

ТЕМА 3. ИСПОЛЬЗОВАНИЕ СТИЛЕВЫХ СПЕЦИФИКАЦИЙ

1. Понятие таблицы стилей
2. Подключение таблиц стилей к документу
3. Селекторы
4. Свойства основных элементов
5. Управление фоном
6. Позиционирование элементов
7. Форматирование текстовых элементов
8. Особенности спецификации CSS3

1. Понятие таблицы стилей

CSS (каскадные таблицы стилей):

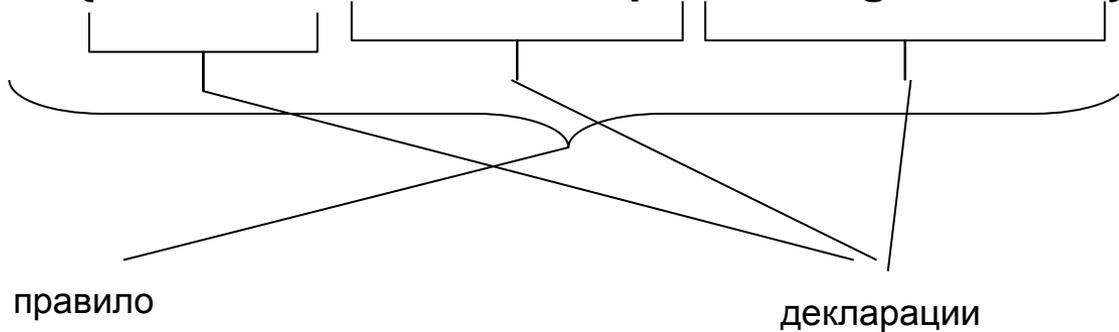
- Расширяют возможности по оформлению страниц
- Отделяют содержание от форматирования
- Каскадные – последовательно используется несколько таблиц, порядок их применения определен алгоритмом браузера
- Версии CSS 2, CSS3

CSS

- Таблица стилей состоит из последовательности правил.
- Правило состоит из селектора и набора деклараций.

Например:

```
H1 {color: red; font-size: 14pt; text-align: center;}
```



- Таблица стилей может размещаться либо в тексте страницы, либо в отдельном файле
- Таблицы стилей обеспечивают разделение контента и параметров форматирования
- Для применения таблиц стилей необходимо разработать правила, указать какую таблицу использовать и какое правило к какому элементу применить

2. Подключение таблиц стилей

Связывание документа с таблицей стилей, размещенной в файле:

```
<LINK REL=STYLESHEET TYPE="text/css"  
  HREF="http://www.myserver.com/mysheet.css">
```

Встраивание таблицы стилей в документ:

```
<STYLE>
```

описание стилей

```
</STYLE>
```

Импорт из файла или другой страницы:

```
<STYLE TYPE="text/css">
```

```
@import url(http://www.myserver.com/mystyle.css);
```

```
</STYLE>
```

Задание стиля для отдельного элемента с помощью атрибута
STYLE:

```
STILE="color:blue"
```

таблица стилей в отдельном файле

Index.htm

```
<!DOCTYPE HTML>
<html>
<head>
...
<LINK REL=STYLESHEET
TYPE="text/css"
HREF="my.css">
</head>
...
```

my.css

```
...
H1 {color: red;
font-size: 14pt;
text-align: center;}
...
```

Таблица стилей в тексте страницы

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
...
```

```
<style >
```

```
H1 {color: red; font-size: 14pt; text-align: center;}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
...
```

Таблица стилей в отдельном файле

Index.htm

```
<!DOCTYPE html>
<html>
  <head>
  ...
  <style >
  @import url(my.css);
  </style></head>
  ...
```

my.css

```
...
H1 {color: red;
font-size: 14pt;
text-align: center;}
...
```

Таблица стилей в теге

```
<h4 STYLE="font-weight:normal;  
font-style:italic;  
font-size:40pt;">ПРОБА</H4>
```

3. Селекторы

<селектор> {<имя свойства>:<значение>;..}

H1 {color: red; font-size: 14pt; text-align: center;}

P,LI {font-size: 12pt;}

Виды селекторов:

имена тегов;

классовые селекторы (Class Selectors);

ID селекторы (ID Selectors);

контекстные селекторы (Contextual Selectors);

псевдоклассы и псевдоэлементы (Pseudo Classes, Pseudo element).

Основные селекторы

Классовые селекторы:

H1.bl {color:blue; size:20pt} или .bl {color:blue}

<H1 CLASS="bl">TEXT</H1>

ID селекторы:

#rd {color:red}

<H1 ID="rd" >TEXT</H1>

<H1 CLASS="bl" ID="rd" >TEXT</H1>

Контекстные селекторы:

OL LI {list-style-type: decimal}

UL LI { list-style-type: square}

H1 B, H2 B, H1 EM, H2 EM { color: red }

.reddish H1 { color: red }

DIV.sidenote H1 { font-size: large }

ДОПОЛНИТЕЛЬНЫЕ селекторы

Селекторы на основе атрибутов

H1[title] {color:blue; size:20pt}

```
<H1 title="заголовок">TEXT</H1>
```

a[href="foo"] {color:red}

```
<a href="foo" >TEXT</a>
```

a[href^="http"] {color:red}

a[href\$=".JPG"] {color:red}

Псевдо-классы :before и :after

```
.clearfix:after {content: "";}...
```

Наследование и приоритет

- Наследование: border-collapse, border-spacing, caption-side, color, cursor, direction, empty-cells, font-family, font-size, font-style, font-variant, font-weight, font, letter-spacing, line-height, list-style-image, list-style-position, list-style-type, list-style, quotes, text-align, text-indent, text-transform, visibility, white-space, word-spacing
- При применении разных правил к одному и тому же элементу учитываются источник (авторский, пользовательский, браузер), тип селектора, порядок следования и директива !important
- Директива повышения приоритета !important:
div {font-size:40; color:black !important}

pr_2_2

Приоритеты при задании стилей в порядке возрастания

- Самым низким приоритетом обладают встроенные стили браузера
- Пользовательский стиль в браузере
- Наследуемые стили от родительских элементов
- CSS-стили, заданные во внешних файлах, применяются последовательно в порядке подключения. Это же правило относится и к взаимному расположению правил внутри таблицы стилей.
- По типу примененных селекторов:
 - теги (h1, div, p...)
 - классы и псевдоклассы (.item { }, :hover ...)
 - атрибуты ([attr="value"], [type="checkbox"] ...)
 - ID-селекторы (#main ...)
- Стили, заданные в атрибуте style тега (<div style="width : 203px"></div>)
- Наиболее высоким приоритетом обладает декларация с сопроводительным словом !important

4. Свойства основных элементов

Группы свойств элементов:

область размещения, позиционирование, фон, шрифт, текст, списки и т.д.

Единицы измерения:

px(пикселы), in(дюймы), cm(сантиметры), mm(миллиметры), pt(1pt=1/72in), pc(1pc=12pt), % (проценты от род. элемента), em (текущий размер шрифта)

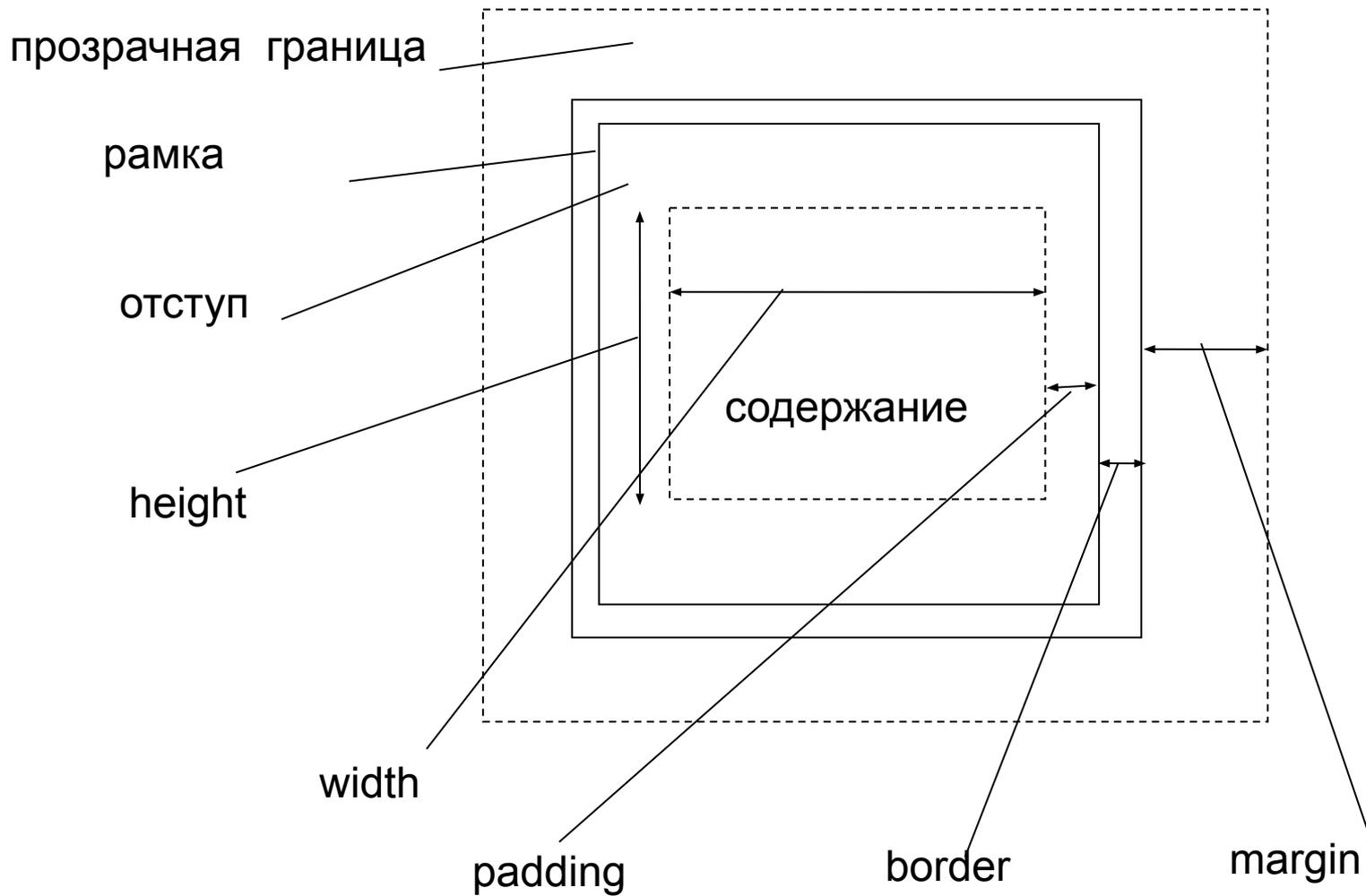
Цвет:

именем, “red” и т.д.

шестнадцатеричным значением, #6669FF

с помощью функции rgb, rgb(100,100,150) или rgb(50%,60%,50%)

Область размещения элемента



Свойства, характеризующие область размещения элемента

Имя	Значение	Примечание
width	<число><размерность> или auto	px, in, ..., pc, pt, %, em
height	<число><размерность> или auto	px, in, ..., pc, pt, %, em
margin	<число><размерность> или auto	px, in, ..., pc, pt, %, em
padding	<число><размерность> или auto	px, in, ..., pc, pt, %, em
border-width	<число><размерность> или auto	px, in, ..., pc, pt, %, em
border-color	black, ..., #000000, ..., rgb()	rgb(r,g,b) или rgb(r%,g%,b%)
border-style	none, dotted, solid, double ...	Разные варианты рамок

Задание свойств

- Свойства и их значения:

```
Margin-left :40px;padding-top:30px;  
width:20%;max-width:900px;  
min-width:500px:
```

- Составные имена свойств:

```
border-color:black; border-style:dotted;  
border-width:40px;
```

- Обобщенное свойство:

```
border:" black dotted 40px";
```

Примеры записи обобщенных свойств:

```
div {border:"2 solid red"}
```

5. Управление фоном

- background-color: <цвет> | transparent | inherit

background-color: #9A8B7F;

background-color: maroon;

background-color: rgb(r,g,b);

background-color: rgba(r, g, b, a);

- background-image:url(путь к файлу) | none[, url(путь к файлу) | none]*

```
body { background-image:url(mammoth.gif), url(fon.gif),  
url(fon1.gif); }
```

- background-attachment: fixed | scroll | local

ФОН

- Несколько изображений

```
body { background-image:url(mammoth.gif), url(fon.gif), url(fon1.gif);
```

- Размер и положение

```
background-repeat: no-repeat, repeat-x, repeat;
```

```
background-position: center 100px, 0px 250px; }
```

```
background-size: 200px 50px;
```

px - в пикселях или любых других единицах измерения, принятых в CSS.

% - в процентах от размеров элемента, которому назначается фон.

cover – Растягивает фоновое изображение на весь блок.

contain – Растягивает фоновое изображение с сохранением пропорций до тех пор, пока его ширина или высота не упрётся в края блока.

auto - Исходный размер изображения (по умолчанию). В случае если значение auto присвоено только одной из сторон, то её размер вычисляется относительно размера второй стороны с указанным размером.

```
.fon1 {background-size: 200px 50px;}
```

```
.fon2 {background-size: 20% 80%;}
```

<http://www.webremeslo.ru/css3/glava5.html>

pr_4

6. *Позиционирование элементов*

Свойства, используемые для позиционирования

display	block, inline, inline-block, list-item, flex, grid, ..., none	определяет тип элемента
position	absolute, fixed, relative, static	задает систему позиционирования
top, right, left, bottom	<число> right, left, none	определяют положение элемента
float	<число>	обтекание
z-index		порядок наложения
visibility		управление видимостью
overflow	visible, hidden, auto, hidden, scroll, visible, collapse	режим отображения содержимого

display = {block, inline, inline-block, list-item, flex, grid, ..., none}

block - элемент показывается как блочный, происходит перенос строк в начале и в конце содержимого, т.е. в его строках даже при наличии места другие элементы не размещаются;

inline - элемент отображается как встроенный в строку, поэтому содержимое элементов начинается с того места, где окончился предыдущий;

inline-block - создается блочный элемент (можно задавать размеры), при этом рядом могут располагаться строчные элементы;

list-item - элемент выводится как блочный и добавляется маркер списка;

pg_5

flex – блок, пространство которого автоматически заполняется дочерними блоками;

grid - для блока определяется виртуальная сетка, к которой могут привязываться дочерние элементы;

none - временно удаляет элемент из документа, занимаемое им место не резервируется и Web-страница формируется так, словно элемента и не было.

pg_6

position={absolute, fixed, relative, static}

absolute - элемент удаляется из обычного потока документа подобно другим элементам, и его положение задается атрибутами `left`, `top`, `right` и `bottom` относительно края окна браузера;

fixed - это значение похоже на `absolute`, привязывается к указанной параметрами `left`, `top`, `right` и `bottom` точке на экране, но не меняет своего положения даже при пролистывании страницы, разные браузеры реагируют на это свойство очень по-разному;

relative - положение элемента устанавливается относительно его исходного места, добавление атрибутов `left`, `top`, `right` и `bottom` изменяет позицию элемента и сдвигает его в ту или иную сторону от первоначального расположения, в зависимости от применяемого параметра;

static - элементы отображаются как обычно, использование параметров `left`, `top`, `right` и `bottom` не изменяет местоположения.

По умолчанию - `static`.

float: left | right | none | inherit

left - выравнивает элемент по левому краю, а все остальные элементы обтекают его по правой стороне.

right - выравнивает элемент по правому краю, а все остальные элементы обтекают его по левой стороне.

none - обтекание элемента не задается.

inherit - наследует значение родителя.

свойство clear (запрет обтекания):

- ◆ left — элемент должен находиться ниже всех плавающих элементов, параметр float которых выставлен в left;
- ◆ right — элемент должен находиться ниже всех плавающих элементов, параметр float которых выставлен в right;
- ◆ both — контейнер должен находиться ниже всех плавающих элементов, параметр float которых выставлен в left или right;
- ◆ none — поведение элемента по умолчанию (т. е. он может располагаться на одной линии с остальными плавающими элементами).

pr_8

Управление видимостью элементов и КОНТЕНТА

visibility={visible, hidden, collapse}

visible - элемент отображается как видимый;

hidden - элемент становится невидимым или правильней сказать, полностью прозрачным, поскольку он продолжает участвовать в форматировании страницы;

collapse - применяется для содержимого ячеек таблиц, они реагируют, словно к ним было добавлено стилевое свойство `display` со значением `none` (строки и колонки убираются, а таблица перестраивается), Значение поддерживается не всеми браузерами.

overflow={auto, hidden, scroll, visible}

overflow управляет отображением содержания блочного элемента, если оно целиком не помещается и выходит за область заданных размеров. Смысл принимаемых значений:

visible - отображается все содержание элемента, даже за пределами установленной высоты и ширины;

hidden - отображается только область внутри элемента, остальное будет обрезано;

scroll - всегда добавляются полосы прокрутки;

auto - полосы прокрутки добавляются только при необходимости.

7. Форматирование текстовых элементов

ИМЯ	ЗНАЧЕНИЯ	ПРИМЕЧАНИЕ
font	<набор значений>	обобщенное свойство
font-family	<список имен>	имена шрифтов
font-size	xx-small, x-small, small...	размер шрифта
font-style	normal, italic, oblique	начертание шрифта
font-variant	normal, small-caps	представление строчных букв
font-weight	normal, bold, bolder, 100, 200, ... 900	насыщенность шрифта

Свойства текстовых элементов (текст в целом)

ИМЯ	ЗНАЧЕНИЯ	ПРИМЕЧАНИЕ
letter-spacing	<число>, normal	интервал между символами
line-height	<число>, <число>%, normal	межстрочный интервал
text-align	left, right, center, justify	горизонтальное выравнивание
text-decoration	none, underline, overline, ...	вариант оформления
text-indent	<число>, <число>%	отступ первой строки
vertical-align	baseline, sub, super, top, ...	вертикальное выравнивание относительно родителя или окружения
word-spacing	<число>, normal	интервал между словами

Свойства списков

Имя	Значения	Примечание
list-style	<набор значений>	обобщенное свойство
list-style-image	url ("путь"), none	изображение маркера
list-style-position	outside, inside	размещение маркера
list-style-type	disc, circle, square, ...	вид маркера

Например, list-style:"url('bg.im') inside"

cursor: auto (вид курсора по умолчанию для текущего элемента), crosshair, default, e-resize, help, move, n-resize, ne-resize, nw-resize, pointer, progress, s-resize, se-resize, sw-resize, text, w-resize, wait, url('путь к курсору')

8. Особенности спецификации CSS3

- закруглённые рамки;
- создание тени к элементам;
- линейные и сферические градиенты;
- возможность задать прозрачный цвет;
- возможность задать свой шрифт;
- трансформации элементов;
- анимация.

Вендорные префиксы

Префикс	Производитель	Браузер	Браузерный движок
-o-, -op-, -xv-	Opera Software	Opera	Presto
-moz-	проект Mozilla	Firefox, SeaMonkey, Camino и др.	Gecko
-ms-	Microsoft	Internet Explorer 8	Trident
-khtml-	проект KDE	Safari до версии 3, Konqueror и др.	KHTML послужил основой для WebKit
-webkit-	Apple	Safari 3+, Google Chrome и др.	WebKit

-ms-filter:"progid:DXImageTransform.Microsoft.Alpha(Opacity=50)"; /* IE 8*/
-moz-opacity:0.5; /* Mozilla 1.6 */
-khtml-opacity:0.5; /* Konqueror 3.1, Safari 1.1 */

Закругления

border-radius

border-radius: 13em 0.5em/1em 0.5em 20em;

Число значений	Результат
1	Радиус указывается для всех четырех уголков.
2	Первое значение задает радиус верхнего левого и нижнего правого уголка, второе значение — верхнего правого и нижнего левого уголка.
3	Первое значение задает радиус для верхнего левого уголка, второе — одновременно для верхнего правого и нижнего левого, а третье — для нижнего правого уголка.
4	По очереди устанавливает радиус для верхнего левого, верхнего правого, нижнего правого и нижнего левого уголка.



pr_12

Тень и градиент

box-shadow, text-shadow

Значения свойства:

none - отменяет добавление тени.

inset - тень выводится внутри элемента

4 числа - сдвиг по x, сдвиг по y, размытие, растяжение

цвет - цвет тени в любом доступном CSS формате (по умолчанию тень черная).

```
box-shadow: inset 0px 3px 20px 3px #f00, 3px 10px ;
```

pr_13, pr_13_1

Градиентная заливка:

```
linear-gradient(to top, #fefcea, #f1da36);
```

```
filter:progid:DXImageTransform.Microsoft.gradient(startColorstr='#cccccc',  
endColorstr='#000000'); /* для IE */
```

```
background:-webkit-gradient(linear, left top, left bottom, from(#ccc), to(#000)); /* для webkit  
*/
```

```
background: -moz-linear-gradient(top, #ccc, #000); /* для firefox 3.6+ */
```

Вместо left, top можно ставить координаты:

```
background: -webkit-gradient(linear, 30% 30%, 70% 70%, from(#ccc), to(#000)); /* для  
webkit */
```

Прозрачность

.o1 {opacity: 0;} /* Полная прозрачность */

.o2 {opacity: 0.05;} /* Почти совсем
прозрачный */

.o3 {opacity: .3;} /* Можно и так писать без
нуля вначале */

.o4 {opacity: 0.5;} /* Полупрозрачный */

.o5 {opacity: 0.874;} /* Можно задавать очень
точную степень прозрачности */

.o6 {opacity: 1;} /* Совсем не прозрачный */

Шрифт

```
<style>  
  @font-face {  
    font-family: Pompadur; /* Имя шрифта */  
    src: url(fonts/pompadur.ttf); /* Путь к файлу со  
шрифтом */  
  }  
  P {  
    font-family: Pompadur;  
  }  
</style>
```

трансформации

- **transform:** функция;

Функция:

rotate

scale

scaleX

scaleY

...

pr_18

- **transition:** переход [, переход]*

Переход:

None

transition-property – список значений свойств

transition-duration – длительность

transition-timing-function – функция изменения времени

transition-delay – задержка перед началом

например, **transition:background 3s ease-out 1s**

pr_17

Анимация

Animation - это обобщенное свойство:

`animation-name` — имя анимации;

`animation-duration` — время проигрывания анимации;

`animation-timing-function` — метод расчета промежуточных значений свойств для анимации;

`animation-delay` — задержка анимации;

`animation-iteration-count` — количество циклов анимации;

`animation-direction` — направление анимации;

`animation-play-state` — определяет, проигрывается ли анимация или стоит на паузе.

anima

Список анимированных (для Safari и Chrome используется @-webkit-keyframes) переходов(ключевых кадров).

Здесь <селектор кадра> — обозначение, которое определяет ключевой кадр (from или 0%, n%, to или 100%).

```
@keyframes <имя анимации> {  
  <селектор кадра 1> {  
    <свойство 1>:<значение 1>;  
    <свойство 2>:<значение 1>;  
  }  
  <селектор кадра 2> {  
    <свойство 1>:<значение 2>;  
    <свойство 2>:<значение 2>;  
  }  
  .....  
  <селектор кадра n> {  
    <свойство 1>:<значение n>;  
    <свойство 2>:<значение n>;  
  }  
}
```

```
@-webkit-keyframes 'movement'  
{  
  from {top: 50px; left: 0px; }  
  50% {top: 150px;left: 100px;}  
  to   {top: 50px; left: 300px; }  
}
```

Задание

- Опробовать разные селекторы и различное размещение таблиц стилей
- Выполнить задания из файла ***задание_1***
- Выполнить задания из файла ***задание_2***
- Создать страницу на основе шаблона из папки ***зад_шаб***, максимально изменить оформление, разместить собственный контент и ссылки на свои страницы.