

Занятие 3. Верстка сайтов. Frame, FORM, CSS

Тег FRAME

- *Разбивает экран на различные страницы.*
- *Помогает делать дублирующие части.*

Тег FRAME. Применение.

```
<HTML>
<HEAD>
<TITLE>Простой документ с фреймами</TITLE>
</HEAD>
<FRAMESET cols="20%, 80%">
  <FRAMESET rows="100, 200">
    <FRAME src="contents_of_frame1.html">
    <FRAME src="contents_of_frame2.gif">
  </FRAMESET>
<FRAME src="contents_of_frame3.html">
<NOFRAMES>
  <P>В этом документе содержится:
  <UL>
    <LI><A href="contents_of_frame1.html">Миленький текстик</A>
    <LI><IMG src="contents_of_frame2.gif" alt="Симпатичная картинка">
    <LI><A href="contents_of_frame3.html">Еще славный текстик</A>
  </UL>
</NOFRAMES>
</FRAMESET>
</HTML>
```

contents_of_frame1.html

contents_of_frame3.html



Правда о FRAME

- *Изживший тег для использования, был популярен в 90-е.*
- *Подходит для ленивых(не нужно интегрировать сайт с CMS).*
- *Плохо индексируется поисковиками.*

Плюсы

При помощи конструкции frameset существует возможность подключать какие-либо формы от других источников (например формы регистрации в CRM системах).

Тег FORM

- *Даёт возможность взять с пользователя информацию.*
- *Да, анкеты, формы обратной связи, авторизации и т.д. сделаны с помощью тега <FORM> и элементов управления*

Тег FORM. Применение.

Синтаксис

```
<form action="URL"> ... </form>
```

Параметры

[action](#) Адрес программы или документа, которые обрабатывают данные формы.

[enctype](#) MIME-тип информации формы.

[method](#) Метод протокола HTTP.

[name](#) Имя формы.

[target](#) Имя окна или фрейма, куда обработчик будет загружать возвращаемый результат.

Тег FORM. INPUT.

Основной параметр тега <input>, определяющий вид элемента — type. Он позволяет задавать следующие элементы формы:

- текстовое поле (text),
- поле с паролем (password),
- переключатель (radio),
- флажок (checkbox),
- скрытое поле (hidden),
- кнопка (button),
- кнопка для отправки формы (submit),
- кнопка для очистки формы (reset),
- поле для отправки файла (file) и кнопка с изображением (image).

Текстовое поле

Параметры:

- **name** - имя элемента,
- **type** - тип элемента (в данном случае - text),
- **size** - размер текстового поля в символах, которые одновременно будут видны, при вводе большего количества символов, они будут прокручиваться,
- **maxlength** - максимальное количество символов, которое можно ввести в поле, если опустить этот параметр, то число символов будет неограниченным,
- **value** - текст, который будет отображаться (его можно стереть), при отсутствии этого параметра поле будет пустым.

Возможны еще два параметра:

- **disabled** - блокирует поле от любых изменений,
- **readonly** - делает поле доступным только для чтения.

Флажки

Параметры:

- *type* - тип элемента (в данном случае - checkbox),
- *name* - имя элемента, указывает программе обработчику формы, какой пункт выбрал пользователь,
- *value* - значение элемента, указывает программе обработчику формы значение пункта, который выбрал пользователь. В нашем примере выбран пункт английский, следовательно, программа-обработчик получит:
`lan1="english"`,
- *checked* - им обычно помечают наиболее вероятные для выбора пункты, пользователь щелчком мыши может выбрать другие пункты.

Переключатели

В отличие от флажков, можно выбрать только один пункт. В связи с этим значения параметра `name` должны быть одинаковы для всех элементов группы. Параметр `type="radio"`, все остальные такие же, как у флажков.

Пример:

```
<form name="forma1"> Укажите ваш пол:<br>  
  <input type="radio" name="sex" value="man" checked>мужской  
  <input type="radio" name="sex" value="woman"> женский  
</form>
```

Многострочное текстовое поле

Для объемных текстов, например для почтовых сообщений, удобно использовать именно этот элемент. Он создается тегами `<textarea>` `</textarea>` и имеет следующие параметры:

- *name* - имя поля,
- *cols* - ширина поля в символах,
- *rows* - количество строк текста, видимых на экране,
- *wrap* - способ переноса слов:
 - *off* - переноса не происходит,
 - *virtual* - перенос отображается, но на сервер поступает неделимая строка,
 - *physical* - перенос и на экране и при поступлении на сервер.
- *disabled* - неактивное поле,
- *readonly* - разрешено только чтение.

Раскрывающиеся списки

Списки бывают с возможностью выбора одного элемента и с множественным выбором. Задаются и те, и другие с помощью тегов `<select>` `</select>`, внутри которых располагаются элементы значений, заданных тегом `<option>`. Рассмотрим параметры этих тегов:

`<select>`:

- *name* - имя списка. Каждый выбранный элемент списка при передаче на сервер будет иметь вид: `name.value`, где значение (`value`) берется из тега `option`.
- *size* - определяет количество видимых элементов в списке: 1 - простой раскрывающийся список, больше 1 - список с полосой прокрутки.
- *multiple* - разрешает выбор нескольких элементов списка.

`<option>`:

- *selected* - им помечают наиболее вероятный для выбора элемент списка, если список со множественным выбором, то можно пометить несколько пунктов.
- *value* - значение, которое будет отправлено серверу, если пункт выбран.

Кнопки

submit - кнопка отправки содержимого формы web-серверу. Ее параметры:

- *type="submit"* - тип кнопки,
- *name* - имя кнопки,
- *value* - надпись на кнопке.

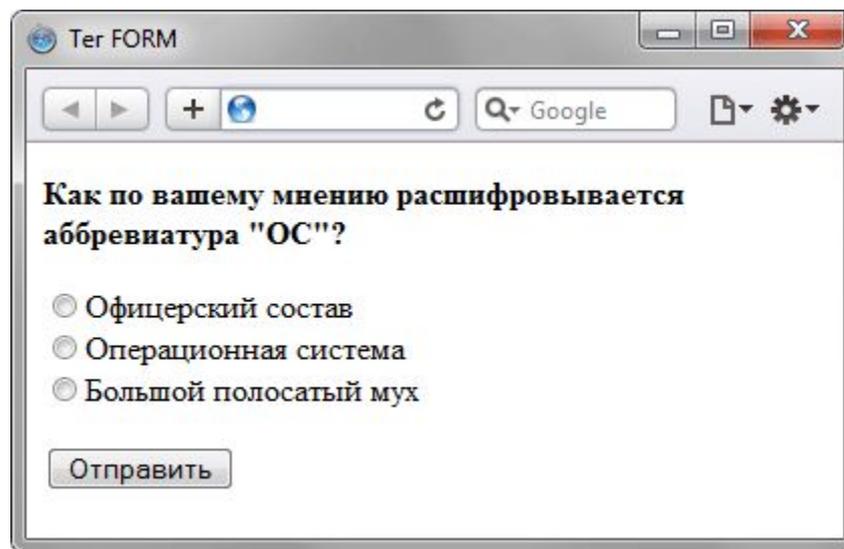
image - графическая кнопка отправки содержимого формы web-серверу. Для ее использования необходимо подготовить картинку кнопки, а потом использовать ее в виде кнопки. Ее параметры:

- *type="image"* - тип графической кнопки,
- *name* - имя кнопки,
- *src* - адрес картинки для кнопки.

reset - кнопка, позволяющая восстановить все значения по умолчанию в форме.

button - произвольная кнопка, ее действия назначаются вами, т.е. сама она делать ничего не умеет.

Пример в окне браузера



Тег FORM. Правда.

- *Тег, без которого не возможно сделать связь с клиентом сайта.*
- *Наиболее распространен.*
- *Имеет множество вариантов использования.*
- *<TEXTAREA> - единственный тег, который делается не*
- *<Input type="...">*

Каскадные таблицы стилей (CSS)

Стилем или CSS (Cascading Style Sheets, каскадные таблицы стилей) называется набор параметров форматирования, который применяется к элементам документа, чтобы изменить их внешний вид.

История

CSS — одна из широкого спектра технологий, одобренных консорциумом [W3C](#) CSS — одна из широкого спектра технологий, одобренных консорциумом W3C и получивших общее название «стандарты Web»[\[1\]](#) CSS — одна из широкого спектра технологий, одобренных консорциумом W3C и получивших общее название «стандарты Web»[\[1\]](#). В 1990-х годах стала ясна необходимость стандартизировать Web, создать какие-то единые правила, по которым программисты и [веб-дизайнеры](#) CSS — одна из широкого спектра технологий, одобренных консорциумом W3C и получивших общее название «стандарты Web»[\[1\]](#). В 1990-х годах стала ясна необходимость стандартизировать Web, создать какие-то единые правила, по которым программисты и веб-дизайнеры проектировали бы сайты. Так появились языки [HTML](#) CSS — одна из широкого спектра технологий, одобренных консорциумом W3C и получивших общее название «стандарты Web»[\[1\]](#). В 1990-х годах стала ясна необходимость стандартизировать Web, создать какие-то единые правила, по которым программисты и веб-дизайнеры проектировали бы сайты. Так появились языки HTML 1.01 и [XHTML](#) и стандарт CSS

Уровень 1 (CSS1)

Рекомендация W3C, принята [17 декабря 1996 года](#) Рекомендация W3C, принята 17 декабря 1996 года, откорректирована [11 января 1999 года](#) Рекомендация W3C, принята 17 декабря 1996 года, откорректирована 11 января 1999 года [\[2\]](#). Среди возможностей, предоставляемых этой рекомендацией:

- Параметры шрифтов. Возможности по заданию гарнитуры и размера шрифта, а также его стиля — обычного, курсивного или полужирного.
- Цвета. Спецификация позволяет определять цвета текста, фона, рамок и других элементов страницы.
- Атрибуты текста. Возможность задавать межсимвольный интервал, расстояние между словами и высоту строки (то есть межстрочные отступы)
- Выравнивание для текста, изображений, таблиц и других элементов.
- Свойства блоков, такие как высота, ширина, внутренние (padding) и внешние (margin) отступы и рамки. Так же в спецификацию входили ограниченные средства по позиционированию элементов, такие как float и clear.

Уровень 2 (CSS2)

Рекомендация W3C, принята [12 мая](#) Рекомендация W3C, принята 12 мая [1998 года](#) Рекомендация W3C, принята 12 мая 1998 года [\[3\]](#). Построена на CSS1 с сохранением обратной совместимости. Добавление к функциональности:

- Блочная вёрстка. Появились относительное, абсолютное и фиксированное позиционирование. Позволяет управлять размещением элементов по странице без [табличной вёрстки](#).
- Типы носителей. Позволяет устанавливать разные стили для разных носителей (например [монитор](#) Типы носителей. Позволяет устанавливать разные стили для разных носителей (например монитор, [принтер](#) Типы носителей. Позволяет устанавливать разные стили для разных носителей (например монитор, принтер, [КПК](#)).
- Звуковые таблицы стилей. Определяет голос, громкость и т. д. для звуковых носителей (например для слепых посетителей сайта).
- Страничные носители. Позволяет, например, установить разные стили для элементов на чётных и нечётных страницах при печати.
- Расширенный механизм селекторов.
- Указатели.
- Генерируемое содержание. Позволяет установить текст или картинку, который будет отображаться до или после нужного элемента.

Способы подключения CSS

Связанные стили

```
<link rel="stylesheet" type="text/css" href="mysite.css">
```

Глобальные стили

```
<style type="text/css">
```

```
H1 {  
    font-size: 120%;  
    ...  
}  
</style>
```

Внутренние стили

```
<h1 style="font-size: 36px; font-family: Times, serif; color: red">Заголовок 1</h1>
```

Базовый синтаксис

Селектор — это некоторое имя стиля, для которого добавляются параметры форматирования

Селектор { свойство1: значение; свойство2: значение; }

...

```
<style type="text/css">
  h1 { color: #a6780a; font-weight: normal; }
  h2 { color: olive; border-bottom: 2px solid black; }
</style>
```

...

...

```
<h1>Заголовок 1</h1>
<h2>Заголовок 2</h2>
```

...

Правила применения стилей

- **Расширенная форма записи**

```
td { background: olive; }
```

```
td { color: white; }
```

```
td { border: 1px solid black; }
```

- **Компактная форма записи**

```
td {
```

```
  background: olive;
```

```
  color: white;
```

```
  border: 1px solid black;
```

```
}
```

- **Комментарии**

```
div {
```

```
  width: 200px;    /* Ширина блока */
```

```
  margin: 10px;   /* Поля вокруг элемента */
```

```
  float: left;    /* Обтекание по правому краю */
```

```
}
```

Классы

Классы применяют, когда необходимо определить стиль для индивидуального элемента веб-страницы или задать разные стили для одного тега.

Тег.Имя класса { свойство1: значение; свойство2: значение; ... }

Классы

```
P.cite { /* Абзац с классом cite */  
  color: navy; /* Синий цвет текста */  
  margin-left: 20px; /* Отступ слева */  
  border-left: 1px solid navy; /* Граница слева от текста */  
  padding-left: 15px; /* Расстояние от линии до текста */  
}
```

```
<p class="cite">текст...</p>
```

Идентификаторы

Идентификатор (называемый также «ID селектор») определяет уникальное имя элемента, которое используется для изменения его стиля и обращения к нему через скрипты.

#Имя идентификатора { свойство1: значение; свойство2: значение; ... }

Идентификаторы

```
<style type="text/css">
  #help { position: absolute; /* Абсолютное позиционирование */
  left: 160px; /* Положение элемента от левого края */
  top: 50px; /* Положение от верхнего края */
  width: 225px; /* Ширина блока */
  padding: 5px; /* Поля вокруг текста */
  background: #f0f0f0; /* Цвет фона */
  }
</style>
```

```
<div id="help"> Этот элемент помогает в случае, когда вы находитесь в осознании
  того факта, что совершенно не понимаете, кто и как вам может помочь. Именно
  в этот момент мы и подсказываем, что помочь вам никто не сможет.
</div>
```

Группирование

Селектор 1, Селектор 2, ... Селектор N { Описание правил стиля }

```
H1 {  
  font-family: Arial, Helvetica, sans-serif;  
  font-size: 160%;  
  color: #003;  
}
```

```
H2 {  
  font-family: Arial, Helvetica, sans-serif;  
  font-size: 160%;  
  color: #003;  
}
```

```
H3 {  
  font-family: Arial, Helvetica, sans-serif;  
  font-size: 160%;  
  color: #003;  
}
```

Группирование

```
H1, H2, H3 { font-family: Arial, Helvetica, sans-serif; }
```

```
H1 {  
  font-size: 160%;  
  color: #003;  
}
```

```
H2 {  
  font-size: 135%;  
  color: #333;  
}
```

```
H3 {  
  font-size: 120%;  
  color: #900;  
}
```

Псевдоклассы

Псевдоклассы определяют динамическое состояние элементов, которое изменяется со временем или с помощью действий пользователя, а также положение в дереве документа.

Селектор:Псевдокласс { Описание правил стиля }

Примером такого состояния служит текстовая ссылка, которая меняет свой цвет при наведении на нее курсора мыши.

Псевдоклассы

- active
- link
- focus
- hover

Пример

```
<style type="text/css">
  INPUT:focus {
    color: red; /* Красный цвет текста */
  }
</style>
```

Псевдоэлементы

Псевдоэлементы позволяют задать стиль элементов не определенных в дереве элементов документа, а также генерировать содержимое, которого нет в исходном коде текста.

Селектор:Псевдоэлемент { Описание правил стиля }

Псевдоэлементы

- **after**

```
<style type="text/css">
  P.new:after {
    content: " - Новьё!"; /* Добавляем после текста абзаца */
  }
</style>
```

- **before**

```
<style type="text/css">
  UL {
    padding-left: 0; /* Убираем отступ слева */
    list-style-type: none; /* Прячем маркеры списка */
  }
  LI:before {
    content: "\20aa "; /* Добавляем перед элементом списка символ в юникоде */ }
</style>
```

- **first-letter**

```
P:first-letter {
  font-family: 'Times New Roman', Times, serif; /* Гарнитура шрифта первой буквы */
  font-size: 200%; /* Размер шрифта первого символа */
  color: red; /* Красный цвет текста */ }
</style>
```