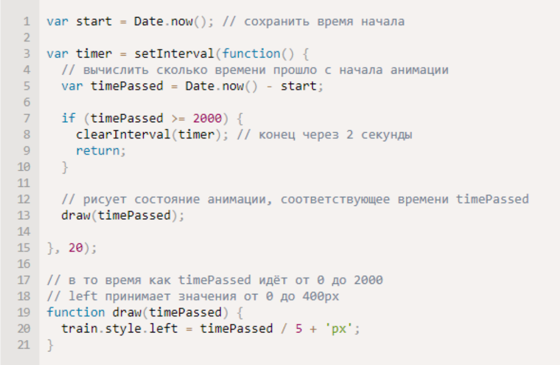
School:		
Date:	**Teacher‘s name: Syzdykova A.A.**	
Grade:	**Number present:**	**absent:**
Topic of the lesson: Animation effects. Widgets		
Learning objective(s) that this lesson is contributing to	To teach students how to create animations in a web programming environment using a Java script	
Lesson objectives	**All learners will be able to**: • Know the types of scripts for creating animations **Most learners will be able to**: • Recognize script assignments for creating animations **Some learners will be able to:** • create a web page using a Java script	
Assessment Criteria	**All learners will be able to**: • Know the types of scripts for creating animations **Most learners will be able to**: • Recognize script assignments for creating animations **Some learners will be able to:** • create a web page using a Java script	
Value links	Building respect for each other's opinions, responsibility, communication skills, critical thinking	
Previous learning	JQuery, jQuery UI, CSS Libraries	
Cross curricular links	Computer science, mathematics	

Time	**Planned activities**	**Resources**
Beginning 5 min	Greetings. Division into subgroups. Puzzles Strategy Descriptors Lay out the pictures in sequence. Sign the pictures. During the discussion of the completed assignment, we arrive at the goal setting. Repeating the previous topic: Hot Microphone Method One student asks a question and passes the microphone, and the next student answers the question and asks his question to the next student, etc.	Picture cards Microphone
Middle 20 min	4. Next, each group in a circle calls a single information or fact, while not repeating the above (a list of ideas is being compiled). 5. All information is briefly written in the form of abstracts by the teacher in the "basket" of ideas (without comments), even if they are erroneous. You can "dump" facts, opinions, names, problems, concepts related to the topic of the lesson into the basket of ideas. Further, during the lesson, these facts or opinions, problems or concepts that are scattered in the child's mind can be connected in logical chains. 6. All errors are corrected further, as new information is mastered. Formative Assessment: Thumb Strategy How do you think you coped with the task? Did everyone succeed?	Cards with a new theme App No. 1

<table>
<tr>
<td></td>
<td>

Depending on the success of the assignment, students lift their finger up, to the side, to the bottom.
Task number 1.
Creating animation by pattern No. 1



```
var start = Date.now(); // сохранить время начала

var timer = setInterval(function() {
  // вычислить сколько времени прошло с начала анимации
  var timePassed = Date.now() - start;

  if (timePassed >= 2000) {
    clearInterval(timer); // конец через 2 секунды
    return;
  }

  // рисует состояние анимации, соответствующее времени timePassed
  draw(timePassed);

}, 20);

// в то время как timePassed идёт от 0 до 2000
// left принимает значения от 0 до 400px
function draw(timePassed) {
  train.style.left = timePassed / 5 + 'px';
}
```

</td>
<td>

Card No1

</td>
</tr>
<tr>
<td><b>End
15 min</b></td>
<td>Task number 2. Create an animation on a web page using a Java script
Moving an object advertising page.
Reflection:
Reflective Card Method
It was interesting…
I realized that ...
I managed…
I’ll tell others at home that ...</td>
<td>“Reflection”
Cards</td>
</tr>
</table>

Differentiation – how do you plan to give more support? **How do you plan to challenge the more able learners?**	**Assessment – how are you planning to check learners’ learning?**	**Health and Safety**

Cards with a new theme

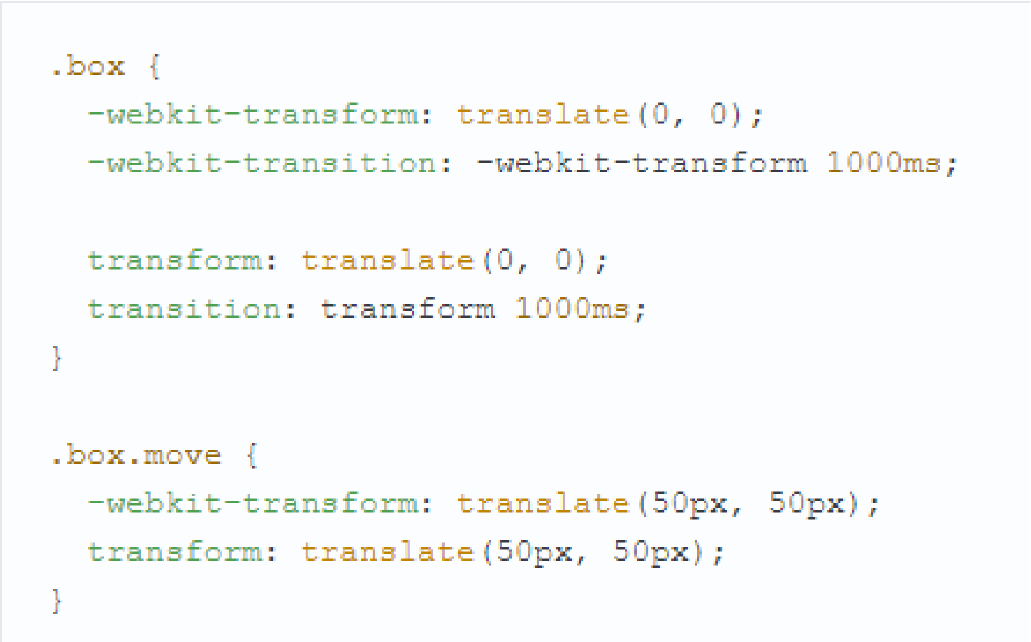
App No. 1

JavaScript animation and CSS animation

There are two main ways to create animations: using JavaScript, using the web animation API, and using CSS. The choice of the method depends on the specific task, so I would like to note right away that it is impossible to unequivocally talk about the advantage of one technology over another.

CSS animation

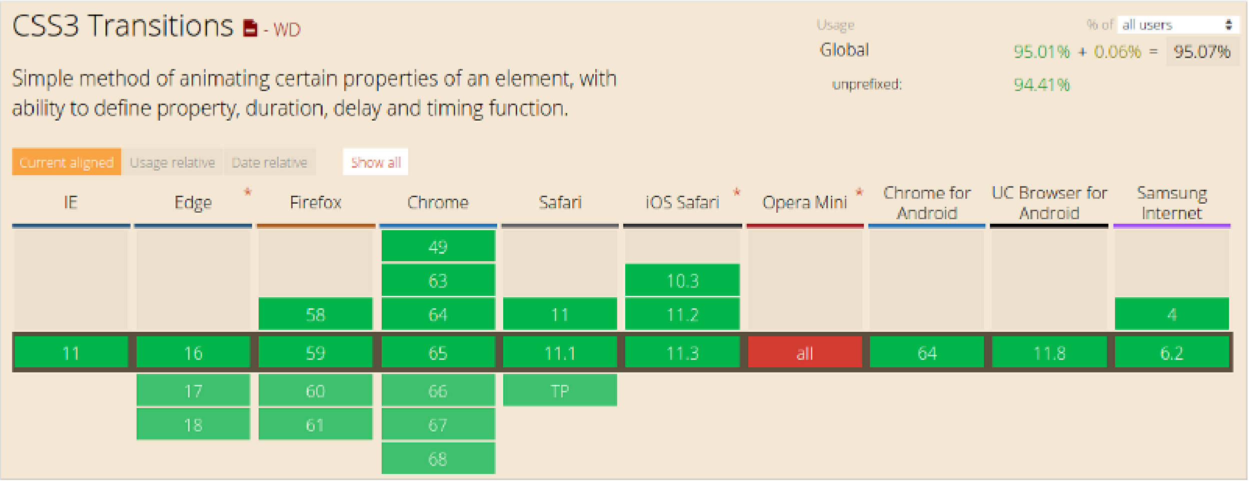
CSS animation is the easiest way to get something moving around the screen. Let's start with a simple example that demonstrates moving an element along the X and Y axes. This is done using the translate CSS transformation, which is configured for a duration of 1000 ms.



```css
.box {
  -webkit-transform: translate(0, 0);
  -webkit-transition: -webkit-transform 1000ms;

  transform: translate(0, 0);
  transition: transform 1000ms;
}

.box.move {
  -webkit-transform: translate(50px, 50px);
  transform: translate(50px, 50px);
}
```

When you add the move class, the transform value changes and the transition begins. In addition to the duration, we can adjust the dynamics of the animation (easing). The essence of this setting is that it affects how the user perceives the animation. We will talk about the dynamics of animation later.

The illustration below shows CSS support for modern browsers.



CSS3 Transitions - WD

Simple method of animating certain properties of an element, with ability to define property, duration, delay and timing function.

Usage % of all users

Global 95.01% + 0.06% = 95.07%

unprefixed: 94.41%

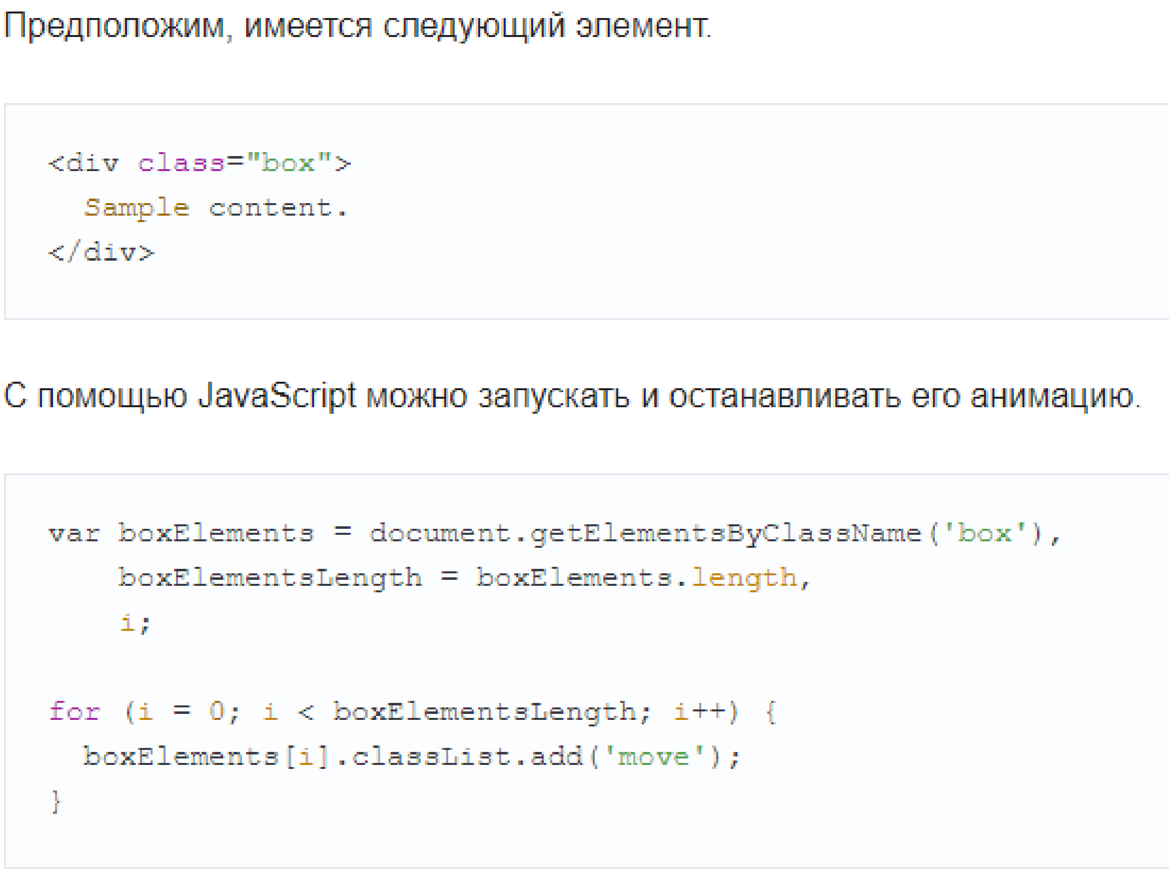
Current aligned
Usage relative
Date relative
Show all

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			49						
			63		10.3				
		58	64	11	11.2				4
11	16	59	65	11.1	11.3	all	64	11.8	6.2
	17	60	66	TP					
	18	61	67						
			68						

As you can see, this feature has a very high level of support.

If, as in the previous code snippet, you create separate CSS classes to control the animation, then you can enable or disable the animation using JavaScript.

Suppose there is the following element.



Предположим, имеется следующий элемент.

```
<div class="box">
  Sample content.
</div>
```

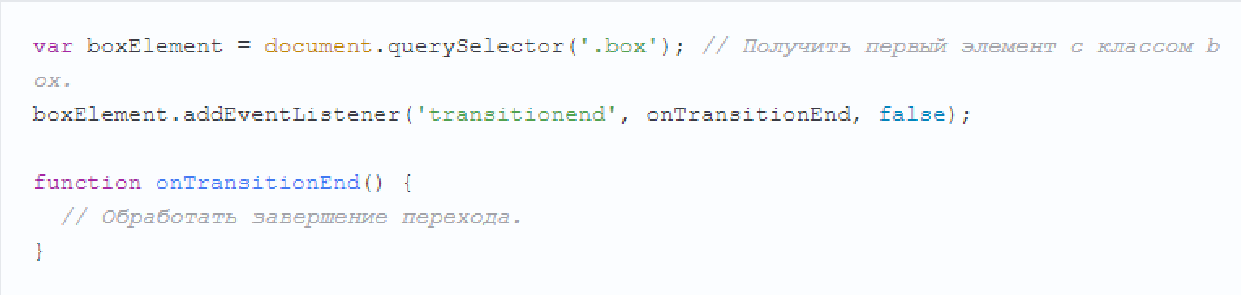
С помощью JavaScript можно запускать и останавливать его анимацию.

```
var boxElements = document.getElementsByClassName('box'),
    boxElementsLength = boxElements.length,
    i;

for (i = 0; i < boxElementsLength; i++) {
  boxElements[i].classList.add('move');
}
```

In this code fragment, we take all the elements to which the box class is assigned and add the move class to them in order to start their animation.

Similar CSS sharing features - for describing animations, and JS - for starting and disabling it, make the application well balanced. A developer can focus on managing the state of elements from JavaScript by simply assigning appropriate classes to the target elements, allowing the browser to independently perform animations described using CSS. If you delve into a similar scenario of working with animation, you can listen to the transitionend event of the element, but you should only do this if you support older versions of Internet Explorer.

The transitionend event is raised at the end of the transition. Here's how to work with it.



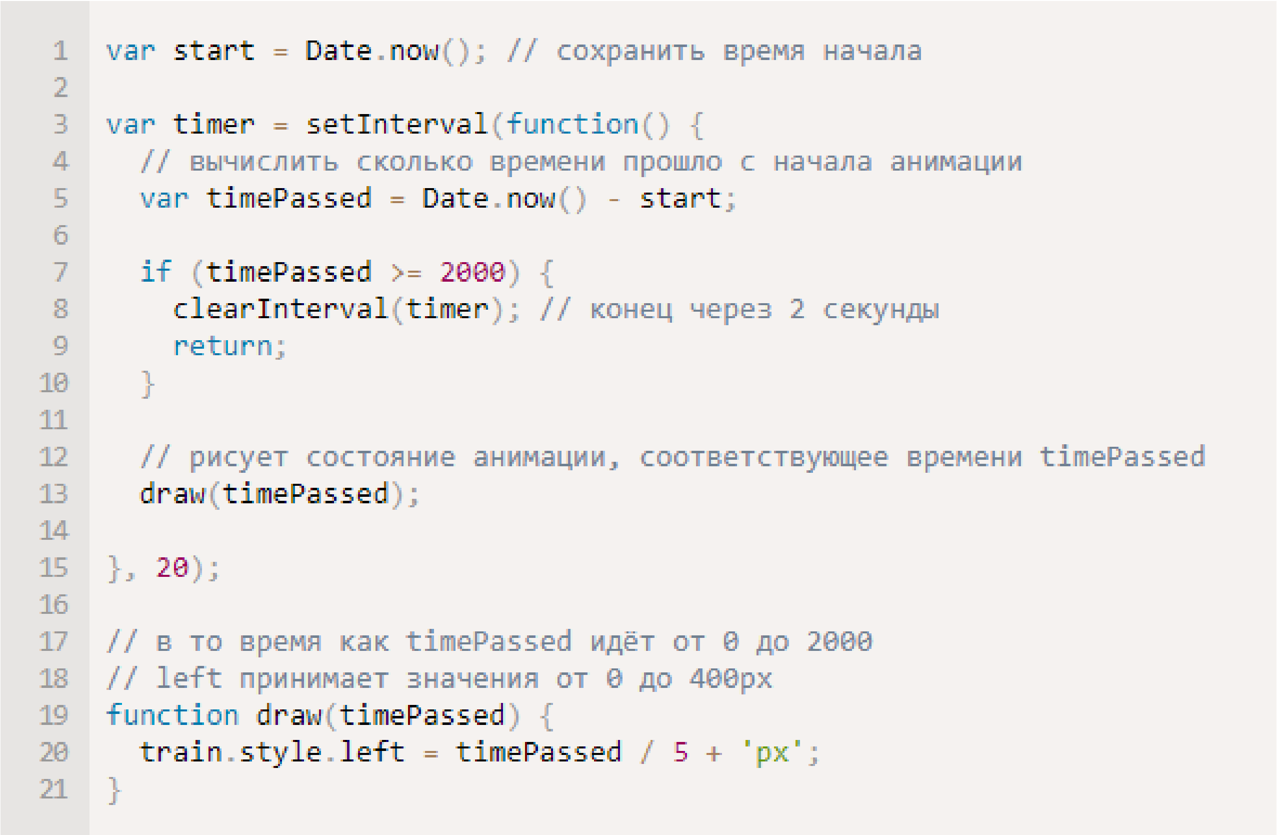
```
var boxElement = document.querySelector('.box'); // Получить первый элемент с классом box.
boxElement.addEventListener('transitionend', onTransitionEnd, false);

function onTransitionEnd() {
  // Обработать завершение перехода.
}
```

In order to make the web interface elements more dynamic, in addition to using CSS transitions, you can also use CSS animations. They give the developer a much greater level of control over individual keyframes of the animation, the duration of the stages of the animation, and iterations of the animation.

Keyframes are used to tell the browser what CSS property values should have at given times. The browser independently finds intermediate values for the properties when moving from one key frame to another.

Card

No. 1



```
var start = Date.now(); // сохранить время начала

var timer = setInterval(function() {
  // вычислить сколько времени прошло с начала анимации
  var timePassed = Date.now() - start;

  if (timePassed >= 2000) {
    clearInterval(timer); // конец через 2 секунды
    return;
  }

  // рисует состояние анимации, соответствующее времени timePassed
  draw(timePassed);

}, 20);

// в то время как timePassed идёт от 0 до 2000
// left принимает значения от 0 до 400px
function draw(timePassed) {
  train.style.left = timePassed / 5 + 'px';
}
```