**Лабораторная работа № 1. Знакомство с HTML**

Цель: изучить структуру HTML документа; принципы создания списков, таблиц, гиперссылок; особенности применения тегов <span> и <div>.

**Краткая теория**

HTML (Hyper Text Markup Language) – язык разметки гипертекста. Для разметки HTML документа используют теги (флаги разметки).

Тег – это определенная последовательность символов, заключенные между знаками < (больше) и > (меньше).

Для того, чтобы создать HTML документ необходимо:

- открыть любой текстовый редактор (например, блокнот встроенный в Windows);

- набрать произвольный текст и разметить его HTML тэгами;

- сохранить файл с расширением .htm или .html.

Теперь если открыть созданный файл с помощью веб-браузера, он будет отображен как веб-страница.

Структура HTML-документа:

<HTML>

<HEAD>

<TITLE>простое название</TITLE>

</HEAD>

<BODY>

Содержание страницы

</BODY>

</HTML>

Параметры документа задаются через тег <BODY>. Параметры – это специальные символьные команды, которые «понимает» браузер. Тег BODY имеет следующие атрибуты:

- bgcolor – цвет фона задан по схеме RGB;

- background – фон – графический файл;

- text – цвет текста;

- link – цвет ссылки;

- alink – цвет активной ссылки;

- vlink – цвет посещенной ссылки.

Шрифт задается тегом FONT, который имеет следующие атрибуты: face – имена шрифтов, разделенные запятыми, size – размер от 1 до 7 (по умолчанию 3), color – цвет шрифта.

В HTML существуют специальные теги для заголовков: от Н1 (самого крупного) до Н6 (самого мелкого).

Списки на Web-страницах бывают маркированные (<UL>– начало текста списка, <LI> – начало каждого элемента в списке) и нумерованные (<OL> – начало текста списка, <LI> – начало каждого элемента в списке). Маркером списка может быть рисунок.

Атрибут TYPE позволяет изменить вид маркера. Значения атрибута:

- circle – ○;

- disk – ●;

- square – ■;

- A – A, B, C;

- а – a, b, c;

- I– I, II, III;

- i – i, ii, iii;

- 1 – 1, 2, 3.

Таблицы в HTML.

Элемент <table> служит контейнером для элементов, определяющих содержимое таблицы. Любая таблица состоит из строк и ячеек, которые задаются с помощью тегов <tr> и <td>. Для отображения границ таблицы используется атрибут border. С помощью тэга <th> можно создать табличный заголовок. Текст элемента th центрируется и выделяется жирным шрифтом. С помощью атрибута colspan можно указать на сколько столбцов должна быть растянута указанная ячейка, а с помощью атрибута rowspan – на сколько строк должна быть растянута указанная ячейка.

Ссылки на Web-страницах.

Ссылки являются основой гипертекстовых документов и позволяют переходить с одной веб-страницы на другую. Реализуются с помощью тега A. Общий синтаксис создания ссылок следующий:

<a href="URL">текст ссылки</a>

Атрибут href определяет URL адрес документа, на который следует перейти (например "cat.html"), а содержимое контейнера <a> является ссылкой.

**Вставка изображений**.

Для добавления в HTML документ изображений используется тэг <img>. Атрибут src тэга <img> должен содержать адрес, по которому находится картинка, которая должна быть отображена. Например,

<img src="mount.jpg" />.

Тег <div>.

Тег <div>является блочным элементом и предназначен для выделения фрагмента документа с целью изменения вида содержимого. Как правило, вид блока управляется с помощью стилей.

Тег <span>.

Тег <span> предназначен для определения строчных элементов документа. В отличие от блочных элементов, таких как <table>, <p> или <div>, с помощью тега <span> можно выделить часть информации внутри других тегов и установить для нее свой стиль. Например, внутри абзаца (тега <p>) можно изменить цвет и размер первой буквы, если добавить начальный и конечный тег <span> и определить для него стиль текста.

**Задания к лабораторной работе № 1**

**Задание 1.** Создать HTML-документ (заголовок Моя первая страница) cо следующим содержимым:

Заголовок в центре страницы (теги h1) «Я, ФИО, изучаю HTML и CSS». Аббревиатуру HTML выделить другим цветом.

Три абзаца текста (можно своя автобиография) по 3–4 строки каждый абзац. Для каждого абзаца свой шрифт, размер, цвет (тег font). Для каждого абзаца придумать заголовок (h3), а также заголовок для всего текста (h2). Внутри каждого абзаца выделить несколько слов элементами span.

Список примерно такого вида с любым содержимым

1. Элемент 1

· Элемент 1.1

· Элемент 1.2

2. Элемент 2

§ Элемент 2.1

§ Элемент 2.2

§ Элемент 2.3

3. Элемент 3

o Элемент 3.1

o Элемент 3.2

Перед списком вставить заголовок, отражающий его содержимое (**h2**).

Таблица, содержащая результаты сессии. Перед таблицей вставить заголовок (**h2**).

|  |  |
| --- | --- |
| ФИО | Иванов Иван Иванович |
| Свое фото | Адрес: ул.Белорусская, д.5, ком.430 |
| 1 семестр 2016/2017 уч. года | Дата | Предмет | Оценка |
| 31.12.2016 | ОАиП |  |  |
| 5.01.2017 | Физика |  |  |
| 11.01.2017 | ОИТ |  |  |
| 16.01.2017 | Математика |  |  |
| 22.01.2017 | АЛОЦВМ |  |  |

**Задание 2.** Создать еще один HTML-документ (заголовок Моя вторая страница) cо следующим содержимым:

Два элемента<div> с произвольным текстом. Перед каждым элементом заголовок (**h3**).

Любой рисунок. Перед рисунком оформить заголовок (**h2**).

Элемент <span>, содержащий авторский знак © и ФИО автора.

**Задание 3.** ФИО в заголовке на первой web-странице и в таблице сделать ссылкой на вторую страницу, а рисунок на второй странице – ссылкой на первую.

**1.2. Лабораторная работа № 2. HTML+CSS**

Цель: изучить основные способы подключения CSS; возможности оформления текста, шрифта, фона на web-странице; научиться использовать селекторы классов и идентификаторов.

**Краткая теория**

Подключение CSS.

Внешнее. Самый мощный и удобный способ определения стилей и правил для сайта. Стили хранятся в отдельном файле, который может быть использован для любых веб-страниц. Для подключения таблицы связанных стилей используется тег <link>в заголовке страницы.

Внутреннее. Стиль определяется в самом документе и обычно располагается в заголовке веб-страницы. По своей гибкости и возможностям этот способ использования стиля уступает предыдущему, но также позволяет размещать все стили в одном месте. В данном случае, прямо в теле документа. Определение стиля задается тегом <style>.

Строковое. Внутренний стиль являются по существу расширением для одиночного тега, используемого на веб-странице. Для определения стиля используется атрибут style, а его значения указываются с помощью языка таблицы стилей.

Текст.

Цвет текста задается с помощью свойства color. Можно задавать разными способами: по шестнадцатеричному значению, по названию, в формате RGB, RGBA, HSL, HSLA.

Горизонтальное выравнивание текста в пределах элемента определяет свойство text-align.

Значения:

- сenter – выравнивание текста по центру, текст помещается по центру горизонтали окна браузера или контейнера, где расположен текстовый блок;

- justify – выравнивание по ширине, что означает одновременное выравнивание по левому и правому краю;

- left – выравнивание текста по левому краю, строки текста выравнивается по левому краю, а правый край располагается «лесенкой»;

- right – выравнивание текста по правому краю, этот способ выравнивания выступает в роли антагониста предыдущему типу;

- auto – не изменяет положение элемента;

- inherit – наследует значение родителя.

Отступ между словами и буквами в тексте:

- word-spacing – устанавливает интервал между словами; если для текста задано выравнивание через text-align со значением justify (выравнивание по ширине), то интервал между словами будет установлен принудительно, но не меньше значения, указанного через word-spacing;

- letter-spacing – определяет интервал между символами в пределах элемента. Значения устанавливаются так же, как и в предыдущем свойстве.

Свойство text-decoration добавляет оформление текста в виде его подчеркивания, перечеркивания, линии над текстом и мигания. Одновременно можно применить более одного стиля, перечисляя значения через пробел.

Значения:

- blink – устанавливает мигающий текст. Такой текст периодически, примерно раз в секунду исчезает, потом вновь появляется на прежнем месте.

- line-through – создает перечеркнутый текст;

- overline – линия проходит над текстом;

- underline – устанавливает подчеркнутый текст;

- none – отменяет все эффекты, в том числе и подчеркивания у ссылок, которое задано по умолчанию.

Шрифт

Свойство font-family устанавливает семейство шрифта, которое будет использоваться для оформления текста содержимого. Список шрифтов может включать одно или несколько названий, разделенных запятой. Если в имени шрифта содержатся пробелы, например, Trebuchet MS, оно должно заключаться в одинарные или двойные кавычки.

Cвойство font-size определяет размер шрифта элемента. Разрешается использовать любые допустимые единицы CSS: em (высота шрифта элемента), ex (высота символа х), пункты (pt), пикселы (px), проценты (%) и др. За 100% берется размер шрифта родительского элемента. Отрицательные значения не допускаются.

Свойство font-style – определяет начертание шрифта (обычное, курсивное или наклонное).

Значения:

- normal – обычное начертание текста;

- italic – курсивное начертание;

- oblique –наклонное начертание,курсив и наклонный шрифт при всей их похожести не одно и то же.

Свойство font-weiqht устанавливает насыщенность шрифта. Значение устанавливается от 100 до 900 с шагом 100. Сверхсветлое начертание, которое может отобразить браузер, имеет значение 100, а сверхжирное – 900. Нормальное начертание шрифта (которое установлено по умолчанию) эквивалентно 400, стандартный полужирный текст – значению 700.

Насыщенность шрифта задается с помощью ключевых слов: bold – полужирное начертание, normal – нормальное начертание. Также допустимо использовать условные единицы от 100 до 900. Значения bolder и lighter изменяют жирность относительно насыщенности родителя, соответственно, в большую и меньшую сторону.

Фон.

Свойство background-attachment устанавливает, будет ли прокручиваться фоновое изображение вместе с содержимым элемента. Изображение может быть зафиксировано и оставаться неподвижным, либо перемещаться совместно с документом.

Свойство background-color определяет цвет фона элемента.

Свойство background-image устанавливает фоновое изображение для элемента. Если одновременно для элемента задан цвет фона, он будет показан, пока фоновая картинка не загрузится полностью. В качестве значения используется путь к графическому файлу, который указывается внутри конструкции url(). Путь к файлу при этом можно писать как в кавычках (двойных или одинарных), так и без них.

Свойство background-position задает начальное положение фонового изображения, установленного с помощью свойства background-image.

У свойства background-position два значения, положение по горизонтали (может быть – left, center, right) и вертикали (может быть – top, center, bottom). Кроме использования ключевых слов положение также можно задавать в процентах, пикселах или других единицах. Если применяются ключевые слова, то порядок их следования не имеет значения, при процентной записи вначале задается положение рисунка по горизонтали, а затем, через пробел, положение по вертикали.

Свойство background-repeat определяет, как будет повторяться фоновое изображение, установленное с помощью свойства background-image. Можно установить повторение рисунка только по горизонтали, по вертикали или в обе стороны.

**Задания к лабораторной работе № 2**

Задания выполняются со **своими** HTML-документами, созданными ранее.

**Задание 1.** Стили для оформления заголовков на web-страницах описать во внешнем файле. В одном правиле задать одинаковый цвет и вид шрифта для заголовков h1, h2, h3 и элементов span (группировка селекторов).

**Задание 2.** Абзацы текста. Заменить тег font в оформлении абзацев текста на классы, используя внутреннее подключение. Для каждого абзаца создать свой стиль, в котором использовать перечисленные выше свойства текста, шрифта. У каждого абзаца должно быть разное выравнивание, отступ между словами и буквами, разный вид и размер шрифта. Продемонстрировать разные способы (три) указания цвета текста и размера шрифта.

Для элемента span первого абзаца задать свой стиль, используя строковое подключение. Продемонстрировать приоритет стилей.

**Задание 3.** Изменить параметры элементов span второго и третьего абзаца (цвет текста, стиль шрифта, оформление). Использовать селекторы идентификаторов.

**Задание 4.** Для первой web-страницы задать цветной фон. Для второй web-страницы задать в качестве фона изображение, которое постоянно будет размещено в правом верхнем углу, используя краткую форму записи.

**1.3. Лабораторная работа № 3. HTML+CSS**

Цель: изучить возможности оформления списков, ссылок, таблиц на web-странице, а также познакомиться с псевдо-классами.

**Краткая теория**

**Списки**(свойство list-style-type).

Нумерованные (значения decimal, lower-latin, lower-roman, upper-latin, upper-roman)

Маркированные (значения none, disc, circle, square)

Использование маркера-картинки (свойство list-style-image)

**Ссылки**

Ссылки могут находиться в четырех различных состояниях и каждое из этих состояний может быть отдельно оформлено с помощью специальных псевдо-классов:

- a:link определяет оформление обычной не посещенной ссылки;

- a:visited определяет оформление посещенной пользователем ссылки;

- а:hover определяет оформление ссылки, на которую наведен курсор мыши;

- a:active определяет оформление ссылки, на которую щелкнули мышкой.

**Таблицы**

Ширина (свойство width, устанавливается в пикселях или %, можно использовать cm и em).

Высота (свойство height, указывается в пикселях, можно использовать cm и em).

Оформление границ (свойство border позволяет одновременно установить толщину, стиль и цвет границы вокруг элемента, border-collapse устанавливает, как отображать границы вокруг ячеек таблицы).

Выравнивание текста в таблице (свойство text-align значения right, left,center, свойство vertical-align значения top, middle, bottom).

Отступ содержимого в ячейке (свойство padding).

**Псевдо-классы**

- first-child позволяет выбрать элемент, который является первым потомком в его родительском элементе;

- first-letter позволяет оформить первую букву указанного элемента;

- first-line позволяет оформить первую строчку указанного элемента;

- before, after позволяет вставлять произвольное содержимое до и после указанных элементов.

**Задания к лабораторной работе № 3**

Задания выполняются со **своими** HTML-документами, созданными ранее.

**Задание 1.** Создать по одному стилю для каждого из уровней списка и применить их. Использовать свойства шрифта, цвета. Создать любой список с использованием маркера-картинки (своя фотография).

**Задание 2.** Использовать псевдо-классы для оформления ссылок.

**Задание 3.** Изучить свойства для выравнивания текста в таблице и для установления отступа в ячейках на готовых примерах (файлы tabl1.html, tabl2.html, tabl3.html). Закрепить знания, выполнив задание в файле zadanie\_tabl.html.

**Задание 4.** Оформить таблицу на своей первой странице при помощи таблицы стилей. Задать шрифты, начертание, цвет, размер содержимого ячеек таблицы, а также выравнивание и отступы. Сохранить страницу в новом файле.

**Задание 5.** Применить на своих страницах псевдо-классы first-child, first-letter, first-line, before, after:

1) изменить цвет текста первого абзаца, используя first-child;

2) первую букву всех абзацев сделать большой и в рамке (пример в файле ps\_klass.html);

3) подчеркнуть первую строку абзацев;

4) перед ФИО добавить текст «студент I курса», после ФИО – «проживающий по адресу …». Пример в файле ps\_klass.html.

**Краткая теория**

Границы

Стиль границ HTML-элемента (свойство border-style значения solid, dashed, dotted, double).

Цвет границы (свойство border-color).

Толщина границы (свойство border-width значения в пикселях или thin, medium, thick).

Задание стилей для отдельных сторон, названия сторон перечислены на рисунке (рис.1.):



Рис.1. Стили и названия сторон блоковой модели

Способ быстрого задания стилей границ.

Пример1*.*Border-style:dashed double solid groove – к верхней границе будет применено dashed, к правой double, к нижней solid, а к левой groove.

Пример2*.*Border-style:dashed double –верхняя и нижняя граница будут оформлены как dashed, а левая и правая граница как double.

Сокращенная форма записи объединяет все свойства оформления границ в одном свойстве border. Порядок следования свойств: border-width, border-style, border-color (можно пропускать неиспользуемые свойства). Пример: border:1px solid.

Отступы (величина в пикселях (px), сантиметрах (cm), процентах (%), em):

- внутренние (свойство padding);

- внешние (свойства margin);

- величина отступа может быть задана отдельно для каждой стороны элемента.

Краткая форма записи: padding: 60px 20px 40px 50px; margin: 30px 50px.

Отображение элементов

Существуют два способа отображения элементов: visibility:hidden и display:none).

Размещение элементов

Местоположение элементов задается с помощью следующих CSS свойств:

- top – величина смещения текущего элемента от верхнего края родительского элемента;

- bottom – от нижнего края родительского элемента;

- left – от левого края родительского элемента;

- right – от правого края родительского элемента.

Описанные выше свойства позиционирования не вступят в силу пока Вы не зададите способ размещения. В CSS существуют 4 различных способа размещения элементов:

- статическое (position:static) действует по умолчанию, элементы всегда отображаются там, где они были объявлены (свойства top, bottom, left и right не работают со статичными элементами);

- фиксированное (position:fixed) элементы не изменяют своего местоположения даже при прокрутке окна браузера (свойства top, bottom, left, right работают);

- относительное (position:relative);

- абсолютное (position:absolute).

Наложение элементов (свойство z-index значение – любое число, может быть отрицательное).

Выравнивание элементов.

По центру с помощью margin. Блоковые элементы могут быть выровнены по центру установкой margin с левой и правой стороны значения auto.

С помощью свойств позиционирования

С помощью свойства float. Элемент, выровненный с помощью float, будет прижат к левой или правой границе родительского элемента (в зависимости от заданного значения) и заставит следующие за ним элементы "обтекать" его с противоположной стороны. Очистить элементы от float можно командой clear:both.

**Задания к лабораторной работе № 4**

**Задание 1**. Для предварительного знакомства c блоковой моделью внимательно посмотрите следующий HTML документ и его отображение браузером.

<html>

<head>

<style type='text/css'>

#wrap {

margin:0px;

padding:20px;

height:160px;

background-color:yellow;

}

.ex1 {

border:10px red solid;

margin-left:50px;

margin-right:10px;

padding:15px;

background-color:green;

color:white;

}

.ex2 {

border:5px brown solid;

margin-top:30px;

margin-left:250px;

margin-right:70px;

padding:15px;

background-color:green;

color:white;

}

</style>

</head> <body>

<div id='wrap'>

<div class='ex1'>Содержимое первого элемента</div>

<div class='ex2'>Содержимое второго элемента</div>

</div>

</body> </html>



На примере изучите внутренние и внешние отступы элементов. Измените цвет элементов, размер элементов, величину отступов и сохраните как свою работу.

**Задание 2.** Оформите элементы согласно их описанию. Стиль границ, цвет и толщину можно придумать свои.

<html>

<head>

<style type='text/css'>

#par1

{

/\* Пишите код здесь \*/

}

#par2

{

/\* Пишите код здесь \*/

}

#par3

{

/\* Пишите код здесь \*/

}

#par4

{

/\* Пишите код здесь \*/

}

#par5

{

/\* Пишите код здесь \*/

}

#par6

{

/\* Пишите код здесь \*/

}

#par7

{

/\* Пишите код здесь \*/

}

</style>

</head>

<body>

<p id='par1'>Я имею сплошную границу коричневого цвета.</p>

<p id='par2'>Я имею сплошную границу красного цвета слева и справа.</p>

<p id='par3'>Я имею сплошную границу розового цвета сверху, и пунктирную границу голубого цвета слева, справа и снизу.</p>

<p id='par4'>Я имею границу оранжевого цвета толщиной 1 пиксель.</p>

<p id='par5'>Я имею границу красного цвета толщиной 2 пикселя сверху и снизу и пунктирную границу серого цвета слева и справа.</p>

<p id='par6'>Я имею двойную сплошную границу #1435AD цвета. </p>

<p id='par7'>Я имею пунктирную границу #FF8100 цвета и толщиной 1 пиксель сверху и сплошную границу #0B6124 цвета толщиной 2 пикселя снизу. </p>

</body>

</html>

**Задание 3.** Узнайте кодовое слово, используя свойство visibility:hidden, затем свойство display:none). Объясните разницу.

<html> <head>

<style type='text/css'>

span

{

font-size:1.6em;

font-family:verdana;

color:green;

}

/\* Пишите код здесь \*/

</style>

</head>

<body>

<b>

<p>Для того, чтобы узнать кодовое слово скройте элементы со следующими id: code, winter, sky, frog, forest, cloud, hide. </p></b><br />

<hr />

<span id='code'>Б</span><span id='mirror'>А</span><span id='winter'>ДЖ</span><span id='grass'>Л</span><span id='sky'>МЖ</span><span id='bright'>Т</span><span id='forest'>Ь</span><span id='sun'>А</span><span id='frog'>ЗП</span><span id='cloud'>ШН</span><span id='true'>Й</span><span id='hide'>РО</span>

<hr />

</body> </html>

**Задание 4.** Поработайте с элементами div на своей второй web-странице. При помощи свойств margin, border, padding задайте для них различный цвет и толщину отступа, границы полей так, чтобы эти составляющие модели блоковых элементов стали видны. Поэкспериментируйте с разными способами их размещения, не менеее 3 вариантов, сохраните в файлах.

На второй web-странице создайте список следующего вида, где элементы списка являются ссылками на страницы с выполненными заданиями:

§ задание 1

§ задание 2

§ задание 3

1) задание 4.1

2) задание 4.2

3) задание 4.3

**Краткая теория**

Фон

Свойство background-size устанавливает размер фоновых изображений (в пикселях или в процентах), например background-size:150px 250px;

В свойстве background-image можно указать несколько фоновых изображений одновременно:- background-image: url (wis.gif), url (mount3.jpg);- background-size: 150px 40px, 100% 100%.

Цвет

- с помощью HSL – оттенок, насыщенность, яркость;

- с помощью RGBA – цвет и прозрачность;

- с помощью HSLA – цвет и прозрачность.

Границы

Свойство border-radius позволяет делать углы элементов сглаженными.

С помощью свойства box-shadow можно добавлять к элементам страницы тени.

С помощью свойства border-image можно вставлять произвольные изображения в качестве границы элемента.

Шрифт

Свойство @font-faceпозволяет подключать шрифты.

@font-face {font-family:opensans;/\* Задаем имя шрифта \*/src:url('opensans.woff')/\* Указываем местонахождение нашего шрифта \*/ }

Текст

Свойство text-shadow добавляет к тексту элементов тени (можно несколько).

Указывается величина смещения тени от текста по горизонтали и вертикали (может быть отрицательной), а также радиус размытия и цвет тени. Примеры:

text-shadow:3px 2px #FFAE00;text-shadow:1px 1px 10px #FFAE00;text-shadow:2px 2px 2px #FFAE00, 2px 2px 15px #1435AD.

Свойство text-overflow указывает, что должно случиться с текстом, вышедшем за пределы границ элемента.

Трансформации

Cвойство **transform** трансформирует элементы. Его значение – функция трансформирования:

- **translate(x,y)** смещает элемент на указанное количество пикселей по осям *X* и *Y* rotate(градусы) поворачивает элемент на указанное количество градусов по часовой стрелке;

- **scale(x,y)** растягивает элемент в ширину или высоту;

- **skew(x,y)** задает перекосы по оси *X* и *Y* под заданным углом.

Градиент

В CSS3 имеются встроенные свойства для создания градиентов. Для поддержки градиентов требуется добавить префиксы: для IE10+ требуется префикс -ms, для Chrome и Safari префикс -webkit, для Opera префикс -o и для Firefox префикс –moz.

- линейный градиент linear-gradient;

- сферический градиент radial-gradient.

Переходы

С помощью свойства **transition** можно создавать эффекты перехода, указав какое CSS свойство будет изменяться и скорость выполнения этих изменений в секундах. Чтобы добавить эффект перехода к нескольким свойствам, надо просто перечислить их названия через запятую.

**Задания к лабораторной работе № 5**

Исходные данные находятся в папке Лаб5.Возможности CSS3.

**Задание 1.** Продемонстрировать на своих web-страницах новые возможности CSS3 для оформления фона web-страницы, текста, границ элементов, цвета. Применить все описанные выше свойства для элементов своих web-страниц.

**Задание 2.** Изучить способы трансформации рисунка на своей второй странице (все 4).

**Задание 3.** Изучить возможности использования градиента на примерах в файлах Gradient1.html, Gradient2.html, Gradient3.html. Применить на своей странице.

**Задание 4.** Научиться применять эффекты перехода на примерах файлов prim1.html, prim2.html, prim3.html. Применить на своей странице

Получившиеся HTML-документы сохранить в новых файлах.

**1.6. Лабораторная работа № 6. Анимация. Создание  
выпадающего меню**

Цель: изучить свойства анимации, научиться создавать выпадающие меню.

Краткая теория

Анимация

Свойства анимации поддерживаются браузерами IE10+, Firefox и Opera. Для браузеров Chrome и Safari перед свойством требуется добавить префикс -webkit.

Для создания анимации в CSS3 используется свойство **@keyframes**. Оно представляет собой контейнер, в который помещаются различные свойства оформления.

Чтоб анимировать элемент, надо добавить ему свойство **animation** и указать в нем имя анимации (1-е значение) и время (2-е значение), в течении которого она будет выполняться. Также можно устанавливать количество повторов анимации (3-е значение).

@keyframes имяАнимации{from {CSS свойства}/\* Оформление элемента перед началом анимации \*/to {CSS свойства}/\* Оформление элемента после завершения анимации \*/}#wrap1 {animation:anim 4s 3;}

С помощью % можно более точно контролировать ход выполнения анимации, например, можно указать, что определенный элемент в начале анимации (0%) имеет один цвет, к середине (50%) окрашивается в другой цвет, а к концу (100%) – в третий цвет.

**Пример1:**

<html>

<style type='text/css'>

@-webkit-keyframes anim {

0% {margin-left:3px;margin-top:3px;background-color:#7F0055;}

30% {margin-left:3px;margin-top:250px;background-color:#7F0055;}

60% {margin-left:500px;margin-top:250px;background-color:black;}

100% {margin-left:3px;margin-top:3px;background-color:#7F0055;}

}

#wrap1 {

border:2px #000 solid;

background-color:#7F0055;

height:100px;

width:100px;

font-size:2em;

animation:anim 6s 3;

-webkit-animation:anim 6s 3;

}

</style>

<body>

<div id="wrap1"></div>

</body>

</html>

**Пример 2:**

<style type="text/css">

#holder {

margin:50px auto;

-webkit-perspective:600px;

-moz-perspective:600px;

text-indent: 0px;

}

.box {

position: relative;

margin: 0 auto;

height: 200px;

width: 200px;

-webkit-animation: spin 20s infinite linear;

-moz-animation: spin 20s infinite linear;

-webkit-transform-style: preserve-3d;

-moz-transform-style: preserve-3d;

}

.box div {

position: absolute;

height: 200px;

width: 200px;

box-shadow: 0px 0px 1px rgba(0, 0, 0, 0.5);

-webkit-animation:rotate 15s infinite linear;

-moz-animation:rotate 15s infinite linear;

}

.box div img {width:200px; height:200px;

border-radius:200px;

-webkit-animation:rot 10s infinite linear;

-moz-animation:rot 10s infinite linear;

padding: 0;

}

.box div.side1 {

background:rgba( 213, 62, 7, 0.8);

-webkit-transform:translateZ(100px);

-moz-transform:translateZ(100px);

}

.box div.side2 {

background:rgba( 127, 255, 212, 0.8);

-webkit-transform:rotateY(90deg) translateZ(100px);

-moz-transform:rotateY(90deg) translateZ(100px);

}

.box div.side3 {

background:rgba( 255, 219, 88, 0.8);

-webkit-transform:rotateY(180deg) translateZ(100px);

-moz-transform:rotateY(180deg) translateZ(100px);

}

.box div.side4 {

background:rgba( 153, 17, 153, 0.8);

-webkit-transform:rotateY(-90deg) translateZ(100px);

-moz-transform:rotateY(-90deg) translateZ(100px);

}

.box div.side5 {

background:rgba( 0, 71, 171, 0.8);

-webkit-transform:rotateX(-90deg) translateZ(100px) rotate(180deg);

-moz-transform:rotateX(-90deg) translateZ(100px) rotate(180deg);

}

.box div.side6 {

background:rgba( 0, 171, 147, 0.8);

-webkit-transform:rotateX(90deg) translateZ(100px);

-moz-transform:rotateX(90deg) translateZ(100px);

}

@-moz-keyframes spin {

0% {-moz-transform: rotateX(0deg) rotateY(0deg);}

100% {-moz-transform: rotateX(1080deg) rotateY(360deg);}

}

@-webkit-keyframes spin {

0% {-webkit-transform: rotateX(0deg) rotateY(0deg);}

100% {-webkit-transform: rotateX(1080deg) rotateY(360deg);}

}

@-moz-keyframes rot {

0% {-moz-transform: rotate(0deg);}

100% {-moz-transform: rotate(360deg);}

}

@-webkit-keyframes rot {

0% {-webkit-transform: rotate(0deg);}

100% {-webkit-transform: rotate(360deg);}

}

</style>

<div id="holder">

<div class="box">

<div class="side1"><img src="1.jpg" /></div>

<div class="side2"><img src="2.jpg" /></div>

<div class="side3"><img src="3.jpg" /></div>

<div class="side4"><img src="4.jpg" /></div>

<div class="side5"><img src="5.jpg" /></div>

<div class="side6"><img src="6.jpg" /></div>

</div>

</div>

**Задания к лабораторной работе № 6**

**Задание 1.** Изучите, как создать анимацию. Для этого сохраните код примера1 и примера2 как веб-страницы и отобразите с помощью браузера. Для правильного отображения примера2 необходимо иметь 6 графических файлов.

**Задание 2.** Разработайте 2–3 своих примера с анимацией (например, прыгающий шар, метеоритный дождь, выезжающая машинка и т.д.). Сохраните в новом файле.

**Задание 3.** Создайте выпадающее меню с пунктами: лаб1, лаб2 …лаб6. Подпункты меню должны быть ссылками на созданные в процессе выполнения соответствующих работ ваши web-страницы. Названия подпунктов придумайте сами.