**Разбираем свойство float в CSS**

**Дата публикации:** 2010-02-22



Пару лет назад, когда разработчики впервые начали переходить к HTML-разметке без таблиц, CSS свойство float внезапно взяло на себя очень важную роль. Причина того, что свойство float стало так распространено, состояла в том, что по умолчанию, блочные элементы не выстраивались в линию рядом друг с другом в формате на основе столбца. Так как столбцы необходимы фактически в каждой схеме CSS, к этому свойству начали привыкать — и даже злоупотреблять им.

Свойство **float в CSS**, позволяет разработчику включать подобные таблице столбцы в разметку HTML без использования таблиц. Если бы не свойство **float**, то схемы размещения CSS не были бы возможны, кроме использования абсолютного и относительного позиционирования — которые были бы неаккуратными и сделали бы схему разметки неудобной в сопровождении.

В этой статье мы расскажем, **что такое свойство float и как оно влияет на элементы в специфических контекстах**. Мы также обратим внимание на некоторые из различий, которые могут возникнуть в связи с этим свойством в наиболее часто используемых браузерах. Наконец, мы продемонстрируем несколько практических применений свойства **float**. Это должно обеспечить также всестороннее и тщательное обсуждение этого свойства и его влияние на развитие CSS.

1. **Определение и синтаксис CSS-свойства Float**

Цель свойства float состоит в том, чтобы, вытолкнуть блочный элемент влево или вправо, взяв его из потока документа. Это позволяет естественным образом текущему информационному содержанию обвалакиваться вокруг плавающего элемента. Эта концепция является аналогичной той, которую вы видите каждый день в печатной литературе, где фотографии и другие графические элементы, выровненные по какой либо стороне, а информационное наполнение (обычно текст) течет естественно вокруг элемента выровненного вокруг левого или правого края.





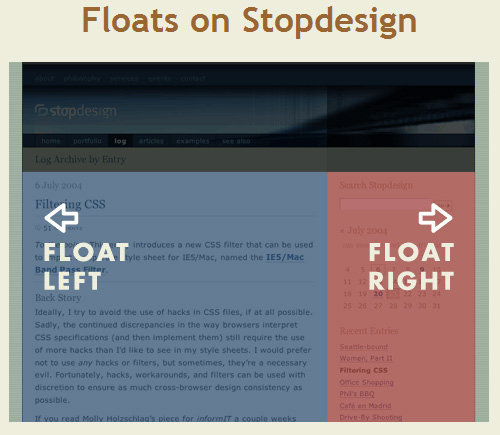
Практический курс по верстке адаптивного лендинга с нуля!

Научитесь с нуля верстать адаптивные лендинги на HTML5 и CSS3 за ближайшие 6 дней

[Узнать подробнее](https://webformyself.com/minikurs/html5/?utm_medium=systema&utm_source=adsinsingle&utm_campaign=html5_new)

На изображении выше раздел “Читатели сайта ”[.net magazine](http://www.netmag.co.uk/) с 3-мя фотографиями читателей, которые выровнены по левому краю в своих столбцах с обертыванием текста вокруг изображений. Такова — основная идея, лежащая в основе свойства float в схемах размещения CSS, и она демонстрирует один из способов, который использован в табличном дизайне.

Эффективность использования float в многоколонной верстке была объяснена Douglas Bowman в 2004 в его классической презентации [No More Tables](http://www.stopdesign.com/present/2004/ddw-seattle/tables/):



Bowman объяснил принципы, лежащие в верстке без таблиц, используя старый сайт Microsoft в качестве образца. Float были заметно использованы в его ложной перестройке разметки Microsoft.

1. ***Синтаксис***

Свойство Float может принимать одно из 4 значений: влево (left), вправо(right), без выравнивания (none) и наследованное (inherit). Это объявляется, как показано в коде ниже:

[view source](https://webformyself.com/tajna-svojstva-float-css/#viewSource)[print](https://webformyself.com/tajna-svojstva-float-css/#printSource)[?](https://webformyself.com/tajna-svojstva-float-css/#about)

1.#sidebar {

2.float: left;

3.}

Наиболее часто используемые значения это влево и вправо. Значение none или первоначальное значение float для какого-либо элемента HTML-страницы являются значением по умолчанию. Значение inherit (наследовать), к которому может быть применено почти к каждому свойству CSS, не работает в версиях Internet Explorer, включая 7.

Свойство float не требует применения любого другого свойства для элемента с float, однако, чтобы функционировать правильно, float будет работать более эффективно при определенных обстоятельствах.

Как правило, плавающий элемент должен иметь **явно установленную ширину** (если это не [replaced element](http://reference.sitepoint.com/css/replacedelements), например изображение). Это гарантирует, что float ведет себя как ожидалось и помогает избежать проблем в некоторых браузерах.

[view source](https://webformyself.com/tajna-svojstva-float-css/#viewSource)[print](https://webformyself.com/tajna-svojstva-float-css/#printSource)[?](https://webformyself.com/tajna-svojstva-float-css/#about)

1.#sidebar {

2.float: left;

3.width: 350px;

4.}

1. ***Особенности плавающих элементов***

Ниже приводится список поведение плавающих элементов, согласно спецификации CSS2:

left-floated блок будет смещен влево до его левого края (или границы края, если край отсутствует) касается или края содержания блока, или края другого floated блока

Если размер floated блока превышает доступное горизонтальное пространство, то floated блок будет сдвинут вниз

Non-positioned, non-floated блочные элементы действуют, как плавающие элементы, т.е. находится вне потока документа

Края floated блока не совпадут с краями смежных блоков

Корневой элемент(<html>) не может быть floated

Встроенный элемент, тот который floated, преобразуется в элемент блочного типа

1. **Float на практике**

Наиболее часто используемым вариантом использования свойства float является плавающее изображение с текстом, обертывающим его. Например:

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper.

Aenean ultricies mi vitae est. Mauris placerat eleifend leo. Quisque sit amet est et sapien ullamcorper pharetra. Vestibulum erat wisi, condimentum sed, commodo vitae, ornare sit amet, wisi. Aenean fermentum, elit eget tincidunt condimentum, eros ipsum rutrum orci, sagittis tempus lacus enim ac dui.

Donec non enim in turpis pulvinar facilisis. Ut felis. Praesent dapibus, neque id cursus faucibus, tortor neque egestas augue, eu vulputate magna eros eu erat. Aliquam erat volutpat. Nam dui mi, tincidunt quis, accumsan porttitor, facilisis luctus, metus.

CSS, применяемая к изображению в поле выше выглядит следующим образом:

[view source](https://webformyself.com/tajna-svojstva-float-css/#viewSource)[print](https://webformyself.com/tajna-svojstva-float-css/#printSource)[?](https://webformyself.com/tajna-svojstva-float-css/#about)

1.img {

2.float: left;

3.margin: 0 15px 5px 0;

4.border: solid 1px #bbb;

5.}

Единственное свойство, которое может добиться такого эффекта работы это свойство float. Другие свойства (margin и border) есть там по эстетическим причинам. Другие элементы в блоке (тэги <p> с текстом в них) не нуждаются ни в каких стилях.

Как упомянуто ранее, floated элементы выбиваются из потока документа, а другие блочные элементы остаются в потоке, действуя, так если бы floated элемент там даже не находятся. Это продемонстрировано визуально ниже:

This box is floated left

This <p> element has a different background color to show that it spans the width of its parent, ignoring the floated element. This inline text, however, wraps around the floated box.

В приведенном выше примере<p>элемент уровня блока, таким образом он игнорирует floated элемент, охватывающий ширину контейнера (за вычетом padding). Все элементы не floated, блочного типа будут вести себя аналогично.

Текст в абзаце является встроенным, поэтому он обволакивает floated элемент. Floated блоку также предоставляются параметры margin, чтобы сместить его из абзаца, делая его визуально чистым, чтобы элемент абзаца игнорировал floated блок.

1. **Очистка floats**

Проблемы расположения с floats обычно решаются использованием CSS свойства clear, которое позволяет вам “убрать” floated элементы с левой или правой или с обеих сторон.

Давайте рассмотрим пример, который часто встречается — нижний колонтитул (footer) обтекает правую сторону 2-ух клоночной разметки:

Left column floated left

Right column also floated left

Footer

Если вы посмотрите на страницу IE6 и IE7, вы не увидите никаких проблем. Левые и правые столбцы на месте, и нижний колонтитул внизу. Но в Firefox, Опере, Safari и Chrome, вы увидеть, что нижний колонтитул (footer) съезжает со своего места. Это вызвано приминением float к столбцам. Это правильное поведение, хотя оно является более проблемным. Чтобы решить эту проблему, мы используем вышеупомянутое clear свойство, относительно нижнего колонтитула:

[view source](https://webformyself.com/tajna-svojstva-float-css/#viewSource)[print](https://webformyself.com/tajna-svojstva-float-css/#printSource)[?](https://webformyself.com/tajna-svojstva-float-css/#about)

1.#footer {

2.clear: both;

3.}

Результат показан ниже:

Left column floated left

Right column also floated left

Footer

The clear property will clear only floated elements, so applying clear: both to both columns would not cause the footer to drop down, because the footer is not a floated element.

Свойство clear очистит только floated элементы. Применение clear таково: оба столбца не будут опускать нижний колонтитул, потому что он не floated элемент.

Поэтому пользуйтесь clear на не- floated элементах, и даже иногда на floated элементах, чтобы заставить элементы страницы занять свои отведенные места.

1. **Фиксация Выпадания родителя**

Один из самых общих признаков разметки с float, является “выпадение — родителя”. Это продемонстрировано в примере ниже:

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper.

Обратите внимание, что основание floated изображения появляется вне своего родителя. Родитель полностью не расширяется, для содержания floated изображение. Это вызвано тем, что мы обсуждали ранее: floated элемент вне естественного потока документа, хотя все элементы блока отображаются, но floated элемента там как бы нет. Это не ошибка CSS, всё в соответствии со спецификациями CSS. Все браузеры делают то же в этом примере. Нужно сказать, что, в этом примере, добавляя ширину контейнера можно предотвратить проблему в IE, но нужно рештиь проблему и для Firefox, Opera, Safari, или Chrome.

1. ***Решение 1:Float для контейнера***

Самый простой способ решения этой проблемы является float для родительского элемента:

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper.

Теперь контейнер расширяется, чтобы уместить все дочерние элементы. Но к сожалению, это исправление будет работать только в ограниченном числе случаев, поскольку floating родитель может повлечь нежелательные последствия для вашего макета.

1. ***Решение 2: Добавление Дополнительной Разметки***

Это — устаревший метод, но это простой вариант. Просто добавьте дополнительный элемент в нижнюю часть контейнера и «очистите» его. Вот как HTML-код будет выглядеть после реализации этого метода:

[view source](https://webformyself.com/tajna-svojstva-float-css/#viewSource)[print](https://webformyself.com/tajna-svojstva-float-css/#printSource)[?](https://webformyself.com/tajna-svojstva-float-css/#about)

1.<div id="container">

2.<img src="//media.smashingmagazine.com/cdn\_smash/wp-content/uploads/2009/10/lifesaver.jpg" width="200" height="222" alt="" />

3.<p>Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper.</p>

4.<div class="clearfix"></div>

5.</div>

И в результате CSS применяется к новым элементам:

[view source](https://webformyself.com/tajna-svojstva-float-css/#viewSource)[print](https://webformyself.com/tajna-svojstva-float-css/#printSource)[?](https://webformyself.com/tajna-svojstva-float-css/#about)

1..clearfix {

2.clear: both;

3.}

Вы также можете сделать это посредством <br/> тэг с действующим стилем. В любом случае, вы получите желаемый результат: родительский контейнер расширится, чтобы включить все его дочерние элементы. Но этот метод не рекомендуется, так как он добавляет несемантический код к Вашей разметке.

1. ***Решение 3: Псевдо-элемент after***

:after псевдо-элемент добавляет некий элемент к выполненной странице HTML. Этот метод использовался весьма широко, чтобы решить проблемы float-clearing. Вот как выглядит CSS:

[view source](https://webformyself.com/tajna-svojstva-float-css/" \l "viewSource" \o "view source)[print](https://webformyself.com/tajna-svojstva-float-css/#printSource)[?](https://webformyself.com/tajna-svojstva-float-css/#about)

1..clearfix:after {

2.content: ".";

3.display: block;

4.height: 0;

5.clear: both;

6.visibility: hidden;

7.}

Класс CSS “clearfix” применяем к любому контейнеру, который имеет дочерние элементы с float и не расширяется, чтобы включить их.



Практический курс по верстке адаптивного лендинга с нуля!

Научитесь с нуля верстать адаптивные лендинги на HTML5 и CSS3 за ближайшие 6 дней

[Узнать подробнее](https://webformyself.com/minikurs/html5/?utm_medium=systema&utm_source=adsinsingle2&utm_campaign=html5_new)

Но этот метод не работает для Internet Explorer до7 версии, поэтому для IE нужно применить один из следующих вариантов:

[view source](https://webformyself.com/tajna-svojstva-float-css/#viewSource)[print](https://webformyself.com/tajna-svojstva-float-css/#printSource)[?](https://webformyself.com/tajna-svojstva-float-css/#about)

1..clearfix {

2.display: inline-block;

3.}

4.

5..clearfix {

6.zoom: 1;

7.}

В зависимости от типа проблемы вы имеете дело с одним из двух решений, которые разрешат эту проблему в Internet Explorer. Следует отметить, что свойство zoom не является стандартом свойств Microsoft и поэтому ваш CSS станет не валидным.

Так, как псевдо — элемент :after не работает в IE6/7, получаем немного раздутый и мудрёный код, и требуется дополнительный недопустимый только для IE стиль, поэтому это решение не является лучшим способом, но вероятно, лучшее, из тех, что мы рассматривали до сих пор.

1. ***Решение 4: Свойство Overflow***

Безусловно, лучший и самый простой способ решить проблему с выпадением родителя добавить overflow: hidden или overflow: auto к родительскому элементу. Это ясно, легко поддерживается, работает почти во всех браузерах и не добавляет лишней разметки.

Заметьте, что я сказал «почти» во всех браузерах. Это объясняется тем, что он не работает в IE6. Но, во многих случаях, у родительского контейнера будет установленная ширина, которая устраняет проблему в IE6. Если у родительского контейнера нет ширины, Вы можете добавить IE6-единственный стиль со следующим кодом:

[view source](https://webformyself.com/tajna-svojstva-float-css/#viewSource)[print](https://webformyself.com/tajna-svojstva-float-css/#printSource)[?](https://webformyself.com/tajna-svojstva-float-css/#about)

1.// This fix is for IE6 only

2..clearfix {

3.height: 1%;

4.overflow: visible;

5.}

В IE6, свойство height неправильно рассматривать как min-height, таким образом, это вынуждает контейнер включить свои дочерние элементы. Свойство Overflow затем устанавливается обратно в “visible”, чтобы гарантировать, что содержимое не скрыто или пролистано.

Единственный недостаток в использовании overflow метода (в любом браузере) является возможность того, что содержание элемента будет иметь полосы прокрутки или скроет информационное наполнение. Если есть какие-нибудь элементы с отрицательными margin или с абсолютным позиционированием в родителе, то они будут скрыты, если они выходят за пределы родителя, поэтому этот метод следует использовать осторожно. Нужно также отметить что, если бы у содержащегося элемента была бы указанная height или min-height, то Вы определенно не смогли бы использовать метод overflow.

Таким образом, на самом деле не просто кроссбраузерно решить данную проблему. Но практически любая проблема очистки float может быть решена одним из вышеупомянутых методов.

1. **Связанные float ошибки в Internet Explorer**

За эти годы имели место многочисленные статьи, опубликованные в сети, обсуждающие общие ошибки, связанные с float в CSS разметке. Все они, что не удивительно, имеют дело с проблемами характерными для Internet Explorer. Ниже, Вы найдете список ссылок на ряд статей, которые обсуждают связанные с float проблемы:

[The Internet Explorer Guillotine Bug](http://css-class.com/articles/explorer/guillotine/index.htm)

[The IE5/6 Doubled Float-Margin Bug](http://www.positioniseverything.net/explorer/doubled-margin.html)

[IE7 Bottom Margin Bug](http://www.maratz.com/blog/archives/2006/11/11/ie-7-quirks-floats-and-margins/)

[The IE Escaping Floats Bug](http://www.positioniseverything.net/explorer/escape-floats.html)

[The IE6 Peekaboo Bug](http://www.positioniseverything.net/explorer/peekaboo.html)

[The IE6 “Ghost Text” Bug](http://www.impressivewebs.com/ie6-ghost-text-bug-with-multiple-solutions/)

[The IE6 Expanding Box Problem](http://www.positioniseverything.net/explorer/expandingboxbug.html)

[The IE6 3-pixel Gap](http://www.positioniseverything.net/explorer/threepxtest.html)

1. **Изменение cвойства float при помощи JavaScript**

Чтобы изменить значение CSS в JavaScript, вы должны получить доступ к стилю объекта, преобразовывая намеченное свойство CSS в «Camel case” Например, свойство CSS “margin-left” становится marginLeft, свойство background-color становится BackgroundColor, и так далее. Но со свойством float, всё по другому, потому что слово float уже зарезервированно в JavaScript. Поэтому следующее будет неправильным:

[view source](https://webformyself.com/tajna-svojstva-float-css/#viewSource)[print](https://webformyself.com/tajna-svojstva-float-css/#printSource)[?](https://webformyself.com/tajna-svojstva-float-css/#about)

1.myDiv.style.float = "left";

Вместо этого, вы должны использовать одно из следующих действий:

[view source](https://webformyself.com/tajna-svojstva-float-css/#viewSource)[print](https://webformyself.com/tajna-svojstva-float-css/#printSource)[?](https://webformyself.com/tajna-svojstva-float-css/#about)

1.// For Internet Explorer

2.myDiv.style.styleFloat = "left";

3.

4.// For all other browsers

5.myDiv.style.cssFloat = "left";

1. **Практическое использование Float**

Floats могут быть использованы для решения целого ряда задач в CSS разметке. Некоторые примеры описаны здесь.