функции не передали параметров:

document.write(Math.min());

Результат выполнения кода: -Infinity

В данном примере будет выведено минимальное значение массива:

var arr = [40, 20, 42, 100, 67];  
document.write(Math.min.apply(null, arr));

Результат выполнения кода: 20

***Функция Math.max***

Функция Math.max возвращает максимальное число из группы чисел, переданных в функцию.

Если в функцию ничего не передано, то будет возращено -Infinity.

По умолчанию функция не работает с массивами, однако с помощью хитрого приема можно найти максимальное значение массива: Math.max.apply(null, arr), где arr - произвольный массив.

См. также функцию Math.min, которая возвращает минимальное число из группы чисел.

Синтаксис

Math.max(число, число, число...);

В данном примере будет выведено максимальное число из группы чисел 1, 5, 10, 34, 100:

document.write(Math.max(1, 5, 10, 34, 100));

Результат выполнения кода: 100

В данном примере будет выведено максимальное число из группы чисел -1, 0, -20, -56, -100:

document.write(Math.max(-1, 0, -20, -56, -100));

Результат выполнения кода: 0

В данном примере будет выведено -Infinity, так как функции не передали параметров:

document.write(Math.max());

Результат выполнения кода: -Infinity

В данном примере будет выведено максимальное значение массива:

var arr = [1, 5, 10, 34, 100];  
document.write(Math.max.apply(null, arr));

Результат выполнения кода: 100

***Функция Math.sqrt***

Функция Math.sqrt возвращает квадратный корень числа.

См. также функцию Math.pow, которое возводит число в заданную степень.

Синтаксис

Math.sqrt(положительное число);

В данном примере будет выведен квадратный корень из 4:

document.write(Math.sqrt(4));

Результат выполнения кода: 2

В данном примере будет выведен квадратный корень из 15:

document.write(Math.sqrt(15));

Результат выполнения кода: 3.872983346207417

В данном примере будет выведен квадратный корень из -100. Так как передано отрицательное число, то будет выведено NaN:

document.write(Math.sqrt(-100));

Результат выполнения кода: NaN

***Функция Math.pow***

Функция Math.pow возводит число в заданную степень. Первым параметром передается число, вторым - в какую степень его возвести.

См. также функцию Math.sqrt, которая извлекает квадратный корень из числа.

Синтаксис

Math.pow(число, степень);

В данном примере число 3 возводится в степень 4:

document.write(Math.pow(3, 4));

Результат выполнения кода: 81

В данном примере число 2 возводится в степень -5:

document.write(Math.pow(2, -5));

Результат выполнения кода: 0.03125

***Функция Math.random***

Функция Math.random возвращает случайное дробное число от 0 до 1.

Синтаксис

Math.random();

Чтобы получить случайно число в определенном промежутке (дробное или целое) следует и пользоваться специальными приемами:

Случайное дробное число между min и max

function getRandomArbitary(min, max) {  
 return Math.random() \* (max - min) + min; }

Случайное целое между min и max

//Использование Math.round() даст неравномерное распределение!

function getRandomInt(min, max) {  
 return Math.floor(Math.random() \* (max - min + 1)) + min; }

В данном примере будет выведено случайное число от 0 до 1:

document.write(Math.random());

Результат выполнения кода: 0.5416665468657356

В данном примере будет выведено случайное целое число от 10 до 100:

function getRandomInt(min, max) {  
 return Math.floor(Math.random() \* (max - min + 1)) + min; }  
document.write(getRandomInt(10, 100));

Результат выполнения кода: 12

***Функция isNaN***

Функция isNaN проверяет, является ли переданный параметр числом или нет.

Функция вернет true, если параметр не является числом и false, если является.

Как она работает: переданный параметр преобразуется к числу. Если это не число (строка, массив и т.п.), то он преобразуется в NaN. Ну, а isNaN проверяет - NaN у нас получился после преобразования или нет.

Учтите, что, к примеру true преобразуется не к NaN, а к числу 1. Есть и другие подобные подводные камни, смотрите их в описании NaN.

Синтаксис

isNaN(значение);

В данном примере будет выведено false, так как параметр является числом:

var num = 13;  
document.write(isNaN(num));

Результат выполнения кода: false

В данном примере будет выведено true, так как параметр не является числом:

var num = 'Я люблю Javascript';  
document.write(isNaN(num));

Результат выполнения кода: true

***Функция isFinite***

Функция isFinite для проверки параметра на то, что он является конечным числом (то есть не строкой, массивом и т.п. и не плюс или минус бесконечностью).

Как работает эта функция: она вернет false, если число является плюс или минус бесконечностью (то есть Infinity или -Infinity) или не числом (то есть NaN), в остальных случаях вернет true.

То есть строки, массивы и т.п. будут преобразованы в NaN и соответственно вернут false.

Однако есть исключения: пустая строка '' возвращает true, строка с пробелами ' ' также возвращает true, null возвращает true, для значений true и false также возвращается true.

Это происходит потому, что эти значения корректно преобразуются в числа (а не в NaN).

Если вам нужна действительно точная проверка на число, которая не считает числом строку из пробелов, логические и специальные значения - используйте следующую функцию isNumeric:

function isNumeric(num) {  
 return !isNaN(parseFloat(num)) && isFinite(num); }

Давайте разберемся, как она работает.

Функция isFinite преобразует параметр к числу и возвращает true, если это не Infinity, -Infinity или NaN.

Таким образом, правая часть отсеет заведомо нечисла, но оставит такие значения как true, false, null, пустую строку '' и строку с пробелами, т.к. они корректно преобразуются в числа.

Чтобы отсеять эти значения, нужна функция parseFloat, которая для true, false, null, '', ' ' вернет NaN.

Так устроена функция parseFloat: она преобразует параметр к строке, т.е. true, false, null становятся "true", "false", "null", а затем считывает из нее число, при этом пустая строка и строка с пробелами дают NaN.

Затем результат parseFloat обрабатывается с помощью !isNaN, чтобы получить true или false вместо NaN.

В результате отсеивается все, кроме строк-чисел и обычных чисел.

Синтаксис

***isFinite(значение);***

В данном примере будет выведено true, так как параметр является числом:

var num = 3;

document.write(isFinite(num));  
Результат выполнения кода: true

В данном примере будет выведено false, так как параметр не является числом:

var num = 'JS';  
document.write(isFinite(num));

Результат выполнения кода: false

В данном примере будет выведено false, так как параметр является бесконечностью:

var num = Infinity;  
document.write(isFinite(num));

Результат выполнения кода: false

В данном примере будет выведено false, так как 1/0 - это по сути Infinity (бесконечность):

var num = 1/0;  
document.write(isFinite(num));

Результат выполнения кода: false

В данном примере будет выведено true, так как пустая строка, не являющаяся числом, - это исключение:

var num = '';  
document.write(isFinite(num));

Результат выполнения кода: true

***Функция parseInt***

Функция parseInt преобразует строку в целое число.

Это нужно для значений типа '12px' - когда вначале стоит число, а потом единицы измерения. Если применить функцию parseInt к '12px', то результатом получится число 12 (и это будет действительно число, а не строка).

Преобразование произойдет, если только целое число стоит в начале строки, иначе будет выведено NaN.

Вторым параметром можно указать систему счисления числа, и функция вернет число, переведенное из указанной системы счисления в десятичную.

Синтаксис

parseInt(строка, [система счисления]);

В данном примере функция преобразует строку в целое число:

var str = '10px';  
document.write(parseInt(str));

Результат выполнения кода: 10

В данном примере при преобразовании дробная часть будет отброшена:

var str = '10.3px';  
document.write(parseInt(str));

Результат выполнения кода: 10

Если вам не нужно отбрасывать дробную часть - используйте функцию parseFloat.

В данном примере функция не смогла прочитать число (так как оно не в начале строки) и вернула NaN:

var str = 'width: 100px;';  
document.write(parseInt(str));

Результат выполнения кода: NaN

***Функция parseFloat***

Функция parseFloat преобразует строку в число с плавающей точкой.

Это нужно для значений типа '12.5px' - когда вначале стоит число, а потом единицы измерения. Если применить функцию parseFloat к '12.5px', то результатом получится число 12.5 (и это будет действительно число, а не строка).

Преобразование произойдет, если только целое число стоит в начале строки, иначе будет выведено NaN.

Вторым параметром можно указать систему счисления числа, и функция вернет число, переведенное из указанной системы счисления в десятичную.

Синтаксис

parseFloat(строка, [система счисления]);

В данном примере функция parseFloat извлечет дробное число из начала строки:

var str = '10.3px';  
document.write(parseFloat(str));

Результат выполнения кода: 10.3

Обратите внимание на то, как в этом случае будет работать функция parseInt - она отбросит дробную часть:

var str = '10.3px';  
document.write(parseInt(str));

Результат выполнения кода: 10

В данном примере функция не смогла прочитать число (так как оно не в начале строки) и вернула NaN:

var str = 'width: 100px';  
document.write(parseFloat(str));

Результат выполнения кода: NaN

**Метод toFixed**

Метод toFixed производит округление числа до указанного знака в дробной части.

Количество знаков указывается параметром.

Если число знаков не указано, то по умолчанию берется 0 знаков, то есть округление до целого числа.

Параметр метода также может быть отрицательным, в этом случае округлятся будут не знаки в дробной части, а знаки в целой части. К примеру, значение -2 округлит число 12345.6789 в число 12300.

См. функции Math.round, Math.ceil, Math.floor, которые также выполняют округление числа.

См. также метод toPrecision, которая выполняет округление числа до определенного количества символов.

Синтаксис

число.toFixed(количество знаков в дробной части);

В данном примере дроби оставлено только 3 знака в дробной части:

var num = 1.1111;  
document.write(num.toFixed(3));

Результат выполнения кода: 1.111

В данном примере дроби также оставлено только 3 знака в дробной части, однако последняя цифра увеличилась на 1, так как округление производится по правилам математики:

var num = 1.1119;  
document.write(num.toFixed(3));

Результат выполнения кода: 1.112

В данном примере дробь округлена до целого числа, так как параметр метода пуст:

var num = 1.111;  
document.write(num.toFixed());

Результат выполнения кода: 1

В данном примере параметр метода превышает количество знаков в дробной части в исходной дроби, поэтому метод добавил две цифры 0 в конец нашей дроби:

var num = 1.1111;  
document.write(num.toFixed(6));

Результат выполнения кода: 1.111100

В данном примере параметром метода передано отрицательное значение - округление будет идти уже по целой части числа:

var num = 12345.6789;  
document.write(num.toFixed(-2));

Результат выполнения кода: 12300

В данном примере параметром метода передано отрицательное значение. По правилам математического округления число 4 будет увеличено на единицу:

var num = 12345.6789;   
document.write(num.toFixed(-1));

Результат выполнения кода: 12350