

Практическая работа №5

Основы работы с javascript

Цель работы: изучение основ работы со скриптами javascript.

Задачи:

1. Изучить теоретический материал
2. Выполнить практическое задание
3. Ответить на контрольные вопросы.
4. Оформить отчет.

Теоретический материал

Элемент script содержит программусценарий. Как правило, такая программа бывает написана на языке JavaScript или VBScript. При помощи языков сценариев можно значительно обогатить возможности HTML-страниц.

Языки сценариев интерпретируются клиентскими приложениями — браузерами. Программа на таком языке не компилируется, программный код доставляется браузеру в таком виде, как он есть. В процессе загрузки код читается и, при обращении к программе или ее части, этот код выполняется встроенным интерпретатором.

Языки JavaScript и VBScript — это объектно-ориентированные языки. В них существуют средства описания и создания объектов. Объектные модели таких языков, как JavaScript и VBScript, порой сильно отличаются от объектных моделей языков C++ или Java. Основное отличие состоит в том, что в JavaScript не существует понятия абстрактного класса. Каждый объект всегда имеет свою конкретную реализацию, нет абстрактных классов и соответствующих им экземпляров классов. Объекты могут быть созданы программистом с использованием средств языка. Помимо этого, программист может воспользоваться заранее определенными объектами, т. е. такими объектами, которые встроены в программу-браузер и всегда доступны для скриптов, а также объектами, которые создаются программой-браузером автоматически при загрузке HTML-страницы и структура которых описана в этой HTML-странице. Как правило, каждому HTML-элементу соответствует некоторый объект. К объекту можно обращаться из программы, как к элементу массива (в браузерах всегда существует массив, состоящий из всех объектов данной страницы), указав индекс элемента этого массива. К нему можно обратиться так же, как к элементу конкретного массива, т. е. массива, состоящего из объектов данного типа (существует несколько массивов для различных объектов). Причем обращение к элементам массива может быть как к элементам ассоциативного массива (указывая в качестве параметра имя объекта), так и задавая числовой индекс элемента массива. Наконец, к объекту можно обращаться по его имени, если это имя было задано в виде свойства id или name в соответствующем ярлыке при описании HTML-элемента. Как правило, свойства объектов совпадают со свойствами соответствующих им HTML-элементов.

Языки сценариев способны обрабатывать события. Примерами событий являются щелчок мышью, завершение загрузки страницы, перемещение мыши по заданной области, наведение фокуса на тот или иной элемент страницы и т. п. С наступлением того или иного события может быть связана определенная функция. Понятно, что такой механизм позволяет значительно расширить возможности HTML-разметки. Существует возможность активно изменять содержание и внешний вид страницы в зависимости от того, что делает пользователь. При этом, в процессе "общения с пользователем", с помощью языков сценариев мы можем кардинальным образом изменить текущую страницу без обращения к серверу, мы можем, например, создать новые объекты, уничтожить ранее созданные, проверить введенную информацию, заставить изображения двигаться и менять размер (языки сценариев могут работать с таймером).

Текст программы на языке JavaScript вставляется непосредственно в файл HTML-страницы.

Чтобы посмотреть, как это выглядит, приведем пример простейшей HTML-страницы (см. пример файла html.html).

```

<HTML>
<body>
<br>
<h1 align=center>Это обычный HTML-документ</h1>
<br>
<script language="JavaScript">
document.write("Это и есть JavaScript!" )
</script>
<br>
А это простой HTML.
</body>
</HTML>

<body>
<br>
<h1 align=center> Это обычный HTML-документ</h1>
<br>
<script language="JavaScript">
document.write("Это и есть JavaScript!" )
</script>
<br>
А это простой HTML.
</body>
</HTML>

```

В языке программирования JavaScript имеются три основных типа данных;

- строка (string)
- число (number)
- булевоe выражение (boolean)

Строка-это связанный набор символов, включающий в себя буквы, цифры и знаки препинания. В JS строка чаще всего представляет собой какой-либо текст. Строки в JS заключаются в двойные или одиночные кавычки. Например: “Добро пожаловать!”. Двойные кавычки могут содержаться внутри строки, заключенной в одинарные кавычки и наоборот. Строка, не содержащая символов, называется пустой строкой и обозначается пустыми кавычками.

Операция конкатенации строк (+).

Пример. A=“Java”, B=“Script”, A + B = “JavaScript”

Для преобразования строковой переменной в числовое используется функция Number, для обратного преобразования используется функция String.

Пример.

Var X1=56, X2=11;

Y1=string(X1);

Y2=string(X2);

Txt1.value=Y1+Y2;

Ответ будет 5611.

Объект String позволяет изменять и форматировать текстовые строки и выделять внутри них части строк.

Метод toUpperCase переводит текст в верхний регистр.

Метод toLowerCase переводит текст в нижний регистр.

Пример.

```
<script Language="Javascript">
```

```
X1=prompt (“Введите текст”, “Здесь!”);
```

```
alert (X1.toUpperCase());
```

```
</Script>
```

Строка текста будет переведена в верхний регистр.

Метод `substring` используется для выделения части строки на указанных позициях внутри объекта `String`.

Пример.

```
<script Language="Javascript">
X2="Привет всем!"
alert (X2.substring(0,6));
</Script>
```

Первое число в скобках соответствует начальной позиции подстроки (нумерация ведется с нуля). Второе число соответствует конечной позиции выделяемой подстроки.

Числа.

JS воспринимает два типа чисел. Это целые числа и числа с плавающей точкой. Числа с плавающей точкой можно представлять в экспоненциальном виде, например: $3.76e2$. Число e обозначает "10 в степени". Например: $4.52e2=452$; $1e-3=0,001$.

Рассмотрим примеры объявления переменных.

```
<script language = "JavaScript">
<var S1 ;
var X=25;
var FIT="Факультет информационных технологий";
var m1,m2, n1, n2
var ms1="Привет", ms2="всем", num1=6 num2=54;
```

Основные операции в Javascript

Арифметические операции

1. Сложение (+) $a + b$
2. Вычитание (-) $a - b$
3. Деление (/) a / b
4. Умножение (*) $a * b$
5. Остаток от деления (%) $a \% b$

Пример.

```
var a = 6, b = 4;
alert (a + b);
```

На экране увидим число 10

Пример.

```
var a = 6, b = 4;
alert (a % b);
```

На экране увидим число 2.

Операции сравнения

1. меньше (<) $a < b$
2. меньше или равно (<=) $a <= b$
3. больше (>) $a > b$
4. больше или равно (>=) $a >= b$
5. равно (==) $a == b$
6. не равно (!=) $a != b$

Логические операции

1. Логическое И (and) &&
2. Логическое ИЛИ (OR) ||
3. Логическое НЕ (not) !

Логическое ИЛИ и И – операции с двумя операндами, а логическое НЕ – операция с одним операндом.

Логическое И означает, что обе части выражения должны быть истинны.

Логическое ИЛИ означает, что хотя бы одна часть выражения должна быть истинной.

Логическое НЕ изменяет значение истина / ложь на обратное.

Операции с одним операндом

1. Префиксное и постфиксное возрастание (increment)
(++) ++ a ; a++

2. Префиксное или постфиксное уменьшение (decrement)

(- -) - - a ; a - -

3. унарный плюс (+)

4. унарный минус (-)

Унарный плюс используется для преобразования операнда в число.

Унарный минус меняет знак операнда на противоположный.

Пример.

Если a = 5, то ++ a + 2 = 8; - - a + 2 = 6;

a++ + 2 = 7; a - - + 2 = 7.

Комбинированные операции

X += 3 равноценно выражению x = x + 3

X -= 3 равноценно выражению x = x - 3

X *= 3 равноценно выражению x = x * 3

X /= 3 равноценно выражению x = x / 3

Функция eval.

Функция eval выполняет указанные вычисления

Пример

Y=eval(x1+op+x2);

Где x1 и x2-числовые выражения

op-тип операции (+,-,/,%)

Пример.

Var x1=6, x2=9;

Op="-";

Txt1.value=eval(x1+op+x2);

Ответ будет -3.

Если op="/", ответ 0.6666

Если op="*", ответ 54.

Пример страницы, которая складывает четные числа в диапазоне от 1 до N.

<html>	
<head>	<H1> Задание №1 </H1>
<title> Лабораторная работа №10 </title>	<hr>
</head>	
<body>	
<script language="JavaScript">	<p> Сложить четные числа в диапазоне от 1
<!-- hide	до N. N задается с клавиатуры. </p>
function count() {	
var N=num.value;	
var result=0;	
for (var i=1;i<=N;i++)	<p> Введите число </p>
{	<input type=text value="" name="num">
if (i%2==0)	
result=result+i;	<input type=button value="Расчет"
}	name="make" onclick="count()">
rez.value=result;	<p> Результат </p>
}	<input type=text readonly=true value=""
//-->	name="rez">
</script>	</body>
	</html>

Практические задания

Разработать web-страницу с использованием языка JavaScript для решения следующих задач:

№	Задание
1.	Написать программу, которая печатает true или false в зависимости от того, имеют три заданных числа одинаковую четность или нет Написать программу, которая вычисляет площадь правильного треугольника, вписанного в окружность заданного радиуса
2.	Вычислить значение $f(x)=x+\ln x$ в заданной точке. Вывести на печать таблицу из 5 значений функции Вычислить длину окружности, площадь круга и объем шара заданного радиуса
3.	Дана длина ребра куба. Найти объем куба и площадь его боковой поверхности. Найти произведение цифр заданного четырехзначного числа
4.	Определить число, полученное записыванием в обратном порядке цифр заданного трехзначного числа Даны три числа, определить, можно ли построить треугольник с такими длинами сторон
5.	Дано натуральное число n, выяснить входит ли цифра 3 в запись этого числа и поменять порядок цифр числа n на обратный Определить, есть ли среди цифр заданного трехзначного числа одинаковые
6.	Определить, равна ли сумма двух первых цифр заданного четырехзначного числа сумме двух его последних цифр Дана длина ребра куба. Найти объем куба и площадь его боковой поверхности.
7.	Треугольник задан величинами углов и радиусом описанной окружности. Найти стороны треугольника Даны 3 вещественных числа. Определить, образуют ли они возрастающую последовательность
8.	Найти площадь сектора, радиус которого равен r, а дуга содержит f радиан Дано действительное число a, за четыре операции умножения получить a^3 и a^{10} , вывести все промежуточные результаты в виде таблицы

Содержание отчета

1. Титульный лист
2. Цели, задачи работы
3. Текст задания
4. Листинг
5. Результаты работы