**Позиционирование элементов**

•Все браузеры в ходе разбора html-кода и формирования веб-страницы располагают элементы по установленному стандарту, согласно которому блочные элементы располагаются вертикально друг под другом, т.к. по умолчанию они стремятся занять всю доступную в данной ситуации ширину, а строчные элементы наоборот, располагаются горизонтально друг за другом до тех пор, пока не будет занята вся доступная в данной ситуации ширина, после чего браузер делает перенос строки и продолжает располагать строчные элементы друг за другом на новой строке. Однако такой способ расположения элементов на странице далеко не всегда удобен, поэтому в CSS присутствует специальное ненаследуемое свойство **position**, которое позволяет определять порядок расположения элементов относительно окна браузера, других элементов на веб-странице или относительно текущего местоположения элемента.

Данное свойство используется в комбинации со свойствами **left**, **right**, **top** и **bottom**, которые определяют относительное смещение позиционированного элемента. В качестве значений эти свойства принимают любые доступные в CSS единицы измерения, включая отрицательные значения. Разрешается использовать проценты, которые у свойств **left** и **right** рассчитываются относительно ширины родительского элемента, а у topи bottomотносительно высоты родительского элемента. Если указать в качестве значения свойств ключевое слово **auto**, положение позиционированного элемента не изменится.

В зависимости от значения свойства **Position** смещение элемента может быть неодинаково. При абсолютном позиционировании он смещается относительно границ родительского элемента, при относительном позиционировании, смещение происходит относительно нормального потока на заданные величины.

Теперь перечислим значения свойства position:

**static**–элемент будет отображаться в том месте, в котором он обычно располагается, находясь в общем потоке документа; применение к элементу свойств **left**, **right**, **top** и **bottom** не даст никакого эффекта;

**absolute**–элемент будет удален из общего потока документа, а другие элементы займут его место, как будто абсолютно позиционированный элемент вообще отсутствует; применение к элементу свойств **left, right, top** и **bottom** приведет к его смещению относительно соответствующей стороны позиционированного родителя (значение свойства у него не должно быть static) или, в случае отсутствия такового, относительно соответствующей стороны первого позиционированного элемента-предка; если же для данного элемента не найдется и позиционированного элемента-предка, то смещение будет происходить относительно соответствующей стороны окна браузера; .pos\_1{position: absolute;}

**fixed**–элемент будет удален из общего потока документа, а другие элементы займут его место, как будто фиксированно позиционированный элемент вообще отсутствует; применение к элементу свойств **left, right, top** и **bottom** приведет к его смещению относительно соответствующей стороны окна браузера, соответственно, при прокрутке содержимого страницы в окне браузера фиксированный элемент будет оставаться на одном и том же месте;

**relative**–элемент останется в общем потоке документа, а применение к элементу свойств **left, right, top** и **bottom** приведет к его смещению относительно текущего положения соответствующей стороны элемента, при этом его место не будет занято другими элементами из потока, т.к. для них относительно позиционированный элемент как будто остается на прежнем месте (в потоке документа).

**css-свойство z-index**

Введение позиционирования значительно облегчило жизнь веб-разработчикам, позволив размещать элементы практически в любом месте веб-документа по своему усмотрению. Однако при использовании позиционирования зачастую происходит наложение элементов друг на друга, что привело к идее введения дополнительного ненаследуемого свойства **z-index** для регулирования положения позиционированных элементов не только в горизонтальной плоскости вдоль осей х и у, но и по вертикали вдоль оси z, имитируя тем самым третье измерение, перпендикулярное экрану

z-index: 2;

В качестве значений свойство принимает целые числа (включая отрицательные). Чем больше значение z-индекса элемента, тем выше он находится по сравнению с элементами, у которых он меньше. Если же элементы имеют одинаковый z-индекс, то на переднем плане будет отображаться элемент, который в html-коде встречается ниже. В качестве значения разрешается также использовать ключевое слово auto, которое присваивает элементу z-индекс родителя.

**css-свойство clip**

Для абсолютно и фиксированнопозиционированных элементов имеется еще одно ненаследуемое свойство **clip**, определяющее прямоугольную область элемента, в которой содержимое будет видимым. Вся остальная часть обрезается и на странице не показывается

В качестве значений свойство **clip** принимает:

**rect**(y1,x1,y2,x2) –задает параметры прямоугольной области, в которой содержимое элемента будет видно на странице; через запятую указываются расстояния от края элемента до верхнего, правого, нижнего и левого края оставляемой области при этом расстояния y1 и y2 отсчитываются от верхнего края элемента, соответственно, до верхегои нижнего края оставляемой (видимой) области, а расстояния x1 и x2 отсчитываются от левого края элемента, соответственно, до правого и левого края оставляемой (видимой) области, так что при указании значений следует быть внимательным и следить за тем, чтобы во избежание полной обрезки элемента выполнялись условия y1<y2 и x1>x2; если с какой-то стороны обрезать элемент не требуется, то разрешается использовать ключевое слово auto; для указания размеров разрешается использовать все доступные в CSS единицы измерения;

**auto**–элемент обрезаться не будет; соответствует rect(auto,auto,auto,auto).

**css-свойство float**

Как мы уже говорили, браузер по умолчанию размещает блочные элементы вертикально друг под другом. При этом даже, если задать их ширину, место по горизонтали все равно уже останется свободным и не будет занято другими элементами, все они по-прежнему будут размещаться либо выше, либо ниже данного блочного элемента по вертикали. Поэтому, чтобы иметь возможность изменять такое поведение браузера, в CSS применяется ненаследуемое свойство float, которое делает элемент плавающим, смещая его к левому или правому краю родительского элемента

В качестве значений свойство float может принимать:

**left**–элемент будет выравниваться по левому краю, а все остальные элементы обтекать его по правой стороне;

**right**–элемент будет выравниваться по левому краю, а все остальные элементы обтекать его по правой стороне;none–элемент будет отображаться обычным образом, т.е. перестанет быть плавающим.

**css-свойство clear**

В CSS имеется возможность для блочных элементов сделать так, чтобы они не обтекали плавающие элементы, которые идут в html-коде выше. Для этого используется ненаследуемое свойство clear, которое может принимать значения, определяющие обтекание каких плавающих элементов осуществлять не нужно:

**left**–данный блочный элемент не будет обтекать плавающие элементы, которые находятся выше в html-коде и для которых применяется свойство float, имеющее значение left;

**right**–данный блочный элемент не будет обтекать плавающие элементы, которые находятся выше в html-коде и для которых применяется свойство float, имеющее значение right;

**both**–данный блочный элемент не будет обтекать плавающие элементы, которые находятся выше в html-коде и для которых применяется свойство float, имеющее значение leftили right;

**none**–элемент будет обтекать плавающие элементы как обычно, т.е. согласно значениям свойства float.