Лабораторная работа 2. Анимация на JS

Цель: изучить возможности управления процессами во времени:

Методы:

setInterval()

setTimeout();

освоить приемы реализации перемещения и движения объектов;

научиться строить и отображать графики.

Теория

Управление процессами во времени

Для организации постоянного периодического (через заданный интервал времени) выполнения некоторого выражения или функции служит метод setInterval() объекта window. Этот метод имеет следующий синтаксис:

setInterval (выражение, период [, язык]).

Первый параметр – строка, содержащая выражение (обычно вызов функции). Второй параметр – целое число, указывающее временную задержку в миллисекундах. Третий, необязательный па-раметр, указывает язык, на котором написано выражение; по умолчанию – JavaScript. Метод setInterval() возвращает некоторое целое число – идентификатор временного интервала, который может быть использован в дальнейшем для прекращения процесса.

В следующем примере функция myfunc() выполняется периоди-чески через 0,5 с.

setInterval("myfunc()", 500)

Для остановки запущенного временного процесса служит метод clearInterval(идентификатор), который принимает в качестве па-раметра целочисленный идентификатор, возвращаемый соответ-ствующим методом setInterval(), например:

var myproc = setInterval("myfunc () , 100")

if (confirm("Прервать процесс ?"))

clearInterval(myproc)

Чтобы выполнить выражение с некоторой временной задерж-кой, используется метод setTimeout() объекта window, который имеет следующий синтаксис:

setTimeout(выражение, задержка [, язык]).

Первый параметр – строка, содержащая выражение (обычно вызов функции). Второй параметр – целое число, указывающее временную задержку в миллисекундах. Третий, необязательный па-раметр указывает язык, на котором написано выражение; по умол-чанию – JavaScript. Метод setTimeout() возвращает некоторое целое число – идентификатор временного интервала, который может быть использован в дальнейшем для отмены задержки выполнения про-цесса.

Для отмены задержки процесса, запущенного с помощью мето-да setTimeout(), используется метод clearInterval(идентификатор), который принимает в качестве параметра целочисленный иденти-фикатор, возвращаемый соответствующим методом setTimeout.

В следующем HTML-документе имеются две кнопки. Щелчок на кнопке Пуск открывает через 5 с новое окно и загружает в него до-кумент mypage.htm. Это действие можно отменить с помощью кнопки Отмена, если щелкнуть на ней, пока окно еще не открыто:

<HTML>

<BUTTON ID="start">Пуск</BUTTON>

<BUTTON ID="stop">Отмена</BUTTON>

<SCRIPT>

var myproc

function start.onclick(){

myproc = setTimeout ("window.open('mypage.htm')", 5000)

}

function stop.onclick(){

clearTimeout(myproc)

}

</SCRIPT>

</HTML>

Перемещение графических объектов

Перемещать мышью изображения можно различными способами. Изучим один из них: пользователь пытается перетащить мы-шью изображение; затем он должен отпустить кнопку мыши и пе-реместить указатель в нужное место; остановившись в нужном ме-сте, пользователь отпускает кнопку мыши или щелкает ею, чтобы прекратить перемещение изображения. Код перемещения изобра-жения мышью:

<НТМL>

<НЕАD><TITLE>Перемещаемое изображение</TITLE></НЕАD>

<BODY id = "mybody">

<IMG ID="myimg" SRC = "pict.gif ondragstart = "drag()" style = "position:absolute; top:10; left:10">

</BODY>

<SCRIPT>

flag = false // нельзя перемещать

var id\_img = ""

function drag() {

flag = mag

id\_img = event.srcElement.id

}

function mybody.onmousemove(){

if (flag){ // если можно перемещать

document.all[id\_img].style.top = event.clientY

document.all[id\_img].style.left = event.clientX } }

function mybody.onmouseup(){

flag = false // нельзя перемещать

}

</SCRIPT>

</HTML>

Здесь функция drag(), обрабатывающая событие ondragstart (попытка перетаскивания), устанавливает переменную-триггер flag и выясняет, кто инициатор события. Значение переменной flag позво-ляет определить, можно или нельзя перемещать элемент. В данном примере инициатором события может быть только один элемент. События onmousemove (перемещение указателя мыши) и on-mouseup (кнопка мыши отпущена) получает не изображение, а объ-ект тела документа mybody.

Перемещение текстовых областей

Рассмотрим пример HTML-документа, в котором можно пере-ме¬щать текстовые области, созданные с помощью тегов <TEX-TAREA>. Размеры области и шрифта текста определяются в ней в зависимости от значения вертикальной координаты.

<HTML>

<НЕАD><ТITLE>Перемещаемые текстовые облас-ти</ТITLE></НЕАD>

<BODY id = "mybody" background = "blue.gif">

<TEXTAREA ID="tl" ondblclick=""drag()" STYLE = "posi-tion:absolute;

top:10; left: 10; fon-size: large"> Это – первый текст</TEXTAREA>

<TEXTAREA ID="t2" ondblclick=""drag()" STYLE = "posi-tion:absolute;

top:100; left:150"> Это – второй текст </TEXTAREA>

<TEXTAREA ID="t3" ondblclick=""drag()" STYLE = "posi-tion:absolute; top:150; left:250"> Это - третий текст </ТЕXTAREA>

</BODY>

<SCRIPT>

resizetext() { // установка размеров текстовых областей

var flag = false }

var id\_img = ""

function drag() {

flag = !flag

id\_img = event.srcElement.id //id элемента, который надо перемещать

function mybody.onmousemove() {

if (flag){

document.all[id\_img].style.top = event.clientY

document.all[id\_img].style.left = event.clientX

resizetext() }} // установка размеров текстовых обла-стей

function mybody.onmouseup() {

flag = false }

function resizetext () { // установка размеров обла-стей

var у, size, idimg, idtext

for (i =0 ; i < document.all.length; i++) {

if (document.all[i].tagName=='TEXTAREA') {

idtext = document.all[i].id

у = parseInt(document.all[idtext].style.top)

size = Math.min(y, 800)

size = Math.max(size, 60)

document.all[idtext].style.width = size

document.all[idtext].style.height = 0.8\*size

document.all[idtext].style.zIndex = у

document.all[idtext].style.fontSize = Math.max(2, y/10)

</SCRIPT>

</HTML>

Схема сценария, осуществляющего непрерывное перемещение видимого элемента документа, имеет следующий вид:

Function init\_move() { // инициализация движения

... // подготовка к запуску функции move()

setInterval("move()", задержка)

function move(){

.../\* изменение координат top и left стиля перемеща-емого элемента \*/

// вызов функции для перемещения элемента

Cоздаются две функции: init\_move(), которая осуществляет подготовку исходных данных и вызывает метод setInterval() с ука-занием в качестве первого параметра имени второй функции move() в кавычках, и move(), которая изменяет координаты элемента. По-скольку метод setInterval() вызывает функцию move() периодически через заданное количество миллисекунд, то координаты элемента изменяются постоянно. При этом создается эффект движения. Ско-рость и плавность движения зависят от величин приращения коор-динат (в функции move()) и временной задержки (второго парамет-ра метода setInterval()). Чтобы начать перемещение элемента надо вызвать первую функцию init\_move().

Остановка движения

Сценарий с запуском и остановкой движения может выглядеть, например, следующим образом:

function init\_move(xid, dx, dy) {

var prmstr = "'"+xid+" ,"+dx+" ,"+dy // строка па-раметров для move()

prmstr = "move(" +prmstr+ ")"

idjnove = setInterval (prmstr , 200) /\* сохраняем идентификатор движения \*/

}

function move(xid, dx , dy) {

у = parseInt(document.all[xid].style.top)

x = parseInt(document.all[xid].style.left)

document.all.myimg.style.top = у+dy

document.all.myimg.style.left = x+dx

if (parseInt(document.all[xid].style.left) > 350) /\* остановка по условию \*/

}

clearInterval(id move)

init\_move(“myimg”, 10, 5) // начинаем движение

В этом примере движение остановится, как только горизон-тальная координата элемента превысит 350 пикселов.

Движение по произвольной кривой

Рассмотрим задачу организации движения видимого элемента по произвольной кривой, заданной выражением с одной перемен-ной. Функция движения в качестве параметров будет иметь два вы-ражения, которые описывают изменения вертикальной и горизон-тальной координат элемента. Эти выражения будут содержать одну переменную, которую мы обозначим через х – строчной латинской буквой. Переменную х можно интерпретировать как независимый параметр движения (например, время). С помощью встроенной функции eval() можно вычислить значения этих выражений при конкретном значении переменной х и присвоить их параметрам left и top таблицы стилей перемещаемого элемента. Функция (пусть это будет move()), которая все это выполняет, передается в качестве первого параметра методу setInterval(), который периодически вы-зывает ее через заданный интервал времени.

Функция инициализации движения curvemove() принимает три строковых параметра (ID перемещаемого элемента, выражение для вертикальной координаты и выражение для горизонтальной коор-динаты) и один числовой параметр (период времени, через который координаты элемента пересчитываются). Ниже приводятся опреде-ления функций curvemove() и move():

function curvemove(xid, уехрr, xexpr, ztime) {

/\* Движение по произвольной кривой.\*/

if (!xid) return null

if (!yexpr) yexpr = "x"

if (!xexpr) xexpr = "x"

if (!ztime) ztime = 100 // интервал времени, мс

x = 0 /\* глобальная переменная, входящая в выражения уехрr и хехрr \*/

setInterval("move('"+xid+" ', + уехрг + " "+ xexpr +"' ztime)

}

function move(xid, yexpr, xexpr) {

x++

document.all[xid].style.top = eval(yexpr)

document.all[xid].style.left = eval(xexpr)

}

Параметры:

 хid – id движущегося объекта, строка;

 уехрr – выражение для вертикальной координаты;

 хехрr – выражение для горизонтальной координаты;

 ztime – интервал времени между вызовами функции move(), мс.

Чтобы сделать переменную х глобальной, не надо использовать ключевое слово var перед первым ее появлением в коде.

Ниже приведен пример HTML документа с движущимся изоб-ражением:

<НТМL>

<IMG ID = "myimg" SRC = "pict1.gif" STYLE = "posi-tion:absolute">

<SCRIPT>

function curvemove(xid, yexpr, xexpr, ztime) {

// код определения функции

}

function move(xid, yexpr, xexpr) {

// код определения функции

}

Curvemove()'myimg' , "100 + 50\*Math.sin(0.03\*x)" , "50 + х", 100)

</SCRIPT>

</HTML>

В этом примере изображение будет перемещаться по синусоиде с 50 пикселов и горизонтальной скоростью 10 пикселов в секунду. Начальные ординаты графического объекта равны 100 и 50 пиксе-лов по вертикали и горизонтали соответственно.

Рисование линий

В JavaScript нет специальных встроенных средств для рисова-ния произвольных линий. Для решения этой задачи ужно вывести на экран изображение размером 1x1 пиксел, залитое цветом, отли-чающимся от цвета фона. Это изображение следует разместить не-сколько раз в соответствии с координатами, которые задаются па-раметрами позиционирования top и left атрибута STYLE тега <IMG>. С помощью сценария можно сформировать строку, содер-жащую теги <IMG> с не¬обходимыми атрибутами, а затем записать ее в документ методом write().

Рисование прямой линии

Для отображения точки в HTML можно использовать следую-щий тег:

<IMG SRC = "point.bmp" STYLE = "position:absolute; top:y; left:x">

Здесь point.bmp – имя графического файла, содержащего один пиксел; у, х – числа, указывающие положение графического файла в пикселах. Изображение точки размером 1x1 пиксел можно создать в любом графическом редакторе. Из соображений экономии его лучше всего сохранить в файле формата BMP, а не JPEG или GIF (при малых размерах изображения алгоритмы сжатия неэффектив-ны).

Чтобы задать размеры отображения точки на экране, следует использовать атрибуты WIDTH и HEIGHT (ширина и высота):

<IMG SRC = "point.bmp" STYLE = "position: absolute; top:y; left:x" WIDTH=n HEIGHT=n>

Одинаковые значения атрибутов WIDTH и HEIGHT задают представление точки в виде квадрата размером n×n пикселов. При этом точка с исходными размерами 1x1 пиксел просто растягивает-ся. Таким образом, имеется возможность задать отображаемые размеры (масштаб) одной точки, а, следовательно, и определить толщину линии.

Далее следует пример определения функции, которая рисует прямую линию с заданными координатами xl, yl, х2, у2 и толщиной линии n.

Function line(xl, yl, х2, у2 , n){

/\*xl, yl – начало линии, х2, у2 – конец, n – толщина \*/

var clinewidth = " WIDTH=" + n + "HEIGHT=" + n /\* строка для учета толщины \*/

var xstr = "" // строка тегов для записи в HTML-документ

var xstr0 = ' <IMG SRC="point.bmp"' + clinewidth + ' STYLE = "positlon:absolute; '

var k = (y2 – yl)/(x2 – xl) // коэффициент наклона линии

var x = xl // начальное значение координаты х

/\* Формирование строки, содержащей теги <IMG. . . >: \*/

while (x <= х2) {

xstr += xstr9 + 'top:' + (yl + k\* (x – xl)) + ': left:1 + x + '

x+ +

// запись в документ

Рисование кривой линии

Функция curve() для рисования кривой принимаtn следующие параметры: имя графического файла с изображением точки (может быть любое изображение), выражение, задающее кривую, коорди-наты начала линии, количество точек линии, толщина линии и дли-на штриха (если потребуется штриховая линия). Далее следует код функции curve() для рисования кривых.

function curve(pict\_file,yexpr, x0, y0, t, n. s){

/\* pict\_file - имя графического файла

yexpr - выражение с переменной х

х0, у0 - координаты начала кривой

t - количество точек кривой (значений переменной х)

n - толщина линии

s - длина штриха и паузы \*/

if (!yexpr) return null

if (!pict\_file) pict\_file = "point.bmp"

if (!s) s = 0

if (!t) t = 0

var clinewidth = ''

if (!n)

clinewidth = 'WIDTH=' + n + 'HEIGHT=' + n

var x

xstrG = '<IMGSRO "' + pict\_file + '"'' + clinewidth + STYLE = "position:absolute;top: '

xstr = ""

var i = 0, draw = true

for(x = 0; x < t; x++) {

if (draw)

xstr += xstr0 + (y0 + eval(yexpr)) + '; left:' + (x0 + x)

if (i > s&&s > 0) {

draw = !draw

i = 0

}

i ++

}

document .write (xstr) // запись в документ

Задание 1.

Текст – привет, абитуриенты ! ( типографика – авторская),

Изображение- некая школа , и наш универ ( отдельные файлики) – нарисовать или взять))) сами знаете где .

НЕ забываем, что м ы дизайнеры, и белый фон как в предыдущей лабораторной работе не приветствуется . ДИЗАЙН СТРАНИЦЫ!!!!! ДИЗАЙН СТРАНИЦЫ!!!!! ДИЗАЙН СТРАНИЦЫ!!!!!

Что сделать: перетащить мышью область школы на универ . в случае попадания- текст – « «поступил , ура! «

Задание2.

Нарисовать график функции- вывод на страницу, выбрав функцию с помощью радио кнопки , (вернулись к первому заданию), и после ввода данных (диапазон) нарисовать на странице график