Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Филиал

«Минский радиотехнический колледж»

Учебная дисциплина «Программные средства создания Internet-приложений»

**Инструкция**

по выполнению лабораторной работы

«Использование слоев и изображений-карт на HTML-странице»

Минск

2017

**Лабораторная работа № 3**

**Тема работы: «Использование слоев и изображений-карт на HTML-странице»**

**1. Цель работы**

Формирование практических умений использования слоев и изображений-карт.

**2. Задание**

Создать HTML-станицу содержащую изображение-карту и слои.

**3. Оснащение работы**

ПК, текстовый редактор, браузер.

**4. Основные теоретические сведения**

Слои

Для создания слоев может использоваться тег <LAYER>. Однако данный тег поддерживается только браузером Netscape Navigator. Более универсальным способом создания слоев является сочетание использования тега <DIV> и каскадных таблиц стилей (CSS).

Шаблон HTML-кода слоя выглядит следующим образом:

<DIV STYLE=”Свойства слоя”> Содержимое слоя</DIV>

В контейнере <DIV> расположены HTML-теги, определяющие элементы, из которых состоит слой.

Свойства слоя записываются следующим образом:

свойство: значение

Можно определить следующие свойства слоя, представленные в таблице 3.1.

Таблица 3.1 – Атрибуты тега div

|  |  |
| --- | --- |
| Свойство | Назначение |
| position: ***absolute| static| relative*** | Точка начала отсчета координат |
| top: ***число*** | Y-координата верхнего левого угла слоя |
| left: ***число*** | X-координата верхнего левого угла слоя |
| width: ***число*** | Ширина слоя |
| height: ***число*** | Высота слоя |
| color: ***цвет*** | Цвет текста |
| background: ***цвет*** | Цвет фона слоя |
| background-image: ***url(путь)*** | Фоновое изображение слоя |
| visibility: ***visible| hidden*** | Первоначальная видимость слоя |
| z-index: ***число*** | Порядок отображения слоев |

Пример создания слоев приведен ниже, результат – на рисунке 3.1.

<DIV style="position:absolute; top:350; left:10; width:180; height:50; background:yellow; color:blue; z-index:1">

<H2 align="center">Планета</H2>

</DIV>

<DIV style="position:absolute; top:0; left:0; z-index:0">

<IMG src="Planet.jpg" width="400" height="400" border="0">

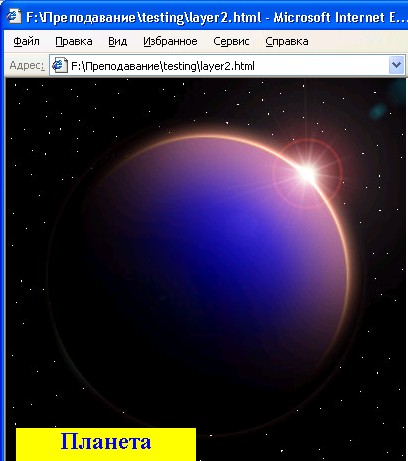
</DIV>

Рисунок 3.1 – Слои на HTML-странице

Как и при использовании других блочных элементов, содержимое тега <div> всегда начинается с новой строки. После него также добавляется перенос строки:

- **align** задает выравнивание содержимого тега <div>;

- [**title**](http://htmlbook.ru/html/div/title) добавляет всплывающую подсказку к содержимому.

Абсолютное позиционирование - Самый простой и менее гибкий  способ создания наложения.

Абсолютное позиционирование позволяет накладывать слои в любом порядке друг на друга. Но при этом требуется знать точные координаты каждого слоя относительно одного из углов окна браузера, что не всегда возможно.

При использовании наложения требуется присвоить свойству **position** значение **absolute**. Само положение слоя регулируется свойствами **left**, **top**, **right** и **bottom**,которые задают координаты соответственно от левого, верхнего, правого и нижнего края.

Более интересный подход к созданию наложения — использование относительного позиционирования. В этом случае слои можно размещать по центру окна браузера или располагать их в любом месте веб-страницы, не задумываясь уже над значением координат слоев.

Относительное позиционирование. Чтобы наложить один слой на другой и не привязываться жестко к координатной сетке, можно попробовать следующий подход. Для первого слоя, который будет располагаться на заднем плане, указываем абсолютное позиционирование, присваивая свойству position значение absolute. Второй слой, расположенный поверх первого, должен иметь относительное позиционирование, что достигается с помощью значения relative у свойства position. Положение верхнего слоя определяется от левого верхнего угла нижнего слоя заданием left и top (на рисунке задание положения верхнего слоя).

Пример представлен на рисунке 3.2.

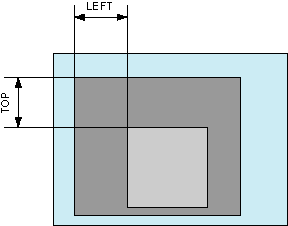


Рисунок 3.2 – Относительное позиционирование

Пример:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset="windows-1251">

<title>Позиционирование</title>

<style type="text/css">

#leftcol {

position: absolute; /\* Абсолютное позиционирование \*/

width: 550px; /\* Ширина левой колонки \*/

background: #e0e0e0; /\* Цвет фона содержимого \*/

padding: 10px /\* Поля вокруг текста \*/

}

#rightcol { /\* Этот слой накладывается поверх \*/

position: relative; /\* Относительное позиционирование \*/

left: 500px; /\* Положение от левого края \*/

top: 20px; /\* Положение от верхнего края \*/

width: 200px; /\* Ширина правой колонки \*/

background: #800000; /\* Цвет фона \*/

color: #fff; /\* Цвет текста \*/

}

</style>

</head>

<body>

<div id="leftcol">Левая колонка</div>

<div id="rightcol">Правая колонка</div>

</body>

</html>

Результат представлен на рисунке 3.3.

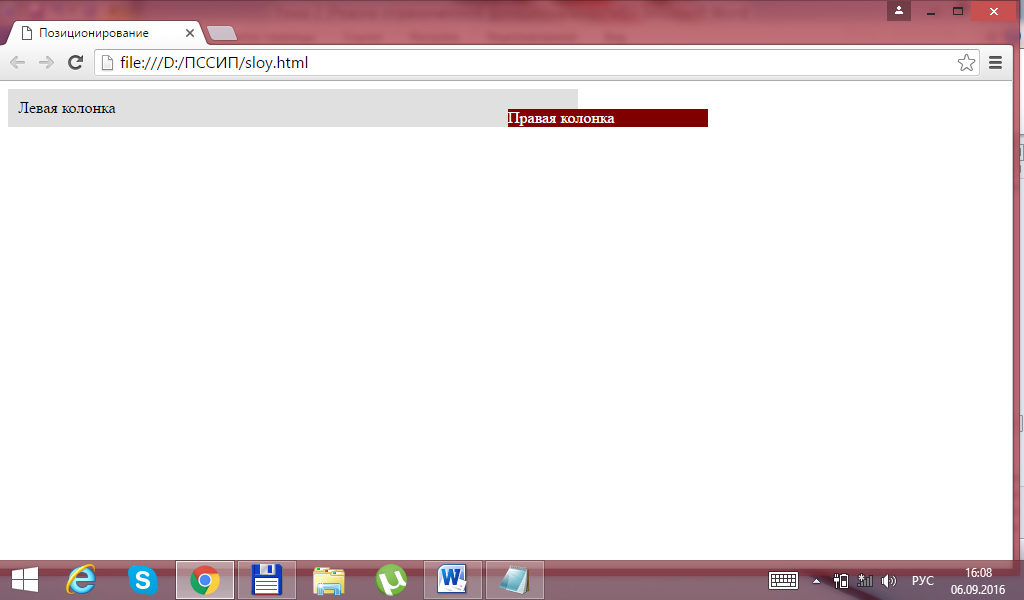


Рисунок 3.3 – Результат положения относительного позиционирования

<!DOCTYPE HTML>

<html>

<head>

<meta charset=" windows-1251 ">

<title>Тег DIV</title>

<style type="text/css">

.block1 {

width: 200px;

background: #ccc;

padding: 5px;

padding-right: 20px;

border: solid 1px black;

float: left;

}

.block2 {

width: 200px;

background: #fc0;

padding: 5px;

border: solid 1px black;

float: left;

position: relative;

top: 40px;

left: -70px;

}

</style>

</head>

<body>

<div class="block1">Lorem ipsum dolor sit amet, consectetuer

adipiscing elit, sed diem nonummy nibh euismod tincidunt ut lacreet

dolore magna aliguam erat volutpat.</div>

<div class="block2">Ut wisis enim ad minim veniam, quis nostrud

exerci tution ullamcorper suscipit lobortis nisl ut aliquip ex

ea commodo consequat.</div>

</body>

</html>

Результат кода представлен на рисунке 3.4.

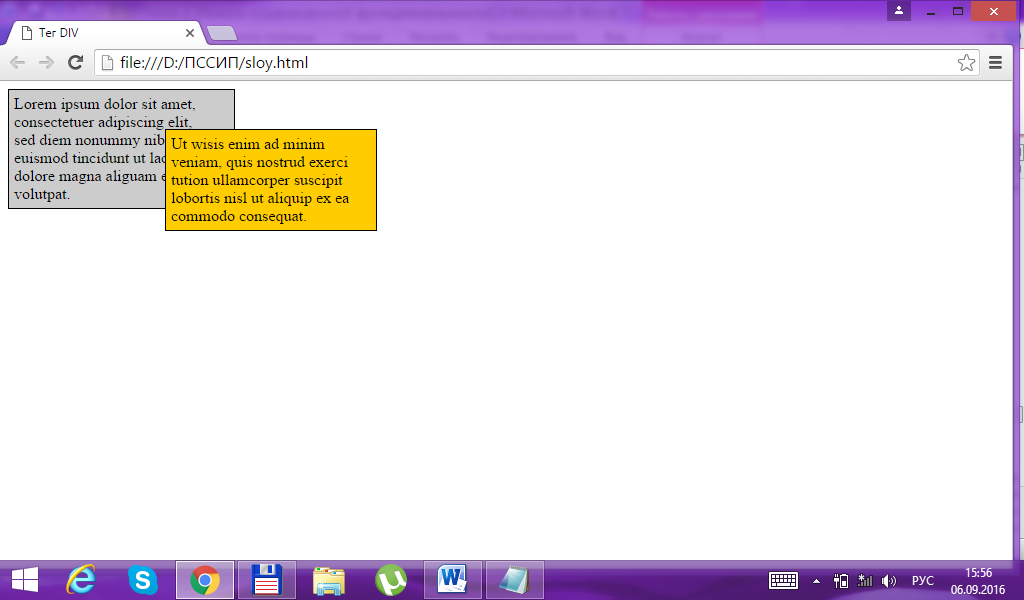


Рисунок 3.4 – Пример реализации

Существует и другой способ наложения слоев, который связан с относительным позиционированием и использует все разнообразие средств размещения разных слоев.

Теоретически, можно разместить слои по вертикали друг под другом и поднять нижний слой вверх с помощью свойства top, задавая ему отрицательное значение или с помощью bottom. На практике добиться подобного довольно сложно, ведь определить высоту слоя простыми средствами, а, следовательно, и величину, на которую следует смещать слой, не представляется возможным, поскольку она зависит от размера шрифта, содержимого слоя и многих других параметров. Проще отсчет координат вести от верхнего угла одного из слоев. Для этого следует разместить слои рядом по вертикали, а потом уже смещать один слой относительно другого.

Реализуется это следующим способом. У каждого слоя необходимо указать конструкцию float: left, которая позволяет один слой пристыковать к другому справа. Добавлять float следует для каждого слоя, иначе в некоторых браузерах появится промежуток между слоями.

Теперь слои располагаются рядом, и прежде, чем указывать координаты, задаем относительное позиционирование значением relative у свойства position. Положение верхнего слоя управляется значением left и top. Но поскольку отсчет координат в данном случае ведется от левого верхнего угла второго слоя, по горизонтали нужно указывать отрицательное значение (рисунок задание положения верхнего слоя). Впрочем, можно использовать также свойство bottom. Пример реализации свойства position представлен на рисунке 3.5.

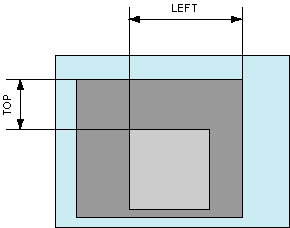


Рисунок 3.5 – Реализация свойства position

Пример:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset= windows-1251">

<title>Позиционирование</title>

<style type="text/css">

#leftcol {

position: relative; /\* Относительное позиционирование \*/

float: left; /\* Совмещение колонок по горизонтали \*/

width: 400px; /\* Ширина слоя \*/

background: #800000; /\* Цвет фона \*/

color: white; /\* Цвет текста \*/

}

#rightcol {

position: relative; /\* Относительное позиционирование \*/

float: left; /\* Совмещение колонок по горизонтали \*/

left: -50px; /\* Сдвиг слоя влево \*/

top: 50px; /\* Смещение слоя вниз \*/

width: 120px; /\* Ширина слоя \*/

background: #e0e0e0; /\* Цвет фона \*/

border: solid 1px black; /\* Параметры рамки \*/

}

#leftcol P {

padding: 10px; /\* Поля вокруг текста \*/

padding-right: 50px; /\* Значение поля справа \*/

margin: 0; /\* Обнуляем значения отступов \*/

}

#rightcol P {

padding: 10px; /\* Поля вокруг текста \*/

padding-top: 0 /\* Значение поля сверху \*/

}

</style>

</head>

<body>

<div id="leftcol">

<p>Левая колонка </p>

</div>

<div id="rightcol">

<p>Правая колонка</p>

</div>

</body>

</html>

Пример реализации представленного выше кода представлен на рисунке 3.6.

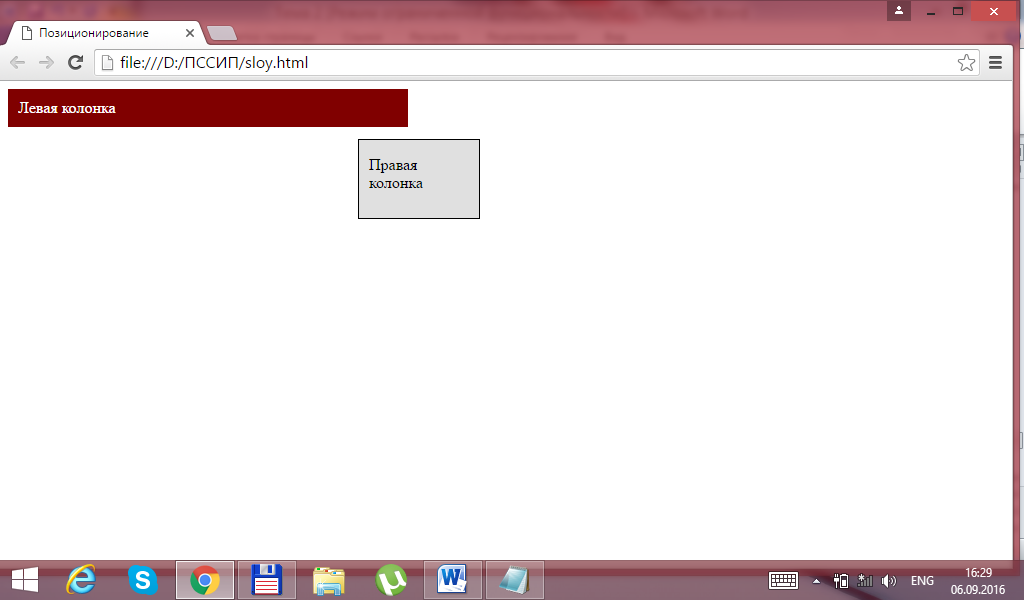


Рисунок 3.6 – Пример реализации кода

Изображения-карты.Различают два вида изображений-карт (карт ссылок):

- обрабатываемые сервером;

- обрабатываемые клиентом.

Для определения изображения-карты, обрабатываемой клиентом, используются теги <MAP> и <AREA> (рисунок 3.7).

<MAP NAME=MAP1>

СОЗДАНИЕ КАРТЫ ССЫЛОК

</MAP>

<AREA …> СОЗДАНИЕ ОБЛАСТИ

<AREA …> СОЗДАНИЕ ОБЛАСТИ

<AREA …> СОЗДАНИЕ ОБЛАСТИ

Рисунок 3.7 – Структура карты ссылок

Атрибуты тега <AREA> [8] приведены в таблице 3.2.

Таблица 3.2 – Атрибуты тега area

|  |  |
| --- | --- |
| Атрибут | Назначение |
| SHAPE | Форма области (rect, circle, poly) |
| COORDS | Координаты области на карте |
| HREF | URL, на который ссылается область |
| NOHREF | Определяет область как неактивную (является атрибутом-флагом) |
| ALT | Альтернативный текст-подсказка для данной области |

При добавлении изображения, являющегося картой ссылок, используется атрибут USEMAP, значением которого является имя карты, указанное в теге <MAP>. Перед именем карты ставится символ #.

Например:

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset= windows-1251">

<title>Карта-изображение</title>

</head>

<body>

<p><img src="images/map.gif" width="411" height="46" **usemap="#map"** alt="Навигация"></p>

<p><map name="map"> <area shape="poly" alt="Закладка 2"

 coords="210,27, 203,9, 202,6, 197,2, 192,1, 120,1, 115,2, 110,6, 112,9, 119,27, 119,32, 211,32, 210,27"

 href="2.html">

<area shape="poly" alt="Закладка 3"

 coords="302,27, 295,9, 293,6, 289,2, 283,1, 212,1, 206,2, 202,6, 203,9, 210,27, 211,32, 284,32, 303,32, 302,27" href="3.html">

<area shape="poly" alt="Закладка 4"

 coords="302,27, 303,32, 394,32, 393,27, 386,9, 382,3, 375,1, 303,1, 298,2, 293,6, 295,9, 302,27"

 href="4.html">

</map></p>

</body>

</html>

Пример определения и использования карты ссылок приведен ниже, результат – на рисунке 3.8.

<MAP NAME="rectangles">

<AREA shape="rect" coords="95,55,295,185" alt="orange" href="2.html">

<AREA shape="rect" coords="0,0,175,120" alt="yellow" href="1.html">

</MAP>

<IMG src="Kard.gif" border="0" USEMAP="#rectangles">

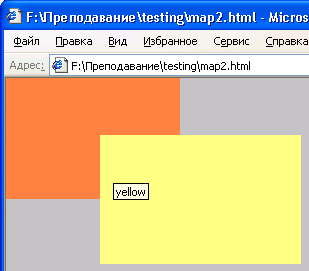


Рисунок 3.8 – Созданная карта ссылок

Атрибут **nonhref** используется для того, чтобы указать, что область не является активной. Например, его можно использовать, чтобы сделать активной область в виде кольца.

**<img src="fig2.gif" border="0" usemap="#Map2">**

**<map name="Map2">**

**<area shape="circle" coords="133,116,59" alt="Нет ссылки (nonhref)" nonhref>**

**<area shape="circle" coords="133,117,89" href="fromimagemap1.htm" alt="На Документ 1">**

**</map>**

Карты-изображения и их исходные коды. Прямоугольник:

<img src="Прямоугольник.gif" usemap="#Прямоугольник" alt="">

<map id="Прямоугольник" name="Прямоугольник">

<area shape="rect" coords="63,18,172,168" href="A:\HTML\Навигационные карты.пример.html"title="Прямоугольник" target="blank" >

<area shape="default" nohref="nohref" alt="" >

</map>

Многоугольник, представленный на рисунке 3.9.

<img src="Многоугольник.gif" usemap="#Многоугольник" alt="">

<map id="Многоугольник" name="Многоугольник">

<area shape="poly" coords="65,153,87,32,120,25,166,48,119,103,164,151,110,174" href="Навигационные%20карты.пример.html" title="Многоугольник" target="blank" >

<area shape="default" nohref="nohref" alt="" >

</map>

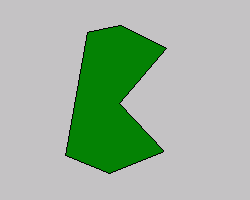


Рисунок 3.9 –Многоугольник

Круг.

<img src="Круг.gif" usemap="#Круг" alt="">

<map name=” Круг”>

<area shape="circle" coords="126,99,73" href="Навигационные%20карты. пример.html" title="круг" target="blank" >

<area shape="default" nohref="nohref" alt="" >

</map>

Кольцо, представленное на рисунке 3.10.

<img src="Кольцо.gif" usemap="#Кольцо" alt="">

<map id="Кольцо" name="Кольцо">

<area shape="circle" coords="128,96,66" nohref="nohref" title="кольцо">

<area shape="circle" coords="128,96,70" href="Навигационные%20карты.

пример.html" title="" target="blank">

<area shape="default" nohref="nohref" alt="">

</map>

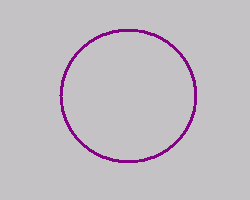


Рисунок 3.10 – Кольцо

Карты-изображения позволяют создавать ссылки на разные области одного изображения.

**5. Порядок выполнения работы**

1. Создайте документ html содержащий карту-изображение улицы на которой вы проживаете.
2. Создайте карту-ссылку на дом, в котором вы проживаете и на три прилегающие улицы. Сделать эти области кликабельными. При нажатии на область в новой вкладке должна открываться информация об улице соответственно или номере дома.
3. Создайте еще один документ html содержащем 2 изображения и надпись.
4. Используя правила наложения слоев отобразите данные.

**6. Форма отчета о работе**

*Лабораторная работа № \_\_\_*

*Номер учебной группы \_\_\_\_\_\_\_\_\_\_\_\_*

*Фамилия, инициалы учащегося \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Дата выполнения работы \_\_\_\_\_\_\_\_\_\_\_\_\_*

*Тема работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Цель работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Оснащение работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Результат выполнения работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

**7. Контрольные вопросы и задания**

1. Перечислите способы создания слоев. В чем их отличие?

2. Как определяется порядок наложения слоев?

3. Какие значения может принимать свойство position? В чем их отличия?

4. Дайте определение понятия «изображение-карта»

5. Как создаются изображения-карты?

6. Перечислите атрибуты тега <AREA>.

7. Какой атрибут указывает, что изображение является картой ссылок?

**8. Рекомендуемая литература**

**Прохоренок, Н.А**. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера / Н.А.Прохоренок. – СПб.: БХВ-Петербург, 2010.

**Никсон, Р.** Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон. — СПб.: Питер, 2016.