

Фреймы разделяют окно браузера на отдельные области, расположенные рядом друг с другом. В каждую из таких областей загружается самостоятельная веб-страница.

Достоинства фреймов

Простота

С помощью фреймов веб-страница разграничивается на две области, которые содержат навигацию по сайту и его контент. Механизм фреймов позволяет открывать документ в одном фрейме, по ссылке, нажатой в совершенно другом фрейме. Такое разделение веб-страницы на составляющие интуитивно понятно и логически обусловлено.

Быстрота

Для верстки без фреймов характерно размещение на одной странице и навигации и содержания. Это увеличивает объем каждой страницы и в сумме может существенно повлиять на объем загружаемой с сайта информации. А так как фреймы используют разделение информации на части, страницы с ними будут загружаться быстрее.

Размещение

Фреймы предоставляют уникальную возможность — размещение информации точно в нужном месте окна браузера. Так, можно поместить фрейм внизу браузера и независимо от прокручивания содержимого, эта область не изменит своего положения.

Изменение размеров областей

Можно изменять размеры фреймов «на лету», чего не позволяет сделать традиционная верстка HTML.

Загрузка

Загрузка веб-страницы происходит только в указанное окно, остальные остаются неизменными. С помощью языка JavaScript можно осуществить одновременную загрузку двух и более страниц во фреймы.

Недостатки фреймов

Навигация

Пользователь зачастую оказывается на сайте, совершенно не представляя, куда он попал, потому что всего лишь нажал на ссылку, полученную в поисковой системе. Чтобы посетителю сайта было проще разобраться, где он находится, на каждую страницу помещают название сайта, заголовок страницы и навигацию. Фреймы, как правило, нарушают данный принцип, отделяя заголовок сайта от содержания, а навигацию от контента. Представьте, что вы нашли подходящую ссылку в поисковой системе, нажимаете на нее, а в итоге открывается документ без названия и навигации. Чтобы понять, где мы находимся или посмотреть другие материалы, придется редактировать путь в адресной строке, что в любом случае доставляет неудобство.

Плохая индексация поисковыми системами

Поисковые системы плохо работают с фреймовой структурой, поскольку на страницах, которые содержат контент, нет ссылок на другие документы.

Внутренние страницы нельзя добавить в «Закладки»

Фреймы скрывают адрес страницы, на которой находится посетитель, и всегда показывают только адрес сайта. По этой причине понравившуюся страницу сложно поместить в закладки браузера.

Несовместимость с разными браузерами

Параметры фреймов обладают свойством совершенно по-разному отображаться в различных браузерах. Причём противоречие между ними настолько явное, что одни и те же параметры интерпретируются браузерами совершенно по-своему.

Создание фреймов

При использовании фреймов необходимо как минимум три HTML-файла: первый определяет фреймовую структуру и делит окно браузера на две части, а оставшиеся два документа загружаются в заданные окна. Количество фреймов не обязательно равно двум, может быть и больше, но никак не меньше двух, иначе вообще теряется смысл применения фреймов.

Рассмотрим этапы создания. Нам понадобится три файла: `index.html`— определяет структуру документа, `menu.html`— загружается в левый фрейм и `content.html`— загружается в правый фрейм. Из них только `index.html` отличается по структуре своего кода от других файлов.

В случае использования фреймов в первой строке кода пишется следующий тип документа.

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

Данный `<!DOCTYPE>` указывают браузеру, что он имеет дело с фреймами, эта строка кода является обязательной. Контейнер `<head>` содержит типовую информацию вроде кодировки страницы и заголовка документа. Вот только учтите, что заголовок остается неизменным, пока HTML-файлы открываются внутри фреймов.

```
<frameset cols="100,*">
<frame src="menu.html" name="MENU">
<frame src="content.html" name="CONTENT">
</frameset>
```

В данном примере окно браузера разбивается на две колонки с помощью атрибута `cols`, левая колонка занимает 100 пикселей, а правая— оставшееся пространство, заданное символом звездочки. Ширину или высоту фреймов можно также задавать в процентном отношении, наподобие таблиц.

В теге `<frame>` задается имя HTML-файла, загружаемого в указанную область с помощью атрибута `src`. В левое окно будет загружен файл, названный `menu.html`, а в правое — `content.html`. Каждому фрейму желательно задать его уникальное имя, чтобы документы можно было загружать в указанное окно с помощью атрибута `name`.

Тег `<frameset>`

Определяет структуру фреймов на веб-странице. Фреймы разделяют окно браузера на отдельные области, расположенные вплотную друг к другу. В каждую из таких областей загружается самостоятельная веб-страница определяемая с помощью тега `<frame>`. С помощью фреймов веб-страница делится на два или более документа, которые обычно содержат навигацию по сайту и его контент. Механизм фреймов позволяет открывать документ в одном фрейме, по ссылке, нажатой в совершенно другом фрейме. Тег `<frameset>` заменяет собой элемент `<body>` на веб-странице. Допустимо использовать вложенную структуру элементов, это позволяет разбить один фрейм на две и более области. Закрывающий тег — Обязателен.

<u>border</u>	Толщина границы между фреймами.
<u>bordercolor</u>	Цвет линии границы.
<u>cols</u>	Устанавливает ширину или пропорции фреймов в виде колонок.
<u>frameborder</u>	Определяет, отображать рамку вокруг фрейма или нет.
<u>framespacing</u>	Аналог атрибута <code>border</code> , задает ширину границы.
<u>rows</u>	Задаёт размер или пропорции фреймов в виде строк.

Тег `<frame>`

Тег `<frame>` определяет свойства отдельного фрейма, на которые делится окно браузера. Этот элемент должен располагаться в контейнере `<frameset>`, который к тому же задает способ разметки страницы на отдельные области. В каждую из таких областей загружается самостоятельная веб-страница определяемая с помощью атрибута `src`. Хотя обязательных атрибутов у тега `<frame>` и нет, рекомендуется задавать каждому фрейму его имя через атрибут `name`. Это особенно важно, если требуется по ссылке из одного фрейма загружать документ в другой. Закрывающий тег - Не требуется.

<u>bordercolor</u>	Цвет линии границы.
<u>frameborder</u>	Отображать рамку вокруг фрейма или нет.
<u>name</u>	Задаёт уникальное имя фрейма.
<u>noresize</u>	Определяет, можно изменять размер фрейма пользователю или нет.
<u>scrolling</u>	Способ отображения полосы прокрутки во фрейме.
<u>src</u>	Путь к файлу, предназначенному для загрузки во фрейме.

Рассмотрим более сложный пример уже с тремя фреймами.

В данном случае опять используется тег `<frameset>`, но два раза, причем один тег вкладывается в другой. Горизонтальное разбиение создается через атрибут `rows`, где для разнообразия применяется процентная запись.

```
<frameset rows="25%,75%">
<frame src="top.html" name="TOP" scrolling="no" noresize>
  <frameset cols="100,*">
    <frame src="menu.html" name="MENU">
    <frame src="content.html" name="CONTENT">
  </frameset>
</frameset>
</html>
```

Как видно из данного примера, контейнер `<frameset>` с атрибутом `rows` вначале создает два горизонтальных фрейма, но вместо второго фрейма подставляется еще один `<frameset>`, который повторяет уже известную вам структуру. Чтобы не появилась вертикальная полоса прокрутки, и пользователь не мог самостоятельно изменить размер верхнего фрейма, добавлены атрибуты `scrolling="no"` и `noresize`.

Ссылки во фреймах

В обычном HTML-документе при переходе по ссылке, в окне браузера текущий документ заменяется новым. При использовании фреймов схема загрузки документов отличается от стандартной. Основное отличие — возможность загружать документ в выбранный фрейм из другого. Для этой цели используется атрибут `target` тега `<a>`. В качестве значения используется имя фрейма, в который будет загружаться документ, указанный атрибутом `name`.

```
<frameset cols="100,*">
<frame src="menu2.html" name="MENU">
<frame src="content.html" name="CONTENT">
</frameset>
</html>
```

В приведенном примере фрейму присваивается имя `CONTENT`. Чтобы документ загружался в указанный фрейм, используется конструкция `target="CONTENT"`, как показано в примере:

```
<a href="text.html" target="CONTENT">Текст</a>
```

Имя фрейма должно начинаться на цифру или латинскую букву. В качестве зарезервированных имен используются следующие:

- `_blank` — загружает документ в новое окно;
- `_self` — загружает документ в текущий фрейм;
- `_parent` — загружает документ во фрейм, занимаемый родителем, если фрейма-родителя нет значение действует также, как `_top`;
- `_top` — отменяет все фреймы и загружает документ в полное окно браузера.

Границы между фреймами

Граница между фреймами отображается по умолчанию и, как правило, в виде трехмерной линии. Чтобы ее скрыть используется атрибут `frame border` тега `<frameset>` со значением 0. Однако в браузере Opera граница хоть и становится в этом случае бледной, все же остается. Для этого браузера требуется добавить `frame spacing = "0"`. Таким образом, комбинируя разные атрибуты тега `<frameset>`, получим универсальный код, который работает во всех браузерах. Линия при этом показываться никак не будет.

Учтите, что атрибуты `frame border` и `frame spacing` не являются валидными и не соответствуют спецификации HTML.

Если граница между фреймами все же нужна, в браузере она рисуется по умолчанию, без задания каких-либо атрибутов. Можно, также, задать цвет рамки с помощью атрибута `bordercolor`, который может применяться в тегах `<frameset>` и `<frame>`. Цвет указывается по его названию или шестнадцатеричному значению, а толщина линии управляется атрибутом `border`. Браузер Opera игнорирует этот атрибут и обычно отображает линию черного цвета.

Атрибуты `bordercolor` и `border` тега `<frameset>` также не являются валидными и не признаются спецификацией HTML.

Браузер Opera никак не изменяет цвет границы между фреймами, Internet Explorer устанавливает широкую границу практически сплошного цвета, а Firefox границу отображает в виде набора линий.

Изменение размеров фреймов

По умолчанию размеры фреймов можно изменять с помощью курсора мыши, наведя его на границу между фреймами. Для блокировки возможности изменения пользователем размера фреймов следует воспользоваться атрибутом **`noresize`** тега `<frame>`.

Атрибут `noresize` не требует никаких значений и используется сам по себе. Для случая двух фреймов этот атрибут можно указать лишь в одном месте. Естественно, если у одного фрейма нельзя изменять размеры, то у близлежащего к нему размеры тоже меняться не будут.

Полосы прокрутки

Если содержимое фрейма не помещается в отведенное окно, автоматически появляются полосы прокрутки для просмотра информации. В некоторых случаях полосы прокрутки нарушают дизайн веб-страницы, поэтому от них можно отказаться. Для управления отображением полос прокрутки используется атрибут **`scrolling`** тега `<frame>`. Он может принимать два основных значения: `yes` — всегда вызывает появление полос прокрутки, независимо от объема информации и `no` — запрещает их появление.

При выключенных полосах прокрутки, если информация не помещается в окно фрейма, просмотреть ее будет сложно. Поэтому `scrolling="no"` следует использовать осторожно.

Если атрибут `scrolling` не указан, то полосы прокрутки добавляются браузером только по необходимости, в том случае, когда содержимое фрейма превышает его видимую часть.

Плавающие фреймы

Разговор о фреймах будет неполным без упоминания плавающих фреймов. Так называется фрейм, который можно добавлять в любое место веб-страницы. Еще одно его название — встроенный фрейм, он называется так из-за своей особенности встраиваться прямо в тело веб-страницы.

Во фрейм можно загружать HTML-документ и прокручивать его содержимое независимо от остального материала на веб-странице. Размеры фрейма устанавливаются самостоятельно согласно дизайну сайта или собственных предпочтений.

Создание плавающего фрейма происходит с помощью тега **`<iframe>`**, он имеет обязательный атрибут `src`, указывающий на загружаемый во фрейм документ.

Ширина и высота фрейма устанавливается через атрибуты `width` и `height`. Заметьте, что если содержимое не помещается целиком в отведенную область, появляются полосы прокрутки.

Еще одно удобство плавающих фреймов состоит в том, что в него можно загружать документы по ссылке. Для этого требуется задать имя фрейма через атрибут name, а в теге <a> указать это же имя в атрибуте target.

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

<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8">
<title>Плавающий фрейм</title>
</head>
<body>
<p><a href="rgb.html" target="color">RGB</a> |
<a href="cmyk.html" target="color">CMYK</a> |
<a href="hsb.html" target="color">HSB</a></p>
<p><iframe src="model.html" name="color" width="100%" height="200"></iframe></p>
</body>
</html>
```

В данном примере добавлено несколько ссылок, они открываются во фрейме с именем color. Тег <iframe> проходит валидацию только при использовании переходного <!DOCTYPE>.

Тег <base>

Элемент <base> определен внутри контейнера <head> и инструктирует браузер относительно полного базового адреса текущего документа. Тег <base> предназначен для документов, в которых используется относительный адрес и эти документы могут переноситься в другую папку или даже на другой компьютер без потери связи. Браузер ищет тег <base>, определяет полный адрес документа и корректно загружает его. Например, если адрес документа указан как <base href="http://www.megasite.ru/hzchd/">, то при добавлении рисунков достаточно использовать относительный адрес . При этом полный путь к изображению будет http://www.megasite.ru/hzchd/images/labuda.gif, что позволяет браузеру всегда находить графический файл, независимо от того, где находится текущая веб-страница. Также можно применять и иерархическую систему пути с двумя точками. Так, если изображение добавляется как , то полный путь к файлу будет http://www.megasite.ru/images/labuda.gif.

Второе применение тега <base> — задание целевого окна для всех ссылок на текущей странице.

Закрывающий тег - Не требуется.

<u>href</u>	Адрес, который должен использоваться для указания полного пути к файлам.
<u>target</u>	Имя окна или фрейма, куда будет загружаться документ, открываемый по ссылке.

Элемент <base> не выводит никакого контента и выполняет исключительно служебную функцию — позволяет указать базовый URL, относительно которого будут устанавливаться другие адреса, например, для изображений и ссылок. Также <base> задаёт значение атрибута target, которое по умолчанию применяется ко всем ссылкам.

В документе разрешается иметь только один <base>.

Атрибут href

Предположим, все изображения для наших документов хранятся в папке assets/images. Чтобы постоянно не указывать этот путь перед именем файла, его можно вынести в значение атрибута href тега <base>.

Атрибут href будет действовать на все адреса, используемые в документе, включая ссылки, поэтому на них путь, заданный в <base> также будет распространяться.

Из-за того, что `<base>` влияет на все используемые ниже него адреса, его рекомендуется ставить в `<head>` раньше элементов со ссылками. Если перед `<base>` стоит, к примеру, `<link>`, то базовый адрес на него действовать не будет. Также базовый адрес не влияет на атрибут `manifest` в корневом элементе `<html>`.

В `<base>` допустимо указывать и адрес документа.

```
<base href="example/index.php">
```

Такой путь не надо подставлять прямолинейно, как это делалось в случае с папками. Адрес документа в `href` говорит, что относительно него следует задавать пути к файлам. В качестве примера рассмотрим следующую структуру файлов на сайте.

Допустимо использовать только один базовый адрес на странице. Если ошибочно добавлено несколько `<base>` с разными `href`, то применяется только первый, остальные игнорируются.

Атрибут `target`

Значение атрибута `target` определяет имя контекста, которое применяется для ссылок (`<a>`) и форм (`<form>`). В качестве имени контекста обычно указывается имя фрейма или ключевое слово: `_blank`, `_self`, `_parent` или `_top`.

В качестве примера рассмотрим добавление на страницу фрейма с именем `frame`, оно определяется атрибутом `name` тега `<iframe>`. Чтобы ссылки за пределами фрейма открывались внутри него, в элементе `<base>` укажем `target="frame"`, как показано в примере.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Ссылки во фрейме</title>
<base target="frame">
</head>
<body>
<p><a href="2.html">2</a><a href="3.html">3</a></p>
<p><iframe name="frame" src="1.html"></iframe></p>
</body>
</html>
```

Использование `<base>` в таком качестве равнозначно тому, что для каждой ссылки добавляется `target="frame"`.

Кроме использования имен допустимы следующие ключевые слова.

- `_blank`— открывает ссылку в новом окне или вкладке;
- `_self`— открывает ссылку в текущем окне или вкладке, равнозначно значению `""` (пустая строка);
- `_parent`— открывает ссылку в родительском контексте. Например, если такая ссылка нажата внутри фрейма, то она открывается не во фрейме, а на странице содержащей фрейм;
- `_top`— открывает ссылку в контексте верхнего уровня. Результат будет заметен при сложной фреймовой структуре, при наличии на странице одного фрейма, открытие ссылки внутри него подобно действию `_parent`.

Если одновременно требуются атрибуты `href` и `target`, они объединяются в пределах одного тега:

```
<base href="http://htmlbook.ru/content" target="_blank">
```

Надо отметить, что значения `_parent` и `_top` применяются довольно редко и перешли в HTML5 по наследству из HTML4, где активно применялись фреймы.