Лабораторная работа № 7
ПРЕОБРАЗОВАНИЯ. ПЕРЕХОДЫ. АНИМАЦИЯ

Цель работы: изучить свойства преобразования, применение анимации к элементам, научиться использовать переходы.

Теоретические сведения для выполнения работы

Преобразования

К свойствам преобразования объекта относится вращение, масштабирование, смещение и наклон. Преобразование осуществляется с использованием свойства ***transform***. Свойство имеет следующие значения:

1. *scale*(0.5) — увеличить или уменьшить в размерах элемент (например, для значения **0.5** в два раза);
2. *rotate*(45deg) — повернуть элемент на определенный угол заданный в deg;
3. *translate*(40px, 60px) — смещение элемента из его текущей позиции на некоторое расстояние вправо на 40px и вниз на 60px;
4. *skew*(15deg) — наклоняет элемент, смещая верхний край элемента в одну сторону, а нижний — в противоположную.

По умолчанию точкой трансформации является центр элемента, но ее можно изменить с помощью свойства ***transform-origin***, например, ***transform-origin:*** 100% 0; устанавливает точку поворота в правом верхнем углу.

Вращения и перемещения могут производиться во всех трех измерениях — X, Y и Z. Значение *rotateZ* эквивалентно *rotate*, так как выполняет вращение вокруг оси Z. Значение *rotateX* и *rotateY* вращают элемент вокруг горизонтальной оси X (наклоняя элемент вперед или назад) и вертикальной оси Y (поворачивая элемент вправо и влево) соответственно.

Также можно задать значение перспективы используя трансформацию со значением *perspective()* или же свойство ***perspective***. В качестве значения указывается расстояние от плоскости монитора до точки сходимости линий. Чем меньше значение, тем более выраженной выглядит перспектива. По умолчанию перспектива отображается так, как если бы наблюдатель находился прямо по

центру элемента. Свойство ***perspective-origin*** смещает вправо или влево, вверх или вниз отображение перспективы.

Свойство ***transform-style*** позволяет располагать элементы в трехмерном пространстве, задавая значение *preserve-3d*.

Переходы

Переход представляет собой анимацию смены одного набора свойств CSS другим за определенный промежуток времени. Переходы задаются с помощью свойства ***transition***. На рисунке 7.1 представлен переход изменения цвета блока с синего на красный с использованием сокращенной записи свойства ***transition***.

```
.div:hover {
background-color: rgb(255,0,0);
transition: background-color 1s linear 0.5s;
}
.div {
background-color: rgb(0,0,255);
}
```

Рис. 7.1 Сокращенная запись свойства transition

В сокращенной записи первым указывается значение свойства ***transition-property***, которое определяет для каких свойств должен присутствовать переход. Значение *all* означает, что переход задается для всех свойств, которые изменяются. Вторым следует свойство ***transition-duration***, которое указывает продолжительность перехода к конечному значению, например, 0.5s.

Для замедления или ускорения перехода после значения продолжительности используется свойство *transition-timing-function* со значениями *linear*, *ease-in* и *ease-out*:

1. *linear* — переход изменяется с постоянной скоростью;
2. *ease-in* — изменение вначале протекает медленно, но ускоряется до самого конца перехода;
3. *ease-out* — изменение начинается быстро, но к концу перехода замедляется.
4. *cubic-bezier*(*x1, y1, x2, y2*) **—** график изменения перехода по кривой Безье

Для задержки перехода последним устанавливается значение свойства ***transition-delay***, например, 0.5s.

Анимация

Создание анимации проходит в два этапа:

1. Определение анимации, которое включает настройку ключевых кадров со списком анимируемых CSS-свойств.

2. Применение анимации к элементу.

Кадры определяются с помощью правила **@keyframes**. На рис. 7.2 представлен вариант создания анимации, состоящей из двух кадров.

```
@keyframes nameAnimation {
from {
/* здесь перечисляются свойства CSS */
}
to {
/* здесь перечисляются свойства CSS */
}
}
```

Рис. 7.2 Создание кадров анимации

В блоке объявления селектора элемента, к которому применяется анимация, может быть использована расширенная запись, представленная на рис. 7.3

```
.anime {
animation-name: nameAnimation;
animation-duration: 2s;
animation-timing-function: ease-in-out;
animation-delay: 5s;
animation-iteration-count: 2;
animation-direction: alternate;
animation-fill-mode: forwards; }
```

7.3 Расширенная запись свойства *animation*

Свойство ***animation-name*** задает имя анимации, определяемое правилом **@keyframes**, свойство ***animation-duration*** задает длительность анимации, свойство ***animation-timing-function*** задает временную функцию, описывающую ускорение и/или замедление воспроизведения анимации, свойство ***animation-iteration-count*** задает количество повторений анимации, свойство ***animation-direction*** задает обратное воспроизведение анимации при повторном воспроизведении.

Чтобы анимация была бесконечной необходимо использовать значение *infinite* для свойства ***animation-iteration-count***. Следует отметить, что для работы анимации необходимо задавать продолжительность и имя анимации.

Для того, чтобы использовать анимацию перемещения элемента, представленную на риунке 7.4, необходимо изначально для него задать свойство ***position*** со значением *absolute* или *relative*. Также рис. 7.4 представлена анимация из нескольких кадров с помощью разбиения ее на несколько кадров.

```
@keyframes around {
0% { left: 0; top: 0; }
25% { left: 200px; top: 0; }
50% { left: 200px; top: 200px; }
75% { left: 0; top: 200px; }
100% { left: 0; top: 0; }
}
p {
position: relative;
animation: around 4s linear infinite; }
```

Рис. 7.4 Создание анимации из нескольких кадров

Следует отметить, что для эффектов анимации можно применять свойство ***filter*** со следующими значениями:

1. *grayscale*(значение) — преобразует цвета в черно-белые и значение задается как в процентах (0% — 100%), так и в десятичных дробях (0–1).

2. *saturate*(значение) — изменяет насыщенность цвета.

3. *sepia*(значение) — создание эффекта сепии, т. е. тонирование в коричневый цвет.

4. *hue-rotate*(угол) изменяет цвета изображения в зависимости от заданного угла, который определяет на сколько изменится данный цвет в цветовом круге от красного до фиолетового.

5. *invert*(значение) инвертирует цвета, т. е. изменяет цвета на противоположные.

6. *opacity*(значение) определяет прозрачность элемента.

7. *brightness*(значение) изменяет яркость цвета.

8. *contrast*(значение) изменяет контрастность цвета.

9. *blur*(радиус) создает эффект размытости и значение указывается в пикселах (px).

Для расположения элементов друг над другом можно использовать для каждого из элементов свойство ***z-index***, значение которого определяет место расположения элемента.

Чтобы задать дополнительную внешнюю границу можно использовать свойство ***outline***. В отличие от ***border*** свойство ***outline*** не участвует в блочной модели CSS и не изменяет размер элемента. Поэтому его используют, когда хотят добавить рамку без изменения других CSS-параметров. Свойств outline-top, outline-left не существует. Все современные браузеры также поддерживают свойство ***outline-offset***, задающее отступ ***outline*** от внешней границы элемента.

Задания к лабораторной работе № 7

Создать новый HTML-документ animation.html с внешним подключением стилей, в котором должны быть элементы согласно заданиям, заключенные в ** **.

Задание 1 Создать четыре изображения, к которым необходимо добавить следующие преобразования при наведении на них:

Задание 1.1. Первое изображение масштабируется в сторону уменьшения в 2 раза;

Задание 1.2. Второе изображение повернуть на 45 градусов;

Задание 1.3. Третье изображение сместить вправо на 50px;

Задание 1.4. При наведении на четвертое изображение оно наклоняется на –20 градусов.

Задание 2 Расположите произвольные изображения согласно рис. 7.5. Значения перспективы выбрать произвольные.

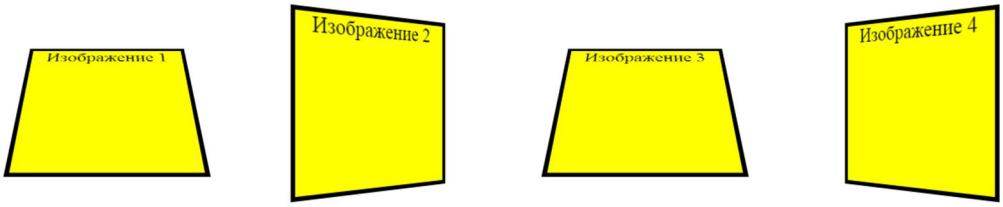


Рис. 7.5 Позиционирование изображений для задания 2

Задание 3 Создайте пять кнопок, при наведении на которые они должны изменяться согласно рис. 7.6. Для кнопок подобрать произвольное название.

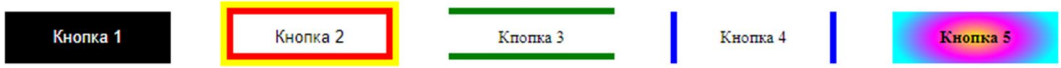
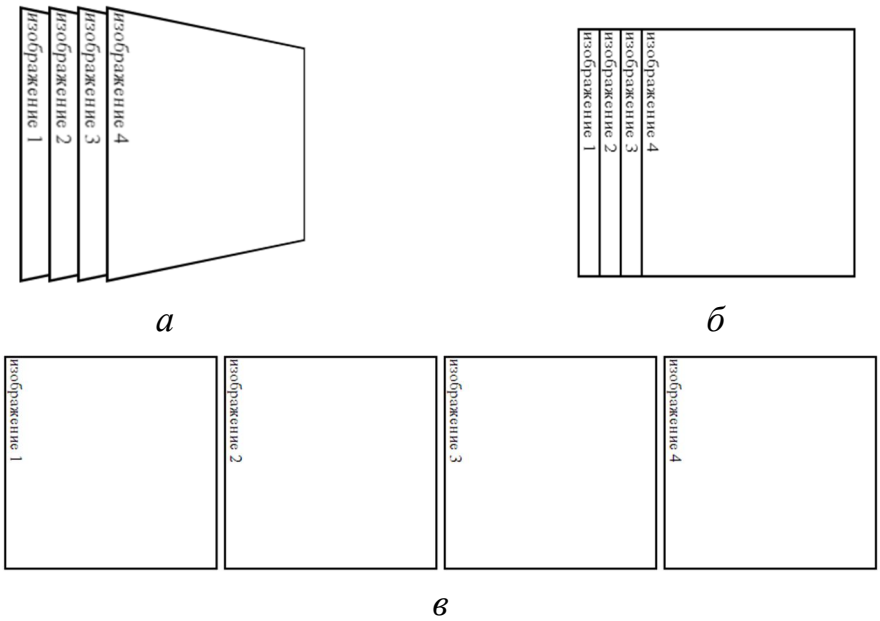


Рис. 7.6 Результат выполнения задания 4

Задания 4 Создайте анимацию 4 изображений, расположение которых соответствует кадрам, представленным на рисунку 7.7. В последнем кадре необходимо для второго изображения установить blur(10px), третьего изображения — grayscale(1), четвертого изображения — invert(1).



а — первый кадр, *б* — 50% кадр, *в* — последний кадр

Рис. 7.9 Результат положения изображений задания 6 в кадрах

Задание 5 Создайте анимацию непрерывной смены 5 изображений с продолжительностью 5 секунд с линейным ускорением. Изображения должны иметь общую границу произвольного цвета.

Задание 6 Создайте из изображения 5 копий и для них примените следующие эффекты:

Задание 6.1 Непрерывное перемещение 1 копии по горизонтали на 200px;

Задание 6.2 Непрерывное перемещение 2 копии по прямой с изменением геометрической формы с квадрата на круг;

Задание 6.3 Непрерывное перемещение 3 копии по прямой с появлением слева и со скрытием справа за экраном

Задание 6.4 Непрерывное перемещение 4 копии по вертикали вниз на 50px;

Задание 6.5. Непрерывное перемещение 5 копии по траектории квадрата, используя **left, top** для указания положения.

Задание 7. Создайте плавный переход цвета текста при наведении на него в блоке размером 300px на 300px, выбрать произвольные цвета, время переход установить в 5 секунд, переход сделать линейным. Блок должен быть максимально заполнен текстом

Задание 8. Для каждого задания необходимо добавить заголовок, к которому применить непрерывное изменение прозрачности от 0 до 80% с обратным воспроизведением.

Контрольные вопросы

1. С помощью какого свойства осуществляется трансформация?
2. Как осуществить наклон?
3. Как осуществить смещение?
4. Как осуществляется вращение?
5. Каким образом масштабировать элементы?
6. Что будет происходить, если использовать *rotateY*?
7. Что будет происходить, если использовать *rotateX*?
8. Как создать перспективу изображения?
9. Для чего используется свойство *transition*?
10. Что такое переходы?
11. Каким образом увеличить продолжительность перехода?
12. Для чего используется *transition-timing-function*?
13. Что входит в сокращенную запись свойства *transition*?
14. Назовите этапы создания анимации

15. Какие свойства включает сокращенная запись *animation*?

16. Каким образом сделать анимацию непрерывной?

17. Как изменить прозрачность элемента?

18. Каким образом создать несколько кадров анимации? Только 2 кадра?

19. Для чего необходимо свойство *z-index*?

20. Чем отличаются переходы от анимации?

21. Как остановить анимацию?

22. Для чего можно применять свойство *filter*?

23. Создайте плавный переход изменения цвета круга с синего на красный при наведении.