

БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ

Кафедра «Автоматизированные системы управления»

ТЕХНОЛОГИЯ ИНТЕРНЕТ- ПРОГРАММИРОВАНИЯ

*Методические указания к лабораторной
работе № 9 для студентов специальности*
**09.03.01 Автоматизированные системы обработки
информации и управления**

Могилев 2020

Лабораторная работа №9

Обработка изображений

Цель работы: Создание HTML страницы с применением CSS

Порядок выполнения работы.

Изучить теоретические сведения.

Выполнить задание к лабораторной работе в соответствии с вариантом.

Оформить отчет.

Требования к отчету.

Цель работы.

Постановка задачи.

Текст программы.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1 Циклическая смена изображений

Необходимо написать сценарий, в результате выполнения которого несколько заданных изображений последовательно появляются в документе через равные промежутки времени. При нажатии на кнопку **Остановить** чередование изображений прекращается. Возобновить просмотр рисунков можно, если нажать кнопку **Начать снова** (рис. 1).

Событие load возникает в тот момент, когда обозреватель заканчивает загрузку окна. Как реакция на событие load вызывается функция `setTimeout`. Считается, что файлы с изображениями хранятся в той же папке, что и документ со сценарием.

Метод `setTimeout` выполняет действие, задаваемое первым параметром, по истечении указанного в миллисекундах промежутка времени, определенного вторым параметром. В рассмотренном примере в качестве первого параметра задается функция `setTimeout` (), тем самым обеспечивается повторение вызова функции через каждые две секунды. Просмотр изображений будет прекращен после нажатия кнопки **Остановить**. Если просмотр изображений требуется продолжить, то следует щелкнуть по кнопке **Начать снова**.

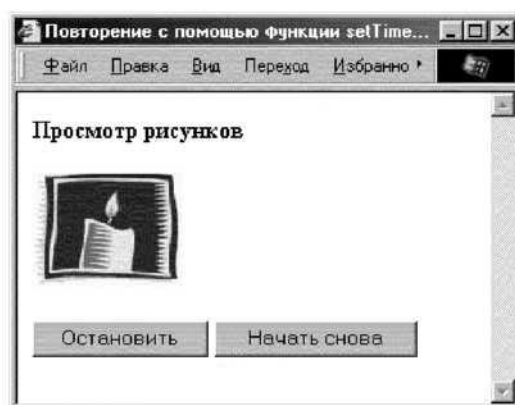


Рис. 1. Чередование рисунков

Страница содержит объекты, зависящие от ее содержания, и эти объекты являются наследниками объекта `document`. Ссылка на объект может быть осуществлена по имени, заданному параметром `name` тега `<html>`. Изображение в документе может быть задано с помощью тега ``, например, так:

```
<IMG src="m1.gif" name="mypict">
```

Доступ к объекту `image` в этом случае может быть осуществлен следующим образом:

```
document.mypict
```

Документ с описанным сценарием выглядит так, как представлено в листинге 1.

Листинг 1 – Использование функции `setTimeout`

```
<HTML>
  <HEAD>
    <TITLE>Повторение с помощью функции setTimeout</TITLE>
    <script language="JavaScript">
      <!-- //
        var k=1
        function ref ()
          {k=5}
        function succpict ()
```

```

{ var d= document
  if (k<=4)
    {if (k==1)
      {d.mypict.src="m1.gif"; k++;}
    else
      if (k==2)
        {d.mypict.src="m2.gif"; k++;}
      else
        if (k==3)
          {d.mypict.src="m3.gif"; k++;}
        else
          if (k==4)
            {d.mypict.src="m4.gif"; k=1}
          setTimeout("succpict()", 2000)
        }
      }
    }
  //-->
</script>
</HEAD>
<BODY onLoad="succpict ()">
  <P>Просмотр рисунков</P>
  <IMG src="m1.gif" name=mypict width=100>
  <FORM name="form1">
    <input type="reset" value=Остановить onClick=ref()>
    <input type="button" value="Начать снова" onClick="k=1;
      succpict()">
  </FORM>
</BODY>
</HTML>

```

2 Смена изображений при наведении указателя мыши

Напишем сценарий, во время работы которого смена рисунков происходит при наведении курсора мыши на изображение.

При перемещении пользователем курсора мыши над изображением возникает событие `Mouseover`. Опишем функцию `succpict ()`, которая будет реакцией на это событие, и определит, какое изображение следует поместить в документ. Воспользуемся тем фактом, что загружаемые изображения хранятся в файлах с именами `m1.gif`, ..., `m4.gif`. Для загрузки k -го изображения формируется имя файла по формуле `"m"+k+".gif"`. Приведем сценарий в листинге 2.

Листинг 2 – Смена изображений при попадании курсора на рисунок I

```
<HTML>
<HEAD>
  <TITLE>Смена рисунков при перемещении курсора мыши
    над изображением</TITLE>
  <script language="JavaScript">
    <!-- //
      var k=1
      function succpict ()
      { var d= document
        if (k < 4)
          k=k+1
        else k=1
        d.mypict.src="m"+k+".gif"
      }
    //-->
  </script>
</HEAD>
<BODY>
  <H4>Для смены изображения поместите курсор мыши над рисунком.</H4>
  <IMG src="m1.gif" name="mypict" width=150 onMouseOver="succpict ()">
</BODY>
</HTML>
```

3 Эффект визуального удаления изображения

Напишем сценарий, во время работы которого при наведении курсора мыши на изображение оно начинает удаляться от зрителя, уменьшаясь в размерах.

При решении такой задачи воспользуемся свойством `width` объекта `image`. При каждом вызове функции `succpict ()` изменяется размер выводимого

изображения и этим достигается эффект удаления от зрителя. Функция `setTimeout("succpict()", 500)` производит повторный вызов функции `succpict` о через каждые полсекунды. Когда размер изображения уменьшится до заданного, движение прекратится. В начальный момент документ имеет вид как на рис. 2.



Рис. 2– Эффект удаления от зрителя

Для каждого рисунка параметры уменьшения размера изображения и время обновления следует подобрать индивидуально. Сценарий описан в листинге 3.

Листинг 3 – Эффект удаления изображения от зрителя

```
<HTML>
<HEAD>
  <TITLE>Удаляющееся изображение</TITLE>
  <script language="JavaScript">
    <!-- //
      function succpict ()
      { var d=document
        var w=d.mypict.width
        if (w>150)
          {d.mypict.width=w-10
            d.mypict.src="msm.jpg"
            setTimeout("succpict()", 500)
          }
      }
    //-->
```

```
        </script>
</HEAD>
<BODY>
    <h4>При наведении курсора мыши над рисунком
        изображение начинает удаляться от зрителя.</H4>
    <IMG src="msm.jpg"  name=mypict  onMouseOver="succpict()">
</BODY>
</HTML>
```

4 Эффект визуального приближения изображения

Напишем сценарий, при выполнении которого заданное изображение начинает увеличиваться, т. е. моделируется эффект приближения изображения.

Эта задача в некотором смысле является обратной к только что рассмотренной. Будем увеличивать при каждом вызове функции размер изображения до тех пор, пока оно не достигнет заданного размера. Повторные вызовы функции `grpict()` обеспечиваются функцией `setTimeout`, параметры которой следует подобрать в зависимости от изображения. В начальный момент документ имеет вид как на рис. 3.

Текст сценария и документ, его содержащий, приведены в листинге 4.

Листинг 4 – Эффект приближения изображения

```
<HTML>
  <HEAD>
    <TITLE>Увеличивающееся изображение</TITLE>
    <script language="JavaScript">
      <!-- //
        function grpict ()
        { var d= document
          var w= d.mypict.width
          if (w < 300)
            { d.mypict.width=w + 10;
              d.mypict.src="mgr.jpg"
              setTimeout("grpict()", 500)
            }
          }
      //-->
    </script>
  </HEAD>
  <BODY>
    <P>При наведении курсора мыши над рисунком
      изображение начинает приближаться к зрителю.</P>
    <IMG src="mgr.jpg" name=myspict width=100 onMouseOver="grpict()">
  </BODY>
</HTML>
```




Рис. 3 – Эффект приближения изображения

5 Вертикальное графическое меню со стрелкой

Необходимо написать сценарий, реализующий вертикальное графическое меню. При наведении курсора мыши на пункт меню слева от выделенного пункта появляется стрелка, как изображено на рис. 4.

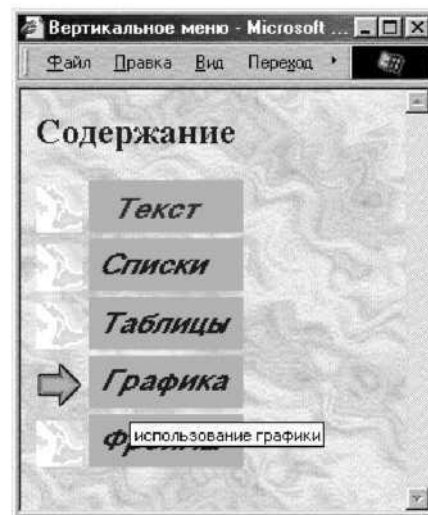


Рис. 4 – Вертикальное графическое меню со стрелкой

В файле с именем pch1.gif хранится изображение, соответствующее первому пункту меню, в файле с именем pch2.gif — второму пункту меню и т. д. При выборе некоторого пункта меню слева от изображения, соответствующего выбранному пункту, появляется стрелка. Если курсор мыши выходит из области изображения пункта меню, то стрелка исчезает. Изображение стрелки хранится в файле с именем but1.gif, а в файле but0.gif размещается изображение, совпадающее с цветом фона, и используемое тогда, когда пункт меню не выбран. В документе располагается десять изображений по два на отдельной строке. Пары изображений, которые нельзя размещать на разных строках, заключаются в теги <nobr> и </nobr>. Первая пара изображений соответствует первому пункту меню и представлена в документе следующим образом:

```
<nobr> <IMG src="but0.gif" width="31" height="31" name="p1">
      <A href="tchl.htm" onmouseover="IMG(p1,true)"
          onmouseout="IMG(p1,false)">
      <IMG src="pch1.gif" alt="форматирование текста" border="0"
          width="103" height="35"></A>
</nobr> <br>
```

При попадании курсора мыши в область изображения возникает событие Mouseover, параметр обработки события onMouseOver получает значение IMG(p1,true).

Назначение функции img() помещать или убирать стрелку слева от изображения, соответствующего выбранному пункту меню. Данная функция описывается следующим образом.

```
function IMG (pict, action)
{ if (action)
  {pict.src="but1.gif" }
  else
  { pict.src="but0.gif" }
}
```

Какой пункт меню выбран, задается первым параметром, второй параметр определяет, помещать или убирать стрелку с помощью загрузки изображения из файла but0.gif или but1.gif.

Документ со сценарием, реализующим графическое меню, представлен в листинге 5.

Листинг 5 – Вертикальное меню со стрелкой

```

<HTML>
  <HEAD>
    <TITLE>Вертикальное меню со стрелкой</TITLE>
    <script language="JavaScript">
      function IMG (pict, action)
      {
        if (action)
        { pict.src="but1.gif" }
        else
        { pict.src="but0.gif" }
      }
    </script>
  </HEAD>
  <BODY background="fon1.jpg">
    <H2><FONT color="#0000FF">Содержание</FONT></H2>
    <nobr> <IMG src="but0.gif" width="31" height="31" name="p1">
      <A href="tchl.htm" onmouseover="IMG(p1,true)"
        onmouseout="IMG(p1,false)">
        <IMG src="pch1.gif" alt="форматирование текста" border="0"
          width="103" height="35"></A>
    </nobr> <br>
    <nobr><IMG src="but0.gif" width="31" height="31" name="p2">

```

```

<A href="tch2.htm" onmouseover="IMG(p2, true)"
        onmouseout="IMG(p2, false)">
<IMG src="pch2.gif" alt="создание списков" border="0" width="103"
        height="35"></A>
</nobr> <br>
<nobr><IMG src="but0.gif" width="31" height="31" name="p3">
    <A href="tch3.htm" onmouseover="IMG(p3, true)"
        onmouseout="IMG(p3, false)">
    <IMG src="pch3.gif" alt="построение таблиц" border="0" width="103"
        height="35"></A>
</nobr> <br>
<nobr><IMG src="but0.gif" width="31" height="31" name="p4">
    <A href="tch4.htm" onmouseover="IMG(p4, true)"
        onmouseout="IMG(p4, false)">
    <IMG src="pch4.gif" alt="использование графики" border="0"
        width="103" height="35"></A>
</nobr> <br>
<nobr><IMG src="but0.gif" width="31" height="31" name="p5">
    <A href="tch5.htm" onmouseover="IMG(p5, true)"
        onmouseout="IMG(p5, false)">
    <IMG src="pch5.gif" alt="создание фреймовой структуры"
        border="0" width="103" height="35"></A>
</nobr> <br>
</BODY>
</HTML>

```

6 Горизонтальное графическое меню со стрелкой

Напишем сценарий, реализующий горизонтальное графическое меню. При наведении курсора мыши на пункт меню сверху от выделенного пункта появляется стрелка, как изображено на рис. 5.

При создании горизонтального графического меню используем таблицу, состоящую из двух строк, по пять ячеек в каждой. Пункты меню в виде графических изображений хранятся в ячейках второй строки. Если выбран некоторый пункт меню, то над ним появляется стрелка. Назначение функции `img()` такое же, как и в предыдущем примере. Обратите внимание на вызов указанной функции: вместо логического значения можно использовать 0 или 1.

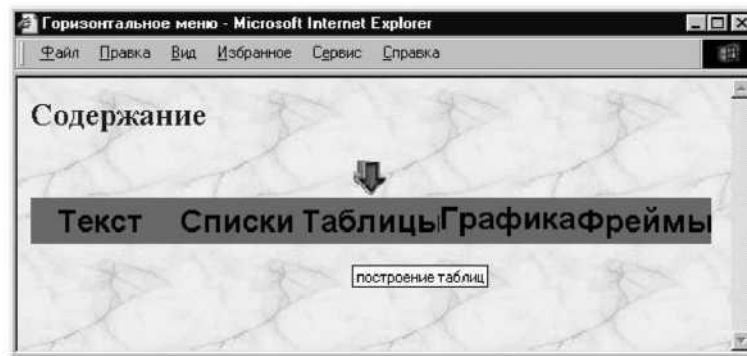


Рис. 5 – Горизонтальное графическое меню со стрелкой

HTML-код представлен в листинге 6.

Листинг 6 – Горизонтальное меню со стрелкой

```
<HTML>
<HEAD>
  <TITLE>Горизонтальное меню</TITLE>
  <script language="JavaScript">
    function IMG(pict, action)
    { if (action)
      {pict.src="rbut1.gif"}
      else
      {pict.src="rbut0.gif"}
    }
  </script>
</HEAD>
<BODY background="fon1.jpg">
  <H2><FONT color="#0000FF">Содержание</FONT></H2>
  <TABLE cellpadding=0 cellspacing=0>
    <TR>
      <TD align=center><IMG src="rbut0.gif"
        width="31" height="31" name="p1"></TD>
      <TD align=center><IMG src="rbut0.gif"
        width="31" height="31" name="p2"></TD>
```

```

        <TD align=center><IMG src="rbut0.gif"
            width="31" height="31" name="p3"></TD>
        <TD align=center><IMG src="rbut0.gif"
            width="31" height="31" name="p4"></TD>

        <TD align=center><IMG src="rbut0.gif"
            width="31" height="31" name="p5"></TD>
    </TR>
    <TR>
        <TD><A href="tch1.htm" target="Main" onmouseover="IMG(p1, 1)"
            onmouseout="IMG(p1, 0)">
            <IMG src="pch1.gif" alt="форматирование текста"
                border="0" width="103" height="35"></A></TD>
        <TD><A href="tch2.htm" target="Main" onmouseover="IMG(p2, 1)"
            onmouseout="IMG(p2, 0)">
            <IMG src="pch2.gif" alt="создание списков"
                border="0" width="103" height="35"></A></TD>
        <TD><A href="tch3.htm" target="Main" onmouseover="IMG(p3, 1)"
            onmouseout="IMG(p3, 0)">
            <IMG src="pch3.gif" alt="построение таблиц"
                border="0" width="103" height="35"></A></TD>
        <TD><A href="tch4.htm" target="Main" onmouseover="IMG(p4, 1)"
            onmouseout="IMG(p4, 0)">
            <IMG src="pch4.gif" alt="использование графики"
                border="0" width="103" height="35"></A></TD>
        <TD><A href="tch5.htm" target="Main" onmouseover="IMG(p5, 1)"
            onmouseout="IMG(p5, 0)">
            <IMG src="pch5.gif" alt="создание фреймовой структуры"
                border="0" width="103" height="35"></A></TD>
    </TR>
</TABLE>
</BODY>
</HTML>

```

Задания к лабораторной работе

Разработать HTML страницу, содержащую меню с указателем. Количество пунктов 5. При выборе каждого из пунктов на странице должно появляться соответствующее изображение или группа изображений.

Вариант 1

Ориентация меню	Вертикальное
Тематика меню	Художники
Тип указателя	кисть
Количество изображений	1
Выполняемое действие	приближение

Вариант 2

Ориентация меню	Вертикальное
Тематика меню	Писатели
Тип указателя	Ручка
Количество изображений	2
Выполняемое действие	циклическая смена

Вариант 3

Ориентация меню	Горизонтальное
Тематика меню	Автомобили
Тип указателя	колесо
Количество изображений	3
Выполняемое действие	циклическая смена

Вариант 4

Ориентация меню	Вертикальное
Тематика меню	Породы котов
Тип указателя	след лапки
Количество изображений	1
Выполняемое действие	вертикальное перемещение

Вариант 5

Ориентация меню	Горизонтальное
Тематика меню	Культовые книги
Тип указателя	буква
Количество изображений	1
Выполняемое действие	Горизонтальное перемещение

Вариант 6

Ориентация меню	Вертикальное
Тематика меню	Культовые фильмы
Тип указателя	ведерко попкорна
Количество изображений	1
Выполняемое действие	появление/исчезание

Вариант 7

Ориентация меню	Горизонтальное
Тематика меню	Белорусские города
Тип указателя	Василек
Количество изображений	1
Выполняемое действие	приближение

Вариант 8

Ориентация меню	Вертикальное
Тематика меню	Праздники
Тип указателя	стрелка
Количество изображений	1
Выполняемое действие	удаление

Вариант 9

Ориентация меню	Горизонтальное
Тематика меню	Страны мира
Тип указателя	стрелка
Количество изображений	1
Выполняемое действие	удаление

Вариант 10

Ориентация меню	Вертикальное
Тематика меню	Известные люди
Тип указателя	фигурка человек
Количество изображений	3
Выполняемое действие	циклическая смена