**Контрольные вопросы**

1. С помощью какого свойства осуществляет трансформация?

*transform*

функции трансформации, которые смещают элемент относительно его текущего положения на странице или изменяют его первоначальные размеры и форму.

2. Как осуществить наклон?

**skew(15deg)**

3. Как осуществить смещение?

**translate(40px, 60px)**

4. Как осуществляется вращение?

**rotate(45deg)**

5. Каким образом масштабировать элементы?

**scale(0.5)**

6. . Что будет происходить, если использовать *rotateY*?

Значение **rotateY()** — поворачивает элемент на определенное количество градусов вокруг оси Y

7. Что будет происходить, если использовать *rotateX*?

Значение **rotateX()** — поворачивает элемент на определенное количество градусов вокруг оси X

8. Как создать перспективу изображения?

Perspective показывает насколько обьект удален на экране.

можно задать значение перспективы используя

трансформацию со значением ***perspective()*** или же свойство

*perspective*.

transform: rotate(30deg) perspective(300px)

9. Для чего используется свойство *transition*?

Переходы задаются с помощью свойств *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?

Свойство *transition-duration* указывает продолжительность перехода к конечному значению, например, 0.5s.

Для задержки перехода используется свойство *transition-delay* со значением, например, 0.5s

12. Для чего используется *transition-timing-function*?

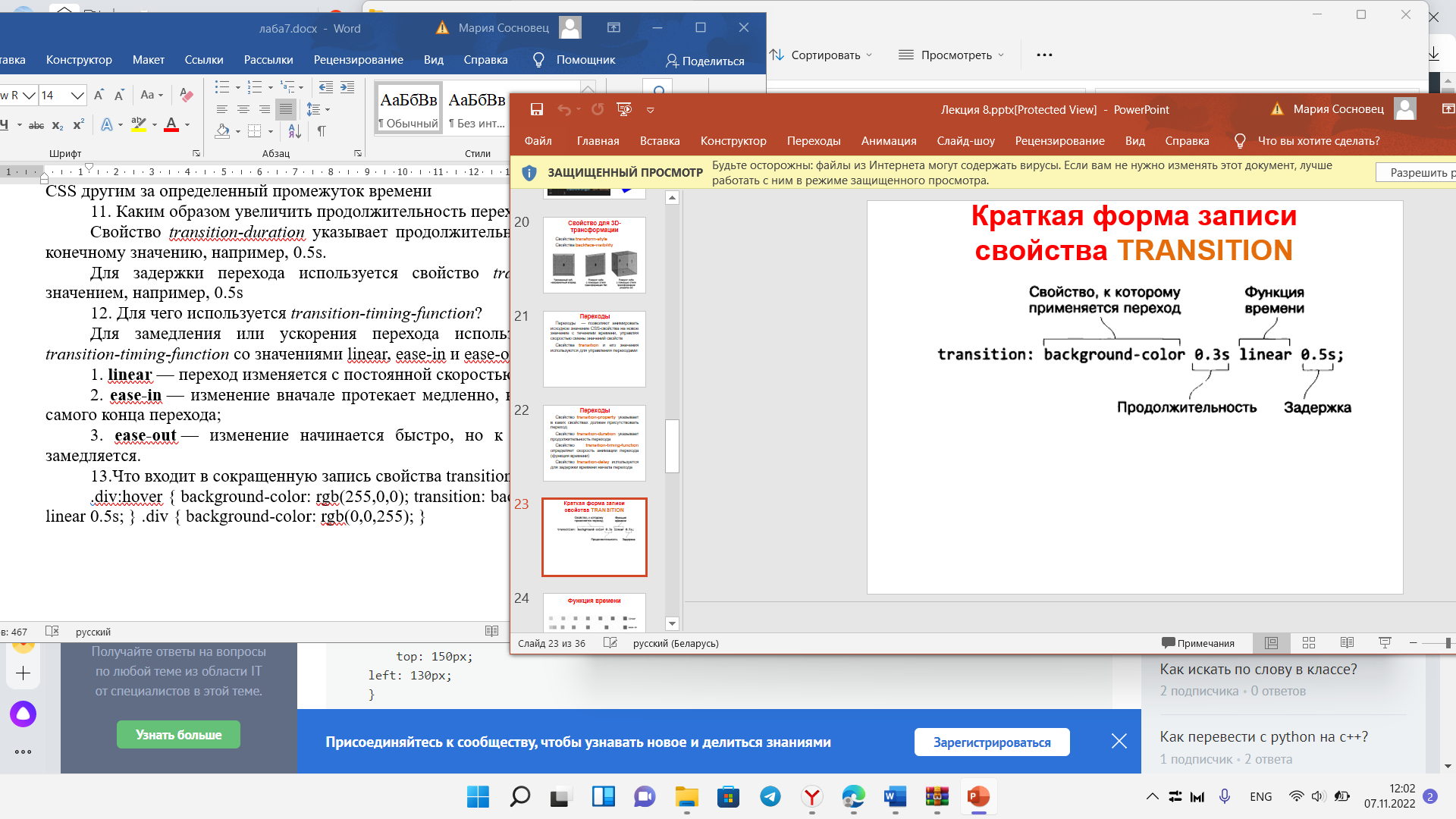
Для замедления или ускорения перехода используется свойство *transition-timing-function* со значениями linear, ease-in и ease-out:

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.

13.Что входит в сокращенную запись свойства transition?



14. Назовите этапы создания анимации

Создание анимации проходит в два этапа.

1. Определение анимации. Включает настройку ключевых кадров со списком анимируемых CSS-свойств.

2. Применение анимации к элементу.

Кадры определяются с помощью правила @keyframes

15. Какие свойства включает сокращенная запись animation?

div {

animation-name: coolAnimation; время задержки перед стартом анимации

animation-duration: 3s; сколько раз будет проигрываться анимационный цикл, перед тем как остановиться

animation-timing-function: ease-in; функция времени

animation-delay: 500ms; время задержки перед стартом анимации

animation-iteration-count: 6; сколько раз будет проигрываться анимационный цикл, перед тем как остановиться

animation-direction: reverse; анимация воспроизводиться вперёд, назад или переменно вперёд и назад

animation-fill-mode: forwards; применение стиля к объекту анимации до и после её выполнения

animation-play-state: paused; состояние анимации, паузы или проигрыша

}

/\* Стиль выше эквивалентен следующему: \*/

div {

animation: coolAnimation 3s ease-in 500ms 6 reverse forwards paused;

}

Animation: имя, длительность, временная функция, **время задержки перед стартом анимации,**  **сколько раз будет проигрываться анимационный цикл, перед тем как остановиться, движется анимация вперед/назад/переменно вперед/переменно назад, применение стили к объекту анимации до и после её выполнения,**

**состояние анимации, паузы или проигрыша.**

|  |  |
| --- | --- |
| 1 | @keyframes name{ |

|  |  |
| --- | --- |
| 2 | from{значение} |

|  |  |
| --- | --- |
| 3 | to{значение} |

|  |  |
| --- | --- |
| 4 | } |

name — имя анимации;  
from — начальное состояние элемента;  
to — конечное состояние элемента.

Но можно и по-другому:

|  |  |
| --- | --- |
| 1 | @keyframes name{ |

|  |  |
| --- | --- |
| 2 | from {значение} |

|  |  |
| --- | --- |
| 3 | 50% {значение} |

|  |  |
| --- | --- |
| 4 | to {значение} |

|  |  |
| --- | --- |
| 5 | } |

или так:

|  |  |
| --- | --- |
| 1 | @keyframes name{ |

|  |  |
| --- | --- |
| 2 | 0% {значение} |

|  |  |
| --- | --- |
| 3 | 20% {значение} |

|  |  |
| --- | --- |
| 4 | 40% {значение} |

|  |  |
| --- | --- |
| 5 | 60% {значение} |

|  |  |
| --- | --- |
| 6 | 80% {значение} |

|  |  |
| --- | --- |
| 7 | 100% {значение} |

|  |  |
| --- | --- |
| 8 | } |

16. Каким образом сделать анимацию непрерывной?

animation-iteration-count: infinite

17. Как изменить прозрачность элемента?

opacity: 0%;

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

0%

25%….

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 элемента

CSS-свойство **z-index** определяет положение [позиционированного](https://developer.mozilla.org/ru/docs/Web/CSS/position) элемента и его дочерних элементов или флекс-элементов по оси z. Перекрывающие элементы с большим значением z-index будут накладываться поверх элементов с меньшим z-index.

20. Чем отличаются переходы от анимации?

[Анимация](https://translated.turbopages.org/proxy_u/en-ru.ru.bda7e5f9-6368ca64-75fd2a6b-74722d776562/https/www.geeksforgeeks.org/css-animations/)CSS позволяет анимировать элементы HTML, в отличие от перехода, который выполняет только операцию из точки A в точку B

[Переходы](https://translated.turbopages.org/proxy_u/en-ru.ru.bda7e5f9-6368ca64-75fd2a6b-74722d776562/https/www.geeksforgeeks.org/css-transitions/) CSS позволяют плавно изменять значения свойств, но их всегда нужно **запускать**, как при наведении курсора мыши. Изменения свойств вступают в силу не сразу

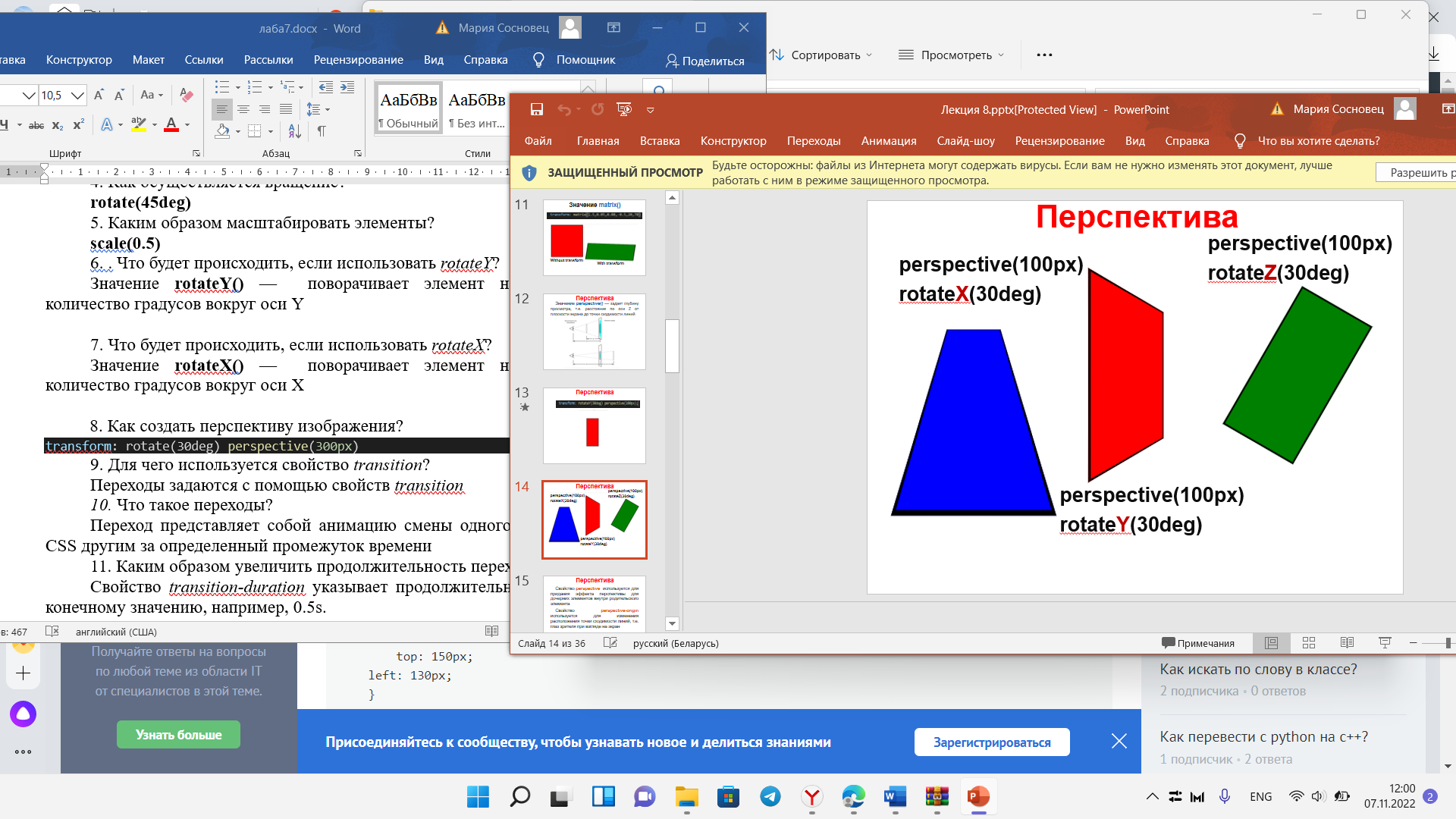
21. Как остановить анимацию?

animation-play-state: pause

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);



animation direction reverse  анимация проигрывается в обратном порядке, от последнего ключевого кадра до первого, после чего возвращается к последнему кадру

background-repeat:no-repeat - запретить повторение, показать только одно изображение

background-size: cover;

1. Непрерывное перемещение 1 копии по горизонтали на 200px;
2. @keyframes anim1 {
3. 0% {left: 0px;}
4. 50% {left: 200px;}
5. 100% {left: 0px;}
6. }

2. Непрерывное перемещение 2 копии по прямой с изменением формы;

@keyframes anim2 {

    0% {top: 0px}

    50% { left: 100px;

    border-radius: 80px;}

    100% { left: 200px;}

}

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

@keyframes anim3 {

    0% {left: -900px;}

    100% { left: 800px;}

4. Непрерывное перемещение 4 копии по вертикали вниз на 50px;

@keyframes anim4{

    0%{top: 0;}

    100%{top: 50px;}

}

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

@keyframes anim5 {

    0% {left: 0px;

    top: 0px;}

    25% {left: 0px;

    top: 50px;

    }

    50% {left: 50px;

    top: 50px;

    }

    75% {left: 50px;

    top: 0px;

    }

    100% {left: 0px;

    top: 0px;

    }

}