

Лабораторная работа №2. Функции. Операторы цикла. Объекты Math, Number

Цель: научиться создавать пользовательские функции на JS, использовать в программах операторы цикла; изучить особенности использования встроенных объектов **Math** и **Number**.

Теория

Функции

Функция представляет собой подпрограмму, которую можно вызвать для выполнения, обратившись к ней по имени. Взаимодействие функции с внешней программой, из которой она была вызвана, происходит путем передачи функции параметров и приема от нее результата вычислений. Функция в JavaScript может и не требовать параметров, а также ничего не возвращать.

В JavaScript есть встроенные функции, которые можно использовать в программах, но код которых нельзя редактировать или посмотреть. Пользователь также может создать свои функции для решения конкретных задач – пользовательские.

Описание функции имеет следующий синтаксис:

```
function имя_функции(параметры) {  
  код  
}
```

Функция **messageWrite()** в примере ниже будет выполнена только после нажатия на кнопку. В примере используется событие **onclick**.

```
<html>  
<head>  
<script type='text/javascript'>  
  // Функция выводит текст на страницу  
  function messageWrite() {  
    document.write('Текст выводится на страницу с помощью  
JavaScript!');  
  }  
</script>  
</head>  
<body>  
<input type='button' value='Нажми на меня'  
onclick='messageWrite()' />  
</body>  
</html>
```

Операторы цикла

Цикл – это блок команд, который может повторно выполняться, пока определенное условие не будет выполнено.

JavaScript поддерживает 3 вида циклов: **for**, **while**, **do..while**.

Оператор for

Цикл for исполняет блок команд, пока заданное условие является истинным. Например,

```
for (i=1;i<=30;i++) {  
    document.write (i+'<br />');  
}
```

Оператор while

Цикл while выполняет блок кода, пока заданное условие истинно. Напрмер,

```
var i=1;  
while (i<=30) {  
    document.write (i+'<br />');  
    i++;  
}
```

Оператор do-while

Цикл do..while часто называют циклом с постусловием, потому что в отличие от предыдущих циклов он вначале исполняет блок команд и только потом проверяет заданное условие. Если условие истинно, блок команд выполняется еще раз, если условие ложно, цикл завершает исполнение.

В отличие от оператора while в операторе do-while код выполняется хотя бы один раз, независимо от условия. В примере код выполнится один раз, так как условие ложно.

```
var i=20;  
do {  
    document.write('Я студент (от лат. studens – усердно  
    работающий, занимающийся) ВГТУ');  
}  
while (i<=3);
```

Объекты

Объекты представляют собой программные единицы, обладающие некоторыми свойствами. Об объекте можно судить по значениям его свойств и описанию того, как он функционирует.

Встроенные объекты имеют фиксированные названия и свойства. Все свойства этих объектов разделяют на два вида: просто свойства и методы. Свойства аналогичны обычным переменным. Они имеют имена и значения. Некоторые свойства объектов доступны только для чтения, их значения

нельзя изменять. Другие свойства доступны и для записи – их значения можно изменять с помощью оператора присваивания. Методы аналогичны функциям, они могут иметь параметры или не иметь их.

Таким образом, объект можно понимать как некоторый контейнер, содержащий переменные-свойства и функции-методы.

Для разработчиков веб-сайтов особенно важны объекты String (обработка строк), Array (массивы), Math (математические формулы и константы) и Date (работа с датами).

Объект Math

Встроенный объект **Math** позволяет производить математические операции. Чтобы обращаться к свойствам и методам этого объекта его не нужно предварительно создавать (в отличие от остальных встроенных объектов JavaScript).

Свойства объекта **Math** содержат значения часто используемых математических констант. В примере выводятся на страницу некоторые константы.

```
//Выведем на страницу число Пи
document.write(Math.PI + '<br />');
//Выведем экспоненту
document.write(Math.E + '<br />');
//Выведем натуральный логарифм 10
document.write(Math.LN10 + '<br />');
//Выведем квадратный корень 2
document.write(Math.SQRT2);
```

С помощью методов объекта можно производить над числами различные математические операции. Ниже перечислены методы **Math**:

- `abs(число)` – возвращает модуль (абсолютное значение) числа;
- `acos(число)` – возвращает арккосинус числа;
- `asin(число)` – возвращает арксинус числа;
- `atan(число)` – возвращает арктангенс числа;
- `atan2(x, y)` – возвращает угол в полярных координатах точки;
- `ceil(число)` – округляет число вверх до ближайшего целого;
- `cos(число)` – возвращает косинус числа;
- `exp(число)` – возвращает число *e* в степени число;
- `floor(число)` – округляет число вниз до ближайшего целого;
- `log(число)` – возвращает натуральный логарифм числа;
- `max(число1, число2)` – возвращает большее из 2 чисел;
- `min(число1, число2)` – возвращает меньшее из 2 чисел;
- `pow(число1, число2)` – возвращает число1 в степени число2;
- `random()` – возвращает случайное число между 0 и 1;
- `round(число)` – округляет число до ближайшего целого;
- `sin(число)` – возвращает синус числа;

- `sqrt(число)` – возвращает квадратный корень из числа;
- `tan(число)` – возвращает тангенс числа.

В примере демонстрируется применение методов объекта **Math**. Все результаты выводятся на страницу.

```
<html>
<body>
<script type='text/javascript'>
//Округлим число 25.34 до ближайшего целого
document.write(Math.round(25.34) + '<br />');
//Округлим число 25.88 до ближайшего целого
document.write(Math.round(25.88));
//Выберем из чисел 10 68 35 12 44 максимальное
document.write(Math.max(10,68,35,12,44) + '<br />');
//Выберем из чисел 10 68 35 12 44 минимальное
document.write(Math.min(10,68,35,12,44));
//Возведем 5 в -1 степень
document.write(Math.pow(5,-1));
//Сгенерируем случайное число между 0 и 1
document.write(Math.random() + '<br />');
//Сгенерируем случайное число между 0 и 100 и округлим
document.write(Math.round(Math.random()*100));
</script>
</body>
</html>
```

Объект Number (Число)

В JavaScript числа могут быть двух типов: целые и с плавающей точкой. Числа можно создавать обычным образом с помощью переменных и оператора присваивания, не прибегая к объекту Number. Однако этот объект обладает некоторыми полезными свойствами и методами, которые иногда могут пригодиться.

Пример создания объекта Number:

```
x = new Number('34.21').
```

Методы Number:

- **toExponential(количество)** – представляет число в экспоненциальной форме, параметр *количество* – целое число, определяющее, сколько цифр после точки следует указывать.
- **toFixed(количество)** – представляет число в форме с фиксированным количеством цифр после точки, параметр *количество* – целое число, определяющее, сколько цифр после точки следует указывать.
- **toPrecision(точность)** – представляет число с заданным общим количеством значащих цифр. Параметр *точность* – целое число, определяющее, сколько всего цифр, до и после точки, следует указывать.
- **toString([основание])** – возвращает строковое представление числа в системе счисления с указанным основанием. Если параметр не указан,

имеется в виду десятичная система счисления. Этот метод имеют все объекты.

Например, следующий код выведет на страницу число 3.422e+1:

```
x = new Number(34.215);  
x=x.toExponential(3);  
document.write(x).
```

Задания к лабораторной работе №2

Задание 1. Все выполненные задания из предыдущей работы оформить в виде функций. Создать кнопки и запускать задание на выполнение щелчком по соответствующей кнопке. Поработать над оформлением результатов вывода (цвет шрифта, кегль, гарнитура).

Задание 2. Вывести таблицу умножения, **a** и **b** ввести с клавиатуры. Использовать оператор цикла **for**.

1	2	...	b
2	4	...	2*b
...			
a	a*2	...	a*b

Применить к ячейкам таблицы свойства форматирования.

Задание 3. Найти площадь круга и длину окружности, радиус меняется от **a** до **b** с шагом 0,3. Результаты округлить и вывести в таблице. Использовать оператор цикла **do-while**.

Радиус	Площадь круга	Длина окружности
a		
a+0,3		
...		
b		

Применить к ячейкам таблицы свойства форматирования.

Задание 4. Создать 2 объекта-числа (Number) – дробное и целое. Применить к каждому из них методы: **toExponential**, **toFixed**, **toPrecision**, **toString**. Результаты проанализировать, представить в таблице в следующем виде:

Число	Метод	Результат	Описание метода
127.18	toString(16)	7f.2e147ae147b	Строковое представление числа в 16-ричной системе счисления