1. С помощью какого свойства осуществляется трансформация?

transform

2. Как осуществить наклон?

rotate(45deg) — повернуть элемент на определенный угол заданный в deg;

3. Как осуществить смещение?

translate(40px, 60px) — смещение элемента из его текущей позиции на некоторое расстояние вправо на 40px и вниз на 60px;

4. Как осуществляется вращение?

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

5. Каким образом масштабировать элементы?

. scale(0.5) — увеличить или уменьшить в размерах элемент (например, для значения 0.5 в два раза)

6. Что будет происходить, если использовать rotateY?

7. Что будет происходить, если использовать rotateX?

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) соответственн

8. Как создать перспективу изображения?

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.

9. Для чего используется свойство transition?

Переходы задаются с помощью свойства transition.

10. Что такое переходы?

Переход представляет собой анимацию смены одного набора свойств CSS другим за определенный промежуток времени.

11. Каким образом увеличить продолжительность перехода?

Для замедления или ускорения перехода после значения продолжительности используется свойство transition-timingfunction 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.

12. Для чего используется transition-timing-function?

13. Что входит в сокращенную 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-timingfunction 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

14. Назовите этапы создания анимации

Создание анимации проходит в два этапа: 1. Определение анимации, которое включает настройку ключевых кадров со списком анимируемых CSS-свойств. 2. Применение анимации к элементу.

15. Какие свойства включает сокращенная запись animation?

animation: around 4s linear infinite; }

16. Каким образом сделать анимацию непрерывной?

Чтобы анимация была бесконечной необходимо использовать значение infinite для свойства animation-iterationcount.

17. Как изменить прозрачность элемента?

свойство filter с opacity(значение) определяет прозрачность элемента;

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

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

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

19. Для чего необходимо свойство z-index?

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

20. Чем отличаются переходы от анимации?

Переход представляет собой анимацию смены одного набора свойств CSS другим за определенный промежуток времени.

21. Как остановить анимацию?

animation-fill-mode: forwards;

22. Для чего можно применять свойство filter

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. e. 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) cоздает эффект размытости и значение указывается в пикселах (px)