

Nysanov E.A., Tumashbaev S.

M.O. Auezov South Kazakhstan State University, Kazakhstan, Shymkent

THE USE OF CSS ANIMATIONS IN CSS3

Introduction. This article discusses modern computer graphics and CSS animation, is an auxiliary material to the practical lesson on computer graphics. A beginner animator working from nature, from memory and imagination can not neglect the conditions of the tonal construction of this object. Proper construction of the tonal range in accordance with the set creative task- an important condition for the creation of this animated work with the use of CSS tools.

In this material plays a big role in flowering. Therefore, we offer special exercises designed to assimilate the initial knowledge of the methods of constructing the tonal structure of modern computer graphics and CSS animation. Tonal exercises are based on the method of gradual complication of tasks – from the compilation of simple details of flat geometric shapes to the development of achromatic images. This article is proposed for use by teachers and students in theoretical and practical classes in computer graphics and animation, also when working in the open air practice. Light-density relations of colors have long been the subject of careful study and programmers, animators, and Methodists in this area. It is known that the same color tones, taken in different light-density relations, produce a very different emotional impression on us.

Professional animator dealing with color, it is extremely important to identify the role and importance of light-density relationships of colors and ways of their construction.

Looking through the well-known programs of modern computer graphics and CSS animation, we are amazed at how coloristic animation is built holistically. Each detail works on its General ideological content, each color spot seems deeply thought out, calculated by tone and area relative to the size of the whole. In such animations it is impossible not to add or reduce a single color spot. Here, form and color are in one

harmony. One of the conditions of such sounding is competent construction of the General tonal basis of animation, finding of the optimum sizes of the area of each spot in their contrast combination.

How to animate? CSS - transition property animation. There are two main tools in CSS that we can use to move elements around a web page. The first property is the transition. First of all, it is with its help that we create animations in CSS. The second tool is the animation property paired with keyframes (@keyframes – keyframes). Now we will look at the CSS transition property. And we'll talk about what you need to animate on the page (when the user needs it), and what not to animate (when the animation looks stupid and inappropriate), based on the user experience (UX — user experience). CSS transition property-HTML and CSS parameters and values.

```
transition: [property] [duration] [timing function] [delay];
```

```
transition: height 1s ease-out 0.2s ;
```

We assign this property to the element we want to animate. This element gets a smooth transition (or step-by-step) between several of its States (for example, the height of 100 pixels and the height of 200 pixels). And how this transition, will depend on the parameters that we set it. The first parameter (value) of the transition property is another property of the animated element (for example, height).

The second parameter is the duration of the animation, that is, how long it will take the element to move from one state to another (for example, 2s or 2000ms).

The third parameter is the timing function [timing function] (using function [easing function]). How the intensity of the animation will be distributed over time. For example, the animation can start abruptly, and at the end slow down and smoothly finish the transition. Can be used as keywords (e.g., ease, ease-in-out, linear), and the functions of the bézier cubic (for example, cubic-bezier (0.17, 0.67, 0.83, 0.67)). Cubic Bezier you can easily and conveniently set up on this resource <http://cubic-bezier.com>, as well as the steps function to create a frame-by-frame animation (step-by-step).

Animation delay is the amount of time you need to wait before the animation (in our case, the transition) starts. Example of CSS transition animation. Let's look at an example of CSS animation: `transition: opacity 300ms ease-in-out 1s;`

Here we animate only the opacity property. This means that if the element we are going to animate has other properties, such as height, width, or color, and even if these properties differ in different States of the element, only the opacity property will have a smooth transition (animation). You can also use the "all" keyword instead of a specific property. This means that we animate absolutely all the properties of the element that have been changed in the new state of the element (not just opacity), and that can be animated at all. Because, as you know, not all properties can be animated.

The second parameter (300ms) tells us that the animation will last only 300 milliseconds. That is, the element will quickly appear or dissolve quickly, depending on the property values in its two States. Timing function (ease-in-out) the third parameter will make the beginning and end of the animation smoother. Delay (1s) shows how much needs to be late animation before her triggered. In General, this is the formula of how we will record the transition animation. You can, if anything, add more than one [property] transition for a single item. For example, you can animate the height and width of an element with different parameters. To do this, use a comma in the transition property instead of a semicolon at the end and write parameters for another property. And don't forget to put a semicolon at the end to make it work.

What can be animated in CSS? So far we have only discussed the syntax of the transition property. Now let's talk about properties, which ones we can animate and which ones we can't. Because there are some things that just don't make sense to animate, and some properties can't be animated. Animated properties Take, for example, properties such as: font-size; background-color; width; left (how far from the left side an element can be moved) [positioned]... Obviously, animating these properties makes sense. If you change their values, it will change the element visually. If you change the font size, say, from 14 to 28 pixels, you will see that the font has smoothly increased, there was an animation that lasted for some time.

Animation always makes sense if the property values are clear (often numeric) values. If you increase the font size to 100 pixels, you will clearly see how the letters grow. A smooth change in the background color can also be observed [because the color on the web has a numeric code, for example as red `rgb(255,0,0)`]. These properties are animated. In General, remember, if you imagine that it is animated-then it can be animated. Most often it works. Non-animated properties here is a list of properties that cannot be animated, for example: `display`; `font-family`; `position`... These are some of them that I took as an example in order for you to feel the difference between animated and non-animated CSS properties.

Take `display`. Can you imagine how visually the change will occur between «`display: block;`» and «`display: inline-block;`»? How can the appearance of an element change smoothly between «`position: relative;`» and «`position: absolute;`»? No, of course, the element will look different when you change these properties. But how can you imagine the transition? You can't do that. You can't imagine turning a Helvetica font into a Georgia font, every letter, it just won't work. You can change these fonts, but they will change dramatically, no animation will happen.

The performance of animations in CSS. In General, many properties are animated, but some are not. Now let's decide which of the animated properties are better suited to animating and which are worse. Speech about performance. We touched on the topic of performance, because animation is a resource-intensive process. In General, here are the things that work best for animation: Positioning on the page Scaling, Rotation And Transparency, the first three items in this list, it is all the parameters of the `transform` property (`translate`, `scale`, `rotate`). Positioning occurs X and Y axes. If you decide to animate something else, you run the risk of not getting a smooth and soft transition of 60 frames per second. Paul Lewis & Irish And that's exactly what we're interested in when we talk about animation performance. In General, the most appropriate properties for animation are `transform` and `opacity`. When animating something another, the fps of the animation can drop far below 60fps. Let us summarize this point as follows. Avoid redrawing elements on a web

page and animating elements that are difficult for the browser to render (such as shadows).

How to make animation in CSS? Now let's see how to run the animation. What you need to do to start animating the element. In other words, how to make an animation in CSS. It's not difficult and there are many different ways to. But there are 2 basic, which you must learn without fail.

The first method is animation when hovering over an element (pseudo-class :hover). You induce the mouse cursor over the element, and the animation is triggered. This way, you can animate the element itself or any of its children. Then I'll show you how it works.

The second way is to change the class of the element. That is, you have a regular element, you change (or add) its class to a class that has very different styles. If you add a class, the animation will occur, and if you remove the class, the reverse animation will occur. This is a good way to animate menus or drop-down lists.

Below are examples of animations created for the CSS3 specification (figure 1-2).

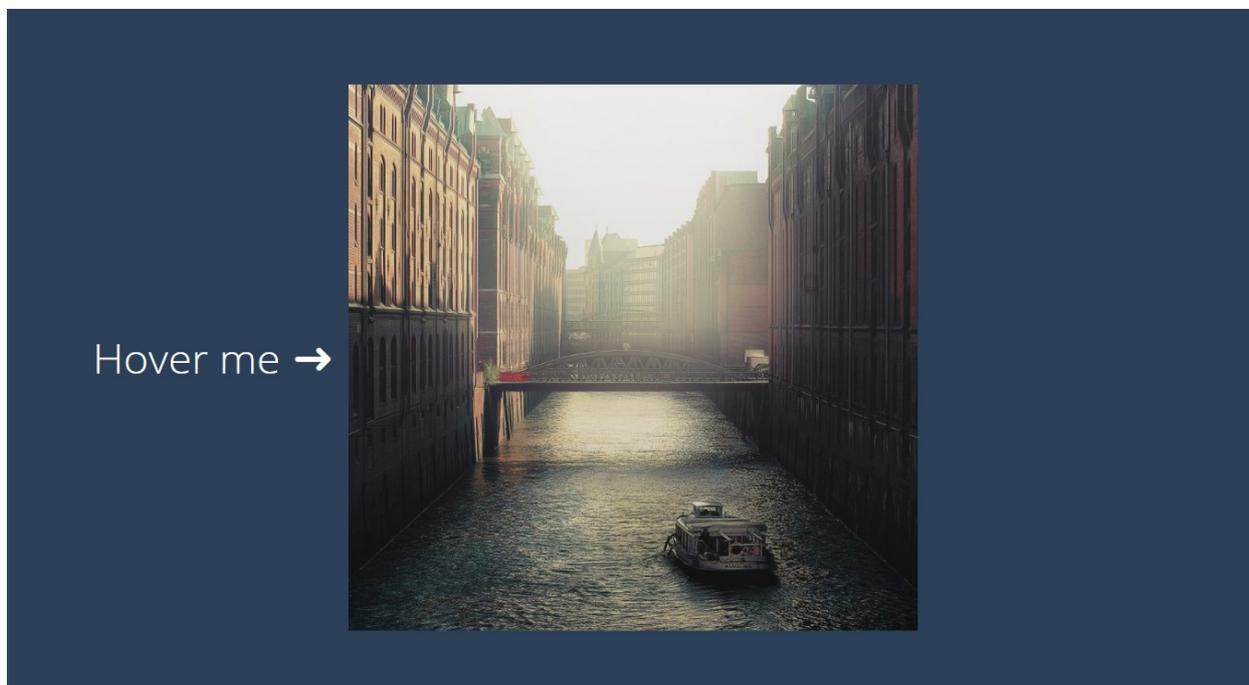


Figure 1

In the case of transition, CSS has a universal animation property, where you can specify all animation parameters in one line, separating them with a space. For

clarity, below are options for long and short recording of all animation properties (to save space, we did not include manufacturer prefixes in our example, but in real practice they will need to be added):

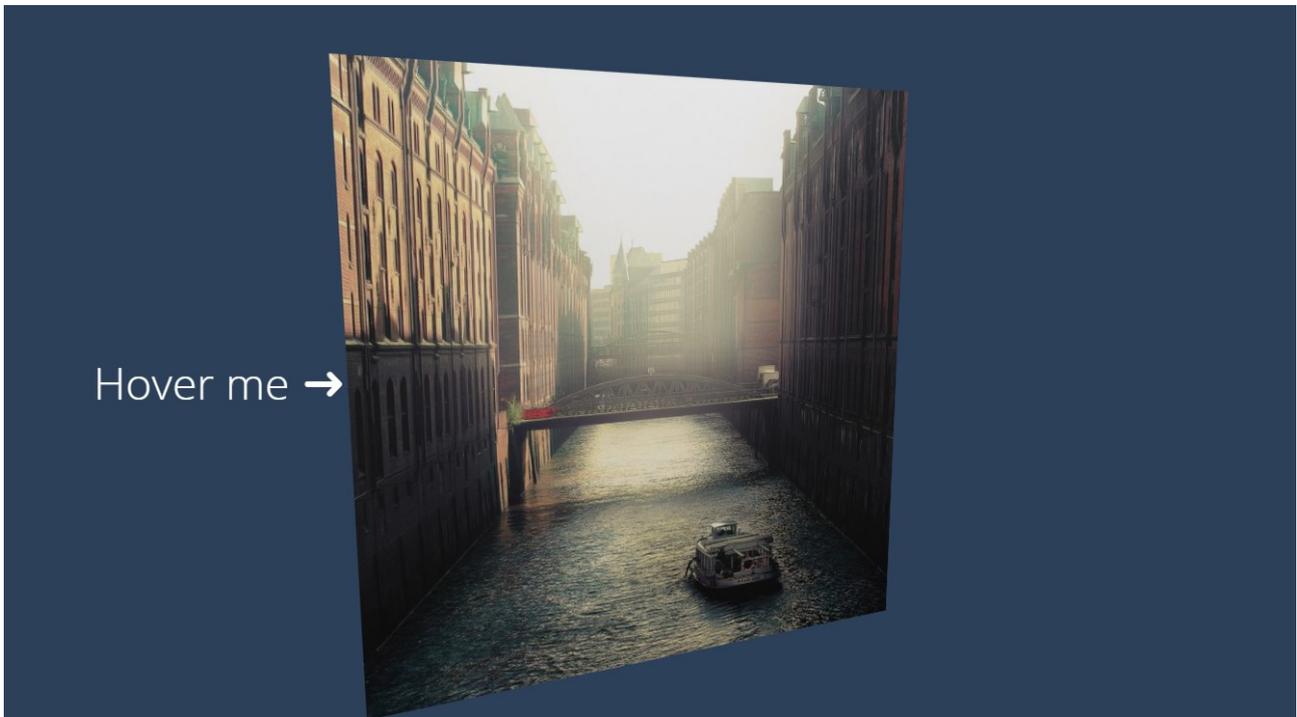


Figure 2

(program)

```
div {  
  animation-name: coolAnimation;  
  animation-duration: 3s;  
  animation-timing-function: ease-in;  
  animation-delay: 500ms;  
  animation-iteration-count: 6;  
  animation-direction: reverse;  
  animation-fill-mode: forwards;  
  animation-play-state: paused;}  
/* The style above is equivalent to the following: */  
div { animation: coolAnimation 3s ease-in 500ms 6 reverse forwards paused;}
```

Of course, there is no need to use each of the eight animation properties, because only two are required - animation-name and animation-duration. And if you need to

add a third parameter (for example, animation-iteration-count) , you can make the following entry:

```
div {  
  animation: coolAnimation 3s 6;}
```

If you want to apply several animations to an element at once and set different settings, separate each group of values with a comma and put a semicolon at the end:

```
div {  
  animation: firstAnimation 300ms ease-in,  
  secondAnimation 1s linear alternate;}
```

And also we will create both ordinary CSS transitions transition, and more complex things using the animation property.

References

1. V.Petkov. Animacija dlja specifikacii CSS3. Komp'juterPress, 2001. – 224 s.
2. Jefferktivnaja rabota: Animacija dlja specifikacii CSS3. A.A. Tajc, A.M. Tajc, M.N. Petrov. – SPb.: Piter, 2003. – 768 s.: il.
3. Retushirovanie i obrabotka izobrazhenija v animacii dlja specifikacii CSS3. / Ajmann Kjetrin.: Per. s angl. – M.: Izdatel'skij dom «Vil'jams», 2003. – 304 s.: il. – Paral. tit. angl.
4. A.Tajc. Animacija dlja specifikacii CSS3. Moskva, "Piter", 2003
5. T.Buljanica Dizajn na komp'jutere: Samouchitel'. – SPb.: Piter, 2003.
6. D.Aderking, Retush' i vosstanovlenie fotografij, Moskva, „Dialektika“, 2000.
7. L.A.Zalogova Komp'juternaja grafika. Jelevktivnyj kurs: -M.: BINOM. Laboratorija znanij, 2005 g.