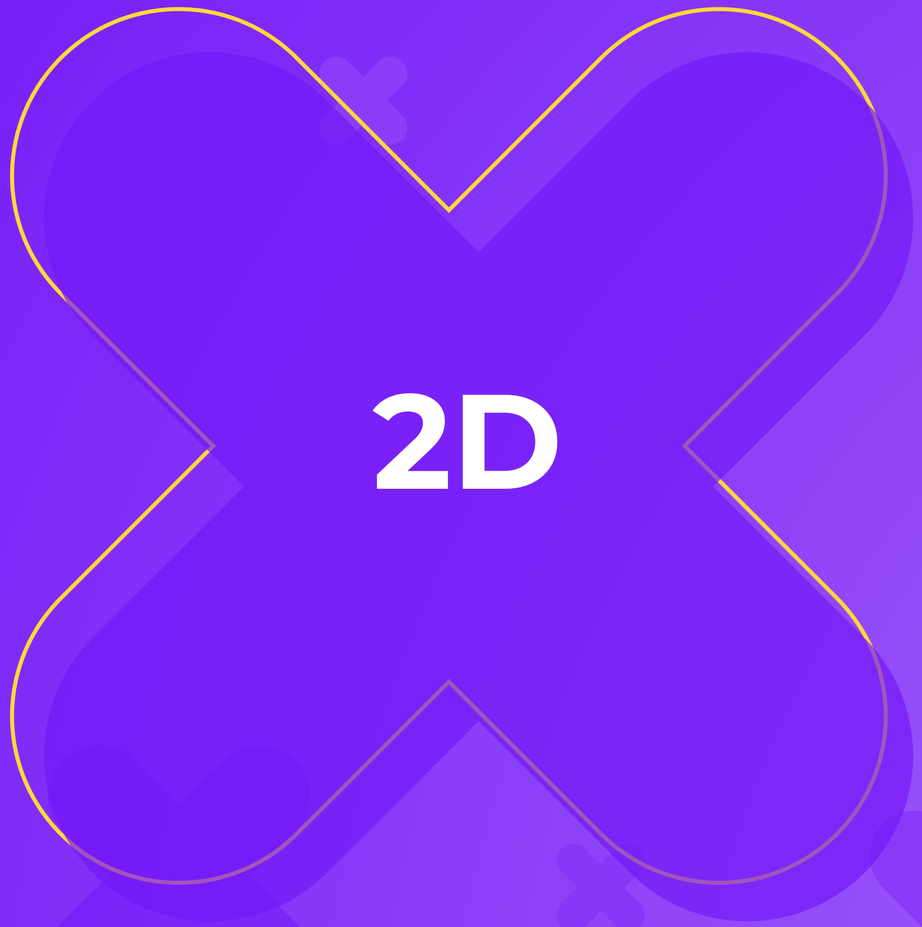
STEP ACADEMY

УРОК №22

Трансформация в CSS


2D

Трансформация

Отдельное свойство CSS, которое мы ещё не рассматривали. Трансформация преобразует существующий HTML элемент на основе одной или нескольких выбранных функций. Трансформации бывают 2D и 3D.

Для применения трансформации используется свойство transform.

Применение трансформаций не влияет на поток.

translate(x,y)

Сдвигает элемент на определенное расстояние по двум осям.

Значение задается в px и в %. Проценты рассчитываются по размерам блока.

Если указывать только одно значение, сдвиг будет по оси x.

Отрицательные значения сдвинут в обратную сторону

```
transform: translate(-50%, 20px)
```

translateX(), translateY()

Сдвиг только по осям X и Y соответственно.

Значения аналогичны translate

scale(x,y)

Масштабирование элементов по осям x и y.

Значения задаются в долях.

При указании только одного значения коэффициент масштабирования применяется ко всем осям целиком.

Доступны функции scaleX и scaleY для масштабирования только по одной оси.

Отрицательные значения отразят элемент зеркально.

```
transform: scale(-1.5, 0.5)
```

rotate (угол)

Поворачивает элемент на определенный угол по часовой стрелке. Угол задаётся в градусах, с помощью deg

Отрицательные значения повернут элемент против часовой стрелки.

```
transform: rotate(-20deg)
```

skew(x, y)

Наклоняет элемент на угол вдоль двух осей. Значения указываются в градусах (deg).

Указание одного значения наклонит элемент вдоль оси X.

skewX и skewY наклонят элемент вдоль конкретной оси.

```
transform: skew(-20deg, -20deg)
```

matrix(a, b, c, d, tx, ty)

Комбинирование всех способов.

matrix(scaleX(), skewY(), skewX(), scaleY(), translateX(), translateY())

Все значения задаются в долях, tx и ty. Для последних двух единицы измерения не указываются.

```
transform: matrix(1, 0, 0, 1, -100, 0);
```

transform-origin

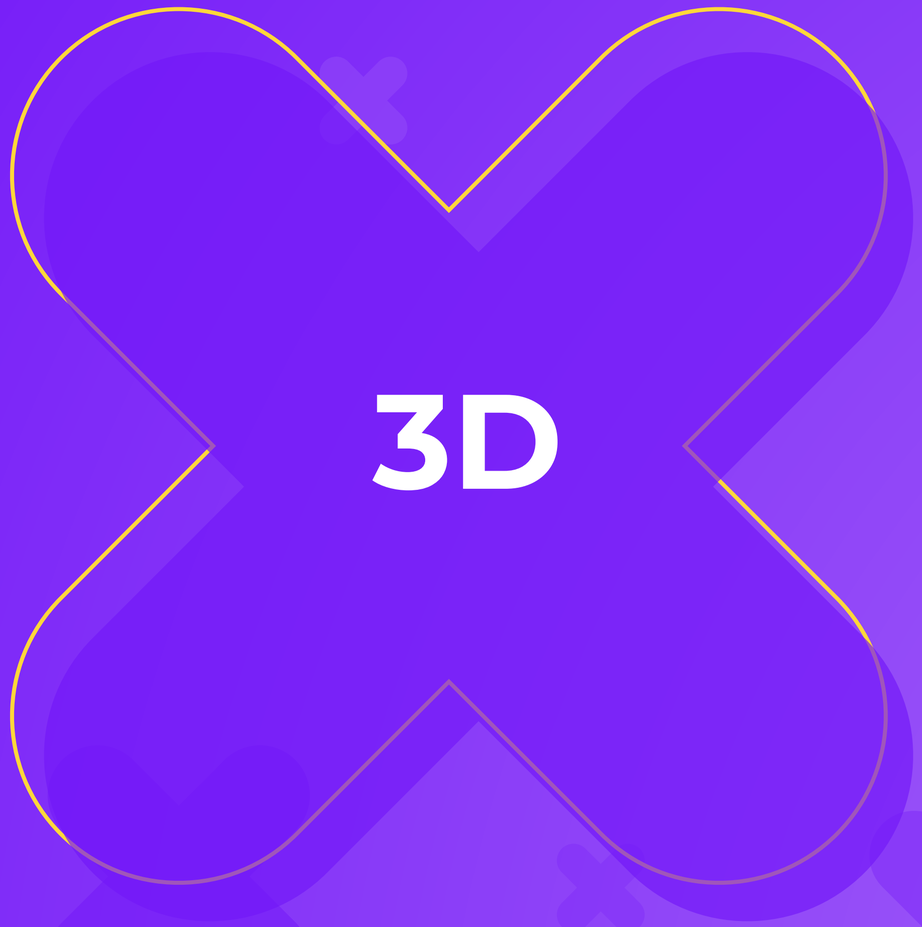
Изменение центра трансформации.

Задаётся значениями в процентах или пикселях (сначала по X, потом по Y), либо ключевыми словами top, bottom, left, right, center.

```
transform-origin: 200px 100px;
```

Множественные трансформации

Функции трансформации можно перечислять через пробел в порядке появления.


3D

transform-style

Свойство, которое указывается в какой плоскости будут происходить трансформации. Значение flat означает плоские, двумерные трансформации, а значение preserve-3d включит трехмерную трансформацию

```
transform-style: preserve-3d;
```

Свойство устанавливается для родительского элемента!

Важное замечание: некоторые из свойств (overflow и opacity, например) в значениях, отличных от значения по умолчанию, предотвращают трехмерную трансформацию.

perspective

Включает перспективу для дочерних элементов, добавляя ощущение 3-мерного пространства.

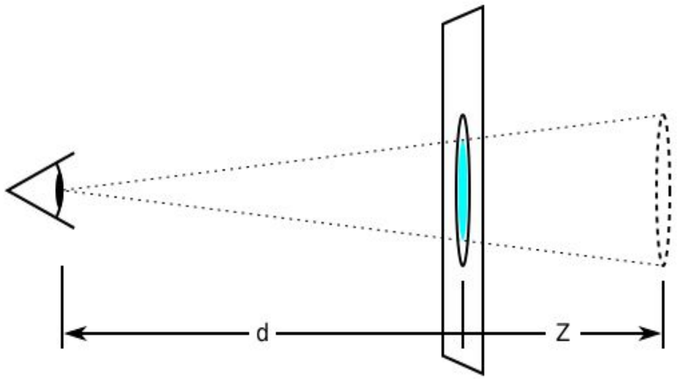
```
.container {
        perspective: 200px;
        transform-style: preserve-3d;
    }
```

Свойство устанавливается для родительского элемента!

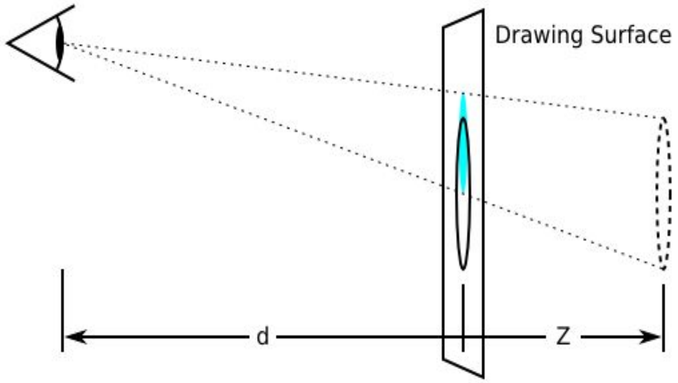
perpective-origin

Устанавливает точку перспективы. По умолчанию точка находится в центре, значения можно установить аналогичным образом, как и в background-position.

```
perspective-origin: 0% 100%;
```



original position of the viewer's eye is at the center



Effect of moving the position of the viewer's eye upwards using perspective-origin

d = distance from the viewer's eye to the screen
Z = position of the element on the z-axis

translate3d(x,y,z)

Сдвиг по 3 осям.

Существуют отдельные функции translateX, translateY, translateZ

scale3d(x,y,z)

Масштабирование по 3 осям.

Аналогично есть отдельные функции масштабирования по каждой оси scaleX, scaleY, scaleZ

rotate3d(x,y,z,угол)

Поворот на угол по вектору, заданному тремя точками. Точки задаются в долях.

Для поворота по отдельным осям есть функции rotateX, rotateY, rotateZ.

perspective(n)

Добавление перспективы для одного элемента.

backface-visibility

Отображения обратной части. По умолчанию отображается как зеркальное отражение поверхности, но со значением hidden обратная часть скрывается.

Конец

Давайте подведем итоги урока!
Чему мы научились? Что мы использовали?
К чему мы пришли?