Контейнер располагается в соответствии со одной из схем позиционирования:

1. *Нормальный поток* (normal flow, далее *поток*), включающий в себя форматирование блока, инлайн-форматирование и относительное позиционирование.x` x
2. *Поплавки* (float). В этом случае, контейнер изначально позиционируется в соответствии с потоком, а затем сдвигается вправо или влево насколько возможно. (Свойство float превращает элемент в плавающий, при этом он прижимается к левому или правому краю родителя, а текст его обходит с других сторон. Такое поведение текста напоминает поток воды, обтекающий камень, поэтому элементы с таким поведением и называются плавающими. )
3. *Абсолютное позиционирование* (absolute positioning). При таком подходе контейнер полностью изымается из потока (т.е. не оказывает никакого влияния на соседние контейнеры) и располагается относительно содержащего контейнера.

Для вёрстки страниц часто используются два основных инструмента — позиционирование (positioning) и свободное перемещение (floating). CSS позиционирование позволяет указать, где появится блок элемента, а свободное перемещение перемещает элементы к левому или правому краю блока-контейнера, позволяя остальному содержимому «обтекать» его.

Свойства элемента с float, которые нужно всегда держать в голове

если оно установлено в значение left или right

* Элемент отображается **как блочный**, так словно ему установлено свойство display: block;
* Элемент **по ширине сжимается** до размеров содержимого, если для элемента явно не установлена ширина width;
* Элемент **прилипает** к левому (left) или правому краю (right);
* **Все остальное** содержимое страницы, идущее в HTML коде **после** элемента с float, обтекает его;

Если display равно none, то float и position должны быть проигнорированы. При этом элемент не сгенерирует контейнера.

* **В случае абсолютного расположения контейнера (position равно absolute или fixed), свойство display устанавливается в block, а float в none, независимо от того, что кто-либо написал в css, но в случае расположения контейнера, как поплавка (float не равно none), свойство display устанавливается вblock,независимо от того, что кто-либо написал в css.**
* Блоки с float по умолчанию не влияют на высоту родителя, то есть если у вас есть некоторый контейнер, а в нем находятся только плавающие блоки, то высота контейнера станет равна нулю.
* Свойство автоматически изменяет вычисляемое (отображаемое в браузере) значение свойства display на display: block для следующих значений: inline,inline-block, table-row, table-row-group, table-column, table-column-group, table-cell, table-caption, table-header-group, table-footer-group. Значение inline-tableменяет на table.
* Плавающий блок принимает размеры своего содержимого с учетом внутренних отступов и рамок.
* При этом остальные блочные элементы будут его(float) игнорировать, а строчные элементы будут смещаться вправо или влево, освобождая для него пространство и обтекая его.
* Верхние и нижние отступы margin плавающих элементов не схлопываются.
* Свойство float применяется как к блочным элементам, так и к строчным элементам.
* Левый или правый внешний край перемещаемого элемента, в отличие от позиционированных элементов, не может располагаться левее (или правее) внутреннего края его блока-контейнера, т.е. выходить за его границы. При этом, если для блока-контейнера заданы внутренние отступы, то плавающий блок будет отстоят от края блока-контейнера на заданное расстояние.
* Свойство не оказывает никакого влияния на элементы с display: flex и display: inline-flex.
* При использовании свойства float для блочных элементов обязательно задавать ширину.
* Свойство float лишь частично влияет на элемент в потоке. Можно управлять положением по горизонтали, меняя значение float с right на left, но по вертикали положение элемента задаётся его местом в коде.

Плавающие элементы достаточно активно применяются при вёрстке веб-страниц и служат для реализации этих и не только задач:

* обтекание картинок текстом;
* создание врезок; (иллюстрации в тексте)
* горизонтальные меню;
* колонки.
* Многоколоночные макеты
* Галереи фотографий

Все эти особенности плавающих элементов могут оказать довольно неприятное воздействие на макет веб-страницы, особенно в случаях подобным перечисленным выше. Основной способ добиться желаемого результата — это в нужный момент отменить обтекание. Для этого есть несколько способов.

1) Если плавающий элемент будет занимать всю доступную ширину, то остальные элементы, следующие за ним, будут начинаться с новой строки. Для этого надо включить свойство width со значением 100%.

2) Свойство overflow управляет отображением содержания блочного элемента, если оно целиком не помещается и выходит за область заданных размеров. Значение auto в частности, добавляет полосы прокрутки при необходимости, а hiddenотображает только область внутри элемента, остальное скрывает. Кроме этого, использование overflow со значениемauto, scroll или hidden отменяет действие float.

3) Свойство clear определяет, как будет располагаться элемент, идущий следом за плавающим элементом. 1) Чтобы отменить действие обтекания, свойство clear надо добавлять к элементу, идущему после плавающего. Обычно вводят универсальный класс, к примеру, clear и вставляют пустой тег **<div>** с этим классом

2)добавляем плавающие элементы в контейнер, и этому контейнеру прописываем class clearfix, которому через псевдоэлемент after: прописываем обязательный атрибут псевдоэлемента after content’ .‘; в котором пишем любой контент ( у нас точка), потом прописываем display block; height 0 visibility:hidden; и в конце clear both; Но кажется можно просто после content’ .‘; прописать clear both;

**Position**

Свойство position позволяет точно задать новое местоположение блока относительно того места, где он находился бы в нормальном потоке документа. По умолчанию все элементы располагаются последовательно один за другим в том порядке, в котором они определены в структуре html-документа. Свойство не наследуется.

Различают четыре типа позиционирования: нормальное, абсолютное, фиксированное и относительное.

Благодаря комбинации свойств position, left, top, right и bottom элемент можно накладывать один на другой, выводить в точке с определёнными координатами, фиксировать в указанном месте, определить положение одного элемента относительно другого и др.

**static:** Значение по умолчанию, означает отсутствие позиционирования. Элементы отображаются последовательно один за другим в том порядке, в котором они определены в html-документе. Свойства left, top, right, bottom если определены, игнорируются.

**Absolut:** При абсолютном позиционировании элемент не существует в потоке документа и его положение задаётся относительно краёв браузера.

Для режима характерны следующие особенности.

* Ширина слоя, если она не задана явно, равна ширине контента плюс значения полей, границ и отступов.
* Слой не меняет своё исходное положение, если у него нет свойств right, left, top и bottom.
* Свойства left и top имеют более высокий приоритет по сравнению с right и bottom. Если left и right противоречат друг другу, то значение right игнорируется. То же самое касается и bottom.
* Если left задать отрицательное значение, то слой уйдёт за левый край браузера, полосы прокрутки при этом не возникнет. Это один из способов спрятать элемент от просмотра. То же относится и к свойству top, только слой уйдёт за верхний край.
* Если left задать значение больше ширины видимой области или указать right с отрицательным значением, появится горизонтальная полоса прокрутки. Подобное правило работает и с top, только речь пойдёт о вертикальной полосе прокрутки.
* Одновременно указанные свойства left и right формируют ширину слоя, но только если width не указано. Стоит добавить свойство width и значение right будет проигнорировано. Аналогично произойдёт и с высотой слоя, только уже участвуют свойства top, bottom и height.
* Элемент с абсолютным позиционированием перемещается вместе с документом при его прокрутке.
* Блок-контейнер для абсолютно позиционированного элемента — ближайший элемент-предок, значение свойства position которого не равно static.
* Пространство, которое занимал такой элемент, схлопывается, как будто элемента не существовало на странице.
* Абсолютно позиционированный элемент может перекрывать другие элементы или быть перекрытым ими (за счёт свойства z-index).
* Любой абсолютно позиционированный элемент генерирует блок, то есть принимает значение display: block;.
* Кроме абсолютного позиционирования для элементов необходимо назначить свойство overflow со значением auto. Тогда при превышении контентом высоты видимой области появится полоса прокрутки.
* Если position равно absolute или fixed (подвид absolute), то display устанавливается в block, а float — в none (таким образом, указанный position отменяет float). Расположение блока в этом случае описывается свойствами top, right, bottom, left и содержащим контейнером.
* Если display равно none, то float и position должны быть проигнорированы. При этом элемент не сгенерирует контейнера.

**relativ:** Если задать значение relative свойства position, то положение элемента устанавливается относительно его исходного места. Добавление свойств left, top, right и bottom изменяет позицию элемента и сдвигает его в ту или иную сторону от первоначального расположения.

Для относительного позиционирования характерны следующие особенности.

* Этот тип позиционирования не применим к элементам таблицы вроде ячеек, строк, колонок и др.
* При смещении элемента относительно исходного положения, место, которое занимал элемент, остаётся пустым и не заполняется ниже или вышележащими элементами.
* Если для относительно позиционированного элемента одновременно задать свойства top и bottom или left и right, то в первом случае сработает только top, во втором — left.

**Fixed:** Фиксирует элемент в нужном месте страницы. Блоком-контейнером фиксированного элемента является окно просмотра, при этом элемент полностью удаляется из потока документа.

Фиксированное положение слоя задаётся значением fixed свойства position и по своему действию похоже на абсолютное позиционирование. Но в отличие от него привязывается к указанной свойствами left, top, right и bottom точке на экране и не меняет своего положения при прокрутке веб-страницы. Ещё одна разница от absolute заключается в том, что при выходе фиксированного слоя за пределы видимой области справа или снизу от неё, не возникает полос прокрутки.

Применяется такой тип позиционирования для создания меню, вкладок, заголовков, в общем, любых элементов, которые должны быть закреплены на странице и всегда видны посетителю.

### 4. Проблемы позиционирования

1. Если ширине или высоте абсолютно позиционированного элемента присвоено значение auto, то её значение будет определяться шириной или высотой содержимого элемента. Если ширина или высота объявлена явно, то именно это значение и будет присвоено.  
2. Если внутри блока с position: absolute расположены элементы, для которых задано обтекание float, то высота этого элемента будет равна высоте самого высокого из этих элементов.  
3. Для элемента с position: absolute нельзя одновременно устанавливать свойствоfloat, а для элемента с position: relative — можно.  
4. Если предок позиционированного элемента является блочным элементом, то блок-контейнер формируется областью содержимого, ограниченной рамкой (border). Если предок — строковый элемент, блок-контейнер формируется внешней границей его содержимого. Если предка нет, блоком-контейнером является элемент body.

**В случае абсолютного расположения контейнера (position равно absolute или fixed), свойство display устанавливается в block, а float в none, независимо от того, что кто-либо написал в css.**  
  
**В случае расположения контейнера, как поплавка (float не равно none), свойство display устанавливается в block, независимо от того, что кто-либо написал в css.**