Министерство образования и науки РФ

ФГБОУ ВПО «Тамбовский государственный технический университет»

Кафедра «САПР»

Лабораторная работа №3

по дисциплине «Интернет технологии»

«JavaScript сценарий»

Выполнил: студент гр. Д. Ю.

Проверил: А. А.

Тамбов,

**Цель работы:** приобретение навыков работы со скриптовым языком JavaScript, написание простейших функций.

**Задание:**

Вариант 8. Разработать скрипты, реализующие управление стилями блоков.

На странице имеются несколько текстовых блоков, с помощью элементов управления задаются стили (не менее 10). Применение стиля к соответствующему блоку осуществляется кликом на нем, повторный клик отменяет заданный стиль и возвращает исходный вид.

Решение:

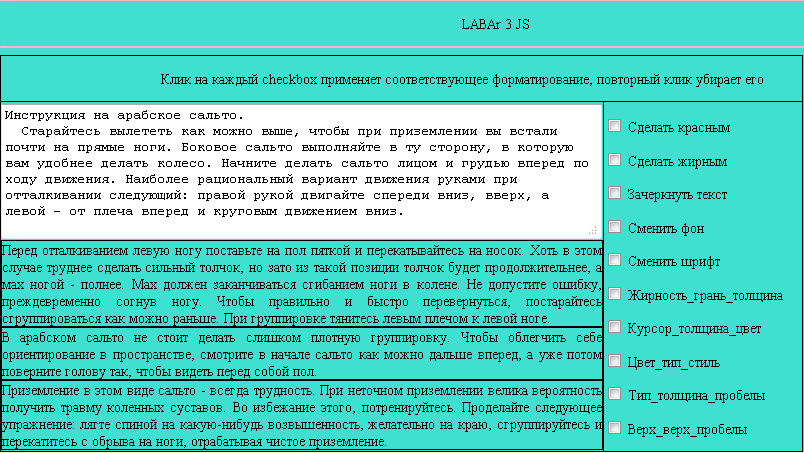
Для удобства создаем таблицу, с лево набор div-блоков, с право компоненты типа input checkbox с описанием стилей для каждого (Рисунок 1.).

Рисунок Таблица

Подразумевается, что пользователь хочет сменить стиль текста, он отмечает требующиеся ему один либо несколько вариантов из правой колонки таблицы и делает щелчок по одному\нескольким блокам текстов.

Код html сведен в таблицу 1.

Таблица 1. HTML код таблицы

|  |
| --- |
| <div style="background-color: Turquoise">  <hr color='Lightpink'><p align=center>LABAr 3 JS</p><hr color='Lightpink'>    <table border=1 cellspacing=0 cellpadding=0 style='border-collapse:collapse;border:none;mso-border-alt:solid black .5pt;'>  <tr>  <td width=638 colspan=2>  <p align="center">Клик на каждый checkbox применяет соответствующее форматирование, повторный клик убирает его</p>  </td>  </tr>  <tr >    <td width=319 >  <textarea name="10" id="10" cols="72" rows="7" onclick="kodimg(id)">Инструкция на арабское сальто.  Старайтесь вылететь как можно выше, чтобы при приземлении вы встали почти на прямые ноги. Боковое сальто выполняйте в ту сторону, в которую вам удобнее делать колесо. Начните делать сальто лицом и грудью вперед по ходу движения. Наиболее рациональный вариант движения руками при отталкивании следующий: правой рукой двигайте спереди вниз, вверх, а левой - от плеча вперед и круговым движением вниз.  </textarea>    <div id="11" onclick="kodimg(id)" class="just">  Перед отталкиванием левую ногу поставьте на пол пяткой и перекатывайтесь на носок. Хоть в этом случае труднее сделать сильный толчок, но зато из такой позиции толчок будет продолжительнее, а мах ногой - полнее. Мах должен заканчиваться сгибанием ноги в колене. Не допустите ошибку, преждевременно согнув ногу.  Чтобы правильно и быстро перевернуться, постарайтесь сгруппироваться как можно раньше. При группировке тянитесь левым плечом к левой ноге.  </div>    <div id="22" onclick="kodimg(id)" class="just">  В арабском сальто не стоит делать слишком плотную группировку. Чтобы облегчить себе ориентирование в пространстве, смотрите в начале сальто как можно дальше вперед, а уже потом поверните голову так, чтобы видеть перед собой пол.  </div>    <div id="33" onclick="kodimg(id)" class="just">  Приземление в этом виде сальто - всегда трудность. При неточном приземлении велика вероятность получить травму коленных суставов. Во избежание этого, потренируйтесь. Проделайте следующее упражнение: лягте спиной на какую-нибудь возвышенность, желательно на краю, сгруппируйтесь и перекатитесь с обрыва на ноги, отрабатывая чистое приземление.  </div>  </td>    <td width=319 valign="top">  <form name="GetStyle" onclick="SwitchCheked()">  <p><input type="checkbox" id="red" name="a" value="Сделать красным"> <label for="red">Сделать красным</label></p>  <p><input type="checkbox" id="bold" name="b" value="Сделать жирным"> <label for="bold">Сделать жирным</label></p>  <p><input type="checkbox" id="crossed" name="c" value="Зачеркнуть текст"> <label for="crossed">Зачеркнуть текст</label></p>  <p><input type="checkbox" id="background" name="d" value="Сменить фон"> <label for="background">Сменить фон</label></p>  <p><input type="checkbox" id="font" name="e" value="Сменить шрифт"> <label for="font">Сменить шрифт</label></p>  <p><input type="checkbox" id="weight" name="f" value="Шрифт"> <label for="weight">Жирность\_грань\_толщина</label></p>  <p><input type="checkbox" id="cursor" name="g" value="Меняем стрелку"> <label for="cursor">Курсор\_толщина\_цвет</label></p>  <p><input type="checkbox" id="vseVodin" name="h" value="Меняем xny"> <label for="vseVodin">Цвет\_тип\_стиль</label></p>  <p><input type="checkbox" id="nocoment" name="j" value="Меняем вариант"> <label for="nocoment">Тип\_толщина\_пробелы</label></p>  <p><input type="checkbox" id="overflow" name="k" value="Меняем потолок"> <label for="overflow">Верх\_верх\_пробелы</label></p>  </form>  </td>  </tr>  </table>  </div> |

И так, что тут есть интересного? Все довольно стандартно, все блоки с текстом заключены в теги <div> и каждому выбран уникальный идентификатор id. Также можно заметить тег textarea, тоже блок с текстом. Главное навесить id, а какой использован тег – не столь важно. К параметру class прицеплен стиль «по умолчанию». Для перехвата события «щелчок мыши» используем обработчик onclick и скриптовую функцию. Весь javascript код вынесен в отдельный файл с расширением js и подключается в самом начале между тегами head:

<script src="switchBar.js" language="javaScript" type="text/javaScript" />

Содержимое этого файла сведено в таблицу 2.

Таблица . Файл switchBar.js

|  |
| --- |
| <!--сформировать строку стиль из чекбоксов и ее применить при клике на див-->  <!--инпуты поместить в контейнер с onclick. Опрашивать все отмеченые чекбоксы и формировать строку стилей-->  var text = document.querySelector("textarea"),  globalStyle = "", flag="0";    function SwitchCheked()  {  globalStyle = "";  var f=document.forms['GetStyle'];  var mas=f.getElementsByTagName('input');  for(i=0; i<mas.length; i++)  {  if(mas[i].checked)  switch(mas[i].name)  {  case "a": globalStyle += " apply-red"; break  case "b": globalStyle += " apply-bold"; break  case "c": globalStyle += " apply-crossed"; break  case "d": globalStyle += " apply-background"; break  case "e": globalStyle += " apply-font"; break  case "f": globalStyle += " apply-weight"; break  case "g": globalStyle += " apply-cursor"; break  case "h": globalStyle += " apply-vseVodin"; break  case "j": globalStyle += " apply-nocoment"; break  case "k": globalStyle += " apply-overflow"; break  default:  }  }  }    function kodimg(id) {    switch(id)  {  case "10":  if(flag==id){ text.className = " just"; flag="0";}  else  {  flag="10";  text = document.getElementById(id);  text.className = globalStyle;  } break  case "11":  if(flag==id){ text.className = " just"; flag="0";}  else  {  flag="11";  text = document.getElementById(id);  text.className = globalStyle;  } break  case "22":  if(flag==id){ text.className = " just"; flag="0";}  else  {  flag="22";  text = document.getElementById(id);  text.className = globalStyle;  } break  case "33":  if(flag==id){ text.className = " just"; flag="0";}  else  {  flag="33";  text = document.getElementById(id);  text.className = globalStyle;  } break  default:  }    } |

Итого, две функции, несколько глобальных переменных.

Функция SwitchCheked() формирует конечный набор стилей. Тут следует уточнить, что теги input заключены в объединяющий тег form, которому задано имя name="GetStyle" и обработчик onclick="SwitchCheked()".

Строкой var f=document.forms['GetStyle'] получаем доступ к форме с конкретным именем. Далее уходим ниже уровнем в иерархии – строка

var mas=f.getElementsByTagName('input') даст доступ ко всем элементам input конкретной формы. При помощи цикла опрашиваем элементы на состояние «вкл.» и если такие есть, то уточняем которые при помощи switch-case. В заключении идет операция конкатенации globalStyle += " apply-font", так формируем стиль. Функция с таким великолепным именем как kodimg(id) крайне не интересна, но эффективна, так как в ней заключена логика «применить\убрать стиль». Идентификатор укажет на блок с текстом, условие ориентируется на флаг, если элемент был активен, то флаг совпадет с id и стиль сбросится на умолчание, если к элементу только предстоит применить стиль, то работает ветка else, флаг запоминает элемент, сформированный стиль применяется к блоку с идентификатором. Если возникает ситуация, когда был выделен элемент например 33, а после – элемент 11, то флаг будет сброшен и тут две ситуации по ветке else, либо будет применен стиль из активных checkbox-ов, либо если нет активных, примениться стиль «пусто», видимо происходит сброс всех стилей.



Кстати сам файл со стилями (Таблица 3) подключается там же, где и скрипт файл строкой <link href="style.css" rel="stylesheet" />

Таблица . Файл стилей style.css

|  |
| --- |
| .just { text-align: justify; width: 600px; border: solid 1px black;}  .apply-red { color: red; border-style: dashed;}  .apply-bold { font-weight: bold; border-style: dotted;}  .apply-crossed { text-decoration: line-through; border-style: solid;}  .apply-background { background-color: silver; border-style: double;}  .apply-font { font-family: Times New Roman; border-style: groove; border-color: green}  .apply-weight { font-weight: thick; border-left-color: yellow; line-height: 50%;}  .apply-cursor { cursor: help; font-size: 200%; color: Cyan;}  .apply-vseVodin { background-color: Firebrick; font-family:Tahoma, sans-serif; font-style:italic;}  .apply-nocoment { font-variant:small-caps; font-weight:bold; letter-spacing:2px;}  .apply-overflow { word-spacing:7px; text-decoration:underline; vertical-align:super;} |

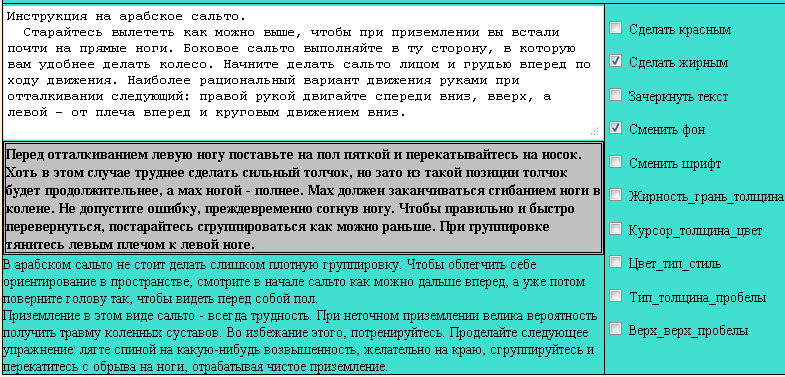


Рисунок . Два пункта на один блок

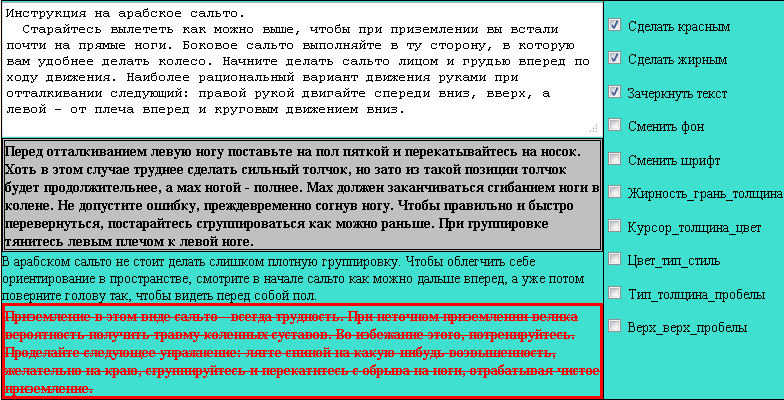
Результаты приведены на рисунках 2, 3, 4, 5.

Рисунок . Три стиля на один блок

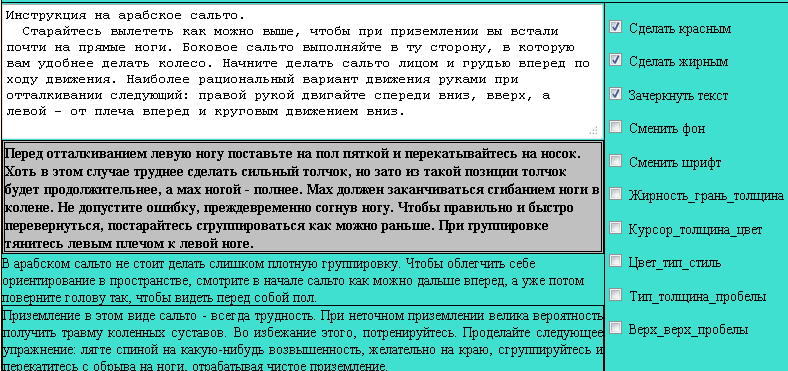
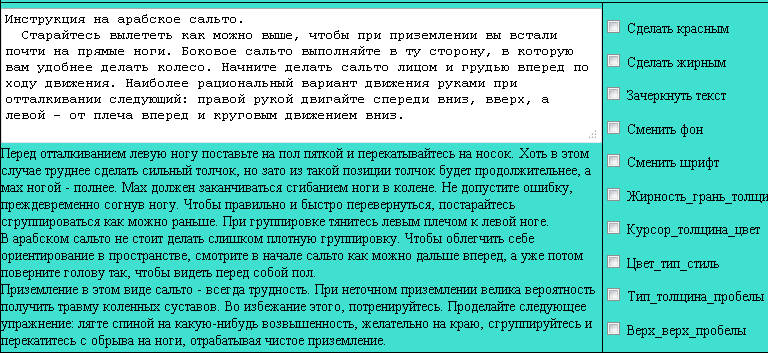


Рисунок 5. Убраны все стили - клик по блокам даст «пустой» стиль, браузерный – у стиля по умолчанию установлено обрамление – рамка, которой тут нет.

Рисунок . Повторный клик сбросит стиль