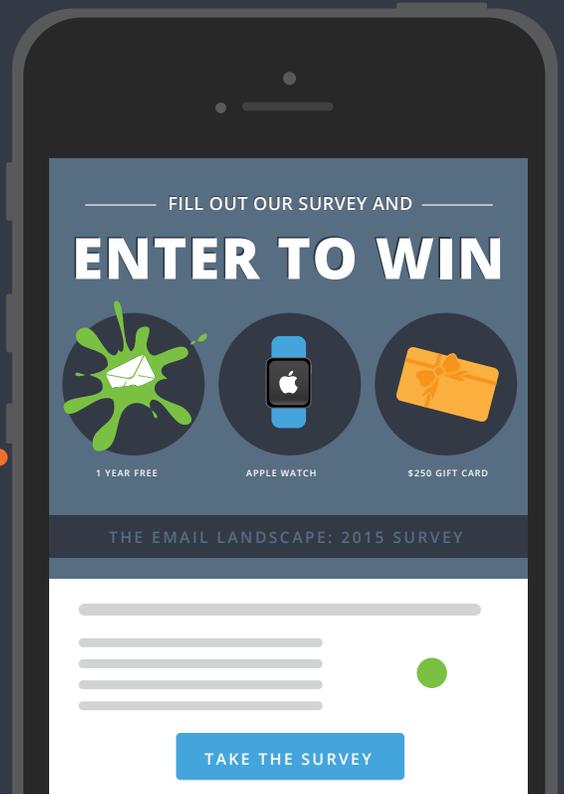


[  CLICK SEE THIS ANIMATION IN ACTION ]



# THE DEFINITIVE GUIDE TO CSS ANIMATION IN EMAIL

# CONTENT

03

CSS Animation + Email

04

Animated CSS  
vs Animated GIFs

05

Considerations Before  
Using CSS Animations

07

Examples of CSS Animations

09

A More inDepth Look  
at CSS Animation

10

Email Animation Techniques

15

Code Drawbacks of  
CSS Email Support

17

Creating the Best Email  
Experience, in Every Client

# CSS Animation + Email

Standing out from the crowd is a constant challenge in email marketing. Templates all begin to look the same, and your subscribers have seen a thousand CTAs by now. Enter: Movement in email. Whether it be through GIFs, CSS animations, SVGs, WebGL or another form of animation, movement can be an extremely powerful tactic in your mailing strategies to increase your click-through rate and in turn, your bottom line.

Animation is a CSS property that allows us to animate individual HTML elements such as `img`, `span` and `div`. This eliminates the need to use an external scripting language such as JavaScript or Flash, which are not supported in email. It also eliminates the need to generate animated GIFs.

In this guide, we'll give a complete breakdown of CSS animations for email. The first section will cover code support, examples, and considerations for using them. Then, we'll jump into code breakdown, takeaway code and higher-end technical specifications. But first, let's start by explaining the pros and cons of GIFs vs. CSS Animation in your email campaigns. **Enjoy!**



# Animated CSS vs Animated GIFs

## The Pros and Cons

There is a time and place to use CSS animations and animated GIFs, but it may seem confusing at first. Let's weigh up the pros and cons of using each type of animation.

### ANIMATED GIFS

- + Easy to create if you have the know-how
- + Show off products in restricted space
- + Good support across major email clients
- Outlook 2007/2010/2013/2016 will only show the first frame of the animation
- Large GIFs can be slow to load and will eat into a subscriber's data plan if viewed on a mobile

### CSS ANIMATIONS

- + Lightweight and fast to load
- + CSS Animations are much smoother compared to other methods of animating
- + Looks very good on high quality mobile screens
- Knowledge required on how to create and troubleshoot
- Limited client support

As you can see, GIFs do have some advantages, but this guide will be focusing mainly on CSS animations. We'll show you how to handle the negatives and use CSS animations to their maximum potential.



# Considerations Before Using CSS Animations

CSS animations aren't going to meet all email needs, so it's important you consider all the points below before diving in.

## 1 TIME

Time is a very important factor when considering making any changes to your email campaigns. We need to consider how much time it's going to take to create a CSS animation:

- Will the animation be complex and take many hours to create? If so, consider the use of animated GIFs.
- Will the animation take a lot of time to test across different platforms?

That being said, you can definitely mitigate these time issues by creating an animation "framework." Every time you create a cool new CSS animation, whether it be a spinning icon, items re-arranging themselves, or a subtle hover animation on a CTA, consider saving the animation code by itself. That way you can create a reusable library of animations you can pull into your email campaigns. This is an example of **modular design**, which can really speed up email development.

## 2 SUBSCRIBER BASE

Perhaps the most important consideration is your subscriber base. CSS animations are supported across a mix of different email clients. However, if you're dealing with primarily a B2B subscriber base you'll definitely want to check the clients your subscribers open emails in. If only 10% of your list uses a device or email client that will support CSS animations, you may want to use an animated GIF instead.

## SUBSCRIBER BASE [CONTINUED]

If you're wondering what the support for CSS animations is like, you can use this handy list:

iOS Mail ✓

Apple Mail ✓

Android Native Mail ✓

Yahoo! Mail ✗

Gmail ✗

Windows Phone 7 ✗

Outlook.com ✗

Office 365 ✗

Outlook 2002/2003 ✗

Gmail App ✗

Outlook 2007/2010/2013/2016 ✗

AOL Mail ✗

Outlook for Mac ✓

### 3 HOW YOU USE YOUR ANIMATIONS

Though less technical, this is still a serious consideration for anyone who wants to try using CSS animations in their email!

Think carefully about how you're actually going to use your animations. I believe that this kind of enhancement needs to be done for a reason. I once heard that "Design for the sake of design is mess," and I run with the same principle here.

If you're using flashy animations in every email campaign just to use animations, chances are you're going to either lose your original message or desensitize people to the animations, thereby reducing their effectiveness in future campaigns. Luckily, below I have some clever examples of how to use CSS animation to enhance your emails.



# EXAMPLES of CSS animation



This fantastic Halloween email was created by UK agency [1973](#) for Seagate. They created an amazing animation, moving the ghost spirits on an arc to have the effect of them flying down the email.

They also wrote [an excellent blog](#) on how they created this animation, which I'd highly encourage reading.

Another example coming out of the UK is this email by the creative email agency known as display block. They've used a few different animation styles to include a bit of motion in their email.

By moving a background layer behind an image they create the effect of a scrolling iPhone:



I encourage checking out the original email, it's full of both animated CSS and animated GIFs that create a really nice effect.

[Original Email Link](#)

Here they move the heart up and down, almost making it look like a beating heart:



# Examples of CSS animation

{continued}



Table TR TD, a fantastic little email blog - check it out, created this amazing email. They use a mixture of clever timing and good use of color to create this incredible dripping style for their email.

**Note:** All of these examples are relying on animated GIFs for me to show you how the animation looks, I'd highly encourage viewing the original emails to see the beautiful animations.

Now let's dig into the actual code, to show what you can and can't do and even give some nice little code takeaways for you to drop straight into your email campaigns!



## A more in Depth Look at CSS animations

Let's look at what CSS animations really are. By using CSS we can add animation to most HTML elements without using common web practices such as JavaScript or Flash. What that means for us email developers is that we can use it as an enhancement for email clients that support CSS.

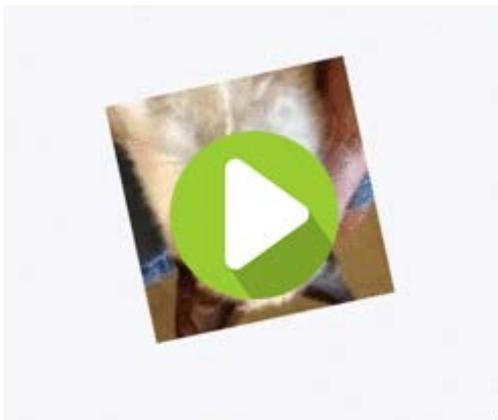
So, what can we actually do? We'll cover each of these points more thoroughly later in the guide but for now we have a brief overview:

- Specify which HTML elements we'd like to animate
- Set timing (easing) functions to change the rate we switch between properties
- Specify the duration of how long the animation should take
- Animate the elements using a small variety of CSS properties

# Email Animation TECHNIQUES

As email developers we're used to having limited support for web techniques in email and, unfortunately, CSS animations are no exception. The following are a couple ways you can add animation to emails with at least moderate support.

## ROTATE



One of the most common animation techniques around, a simple rotate can add a lot to your email technique. We'll also break down each line of code to see what's really going on.

Grumpy Cat on the left is what we're going to achieve.

Here's the code that handles the animation.

```
@-moz-keyframes spin { 100% { -moz-transform: rotate(360deg); } }  
@-webkit-keyframes spin { 100% { -webkit-transform: rotate(360deg); } }  
@keyframes spin { 100% { -webkit-transform: rotate(360deg);  
transform:rotate(360deg); } }
```

So, let's look at what's going on here! This bit of code:

```
transform:rotate(360deg);
```

It's the magic ticket, we use this to control the animation. It's pretty simple, but you can see we're simply telling the browser to "transform" the CSS element. The transform we're calling in is the rotate transformation and we're going to rotate it 360 degrees.

# Email Animation Techniques

{continued}

Then, the rest of the code is simply covering different browsers and email clients. You can see we're targeting Mozilla clients, WebKit clients and non Webkit clients alike. If you'd like to read more about the different types of web browser and WebKit vs non-WebKit I'd recommend [this fantastic Stackoverflow question](#).

So, now that we've set up the CSS animation we simply need to declare it for the element(s) we wish to animate. Just create a class and call the animation we created, named "spin" like so:

```
.spin {  
  -webkit-animation:spin 4s linear infinite;  
  -moz-animation:spin 4s linear infinite;  
  animation:spin 4s linear infinite;  
}
```

You can see we've done what we did before with calling the animation for Mozilla, WebKit and non WebKit clients. All we're doing here is telling the browser that any element with the class spin applied to it, we want to call our animation and have it spin for 4 seconds for an infinite loop.

Pretty simple right? How about checking out some more modifications

## FADING



Here we're going to use the same principles as we just looked at above to create a fading animation. This is what we're going for on the left:

Again, just like above, we're going to target the three main different browser types to ensure our code appears in every possible client it can.

```
@-webkit-keyframes fade {
  0% { opacity: 0; }
  50% { opacity: 1; }
  100% { opacity: 0;}
}

@-moz-keyframes fade {
  0% { opacity: 0; }
  50% { opacity: 1; }
  100% { opacity: 0;}
}

@keyframes fade {
  0% { opacity: 0; }
  50% { opacity: 1; }
  100% { opacity: 0;}
}
```

The big difference here is that we've thrown in some extra keyframes. All this means is that at 0% of the animation cycle the opacity will be 0, at 50% or halfway through it'll be 1 and then at 100% of the animation cycle it'll be back at 0 opacity.

By using this keyframe method, we can control exactly what we want our animation to do and when. By having it start and end on 0 opacity, we create a nice fading loop.

We call it exactly the same as we did for the rotation:

```
.fade {
  -webkit-animation:fade 3s linear infinite;
  -moz-animation:fade 3s linear infinite;
  animation:fade 3s linear infinite;
}
```

Pretty easy to achieve!

# Email Animation Techniques

{continued}

## SIMULATED "LOADING BAR"

Here we're going to use a sequence of changing background colors to mimic the effect of a loading bar, like so:



As this code is a bit longer and a bit more complicated, I've put it onto Codepen so we don't fill the page but I'll be referencing the code here.

For the full working code [see here](#).

Basically, what we're doing here is creating 5 different animations. Each one has 5 stages of animation:

- 0%
- 25%
- 50%
- 75%
- 100%

We color each box black to create this effect at a different stage of the animation. For example, the first box has this:

```
0% { background: #000000 }
25% { background: #ffffff }
50% { background: #ffffff }
75% { background: #ffffff }
100% { background: #ffffff }
```

All we're saying here is that for the first 25% of the animation cycle, the background color will be black for that box.

On the third box we have the same sort of thing, but the cycle will change the box to black at the third 25% of the animation:

```
0% { background: #ffffff }
25% { background: #ffffff }
50% { background: #000000 }
75% { background: #ffffff }
100% { background: #ffffff }
```

If you look at the full code, you'll also notice we still make sure to target the three browser types - hence why the code is so long!

## BACKGROUND ANIMATIONS

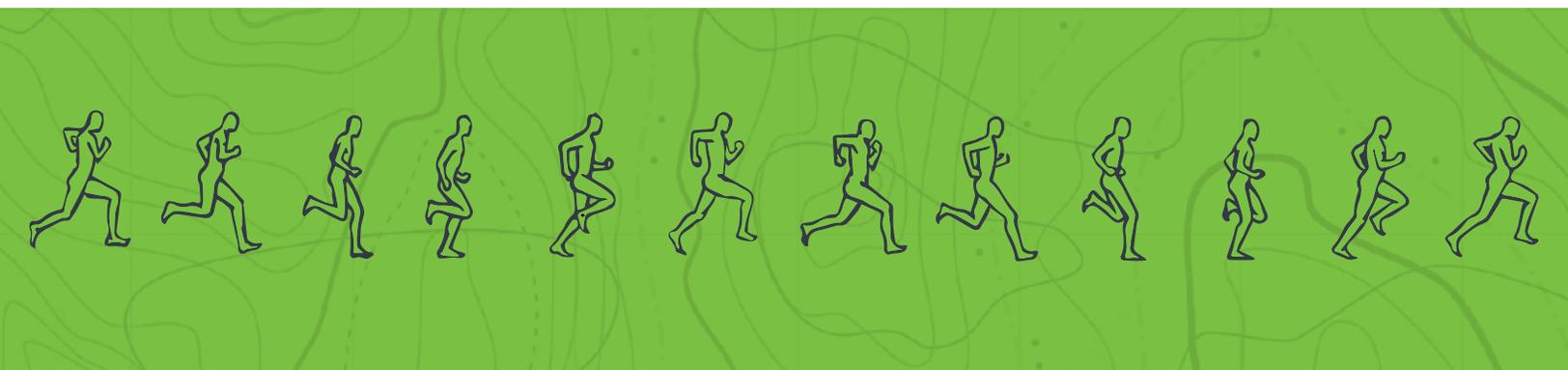
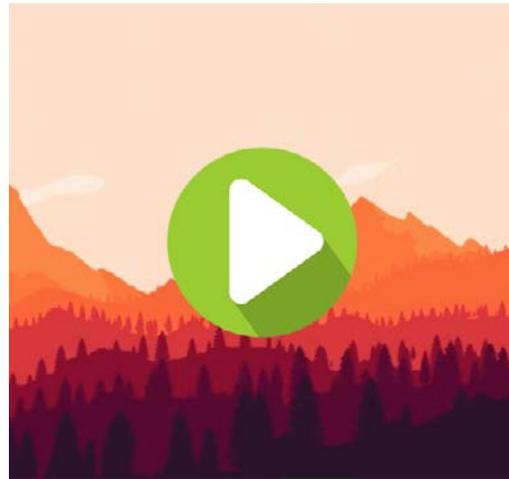
A really powerful feature we have access to is the ability to animate backgrounds. This is one of my favorite ways to add subtle animations to give your emails an edge without having to worry about fallbacks because when it doesn't animate it's just a regular background.

*Please note: I haven't included [VML fallbacks for Outlook background images](#) in this code sample, so you'll need to add them yourselves.*

This is the effect we'll be creating:

[Check out this working code sample](#) before you get started.

So how do we create this amazing effect? Again, it's all about getting your head around what's supported and how to use keyframe animations.



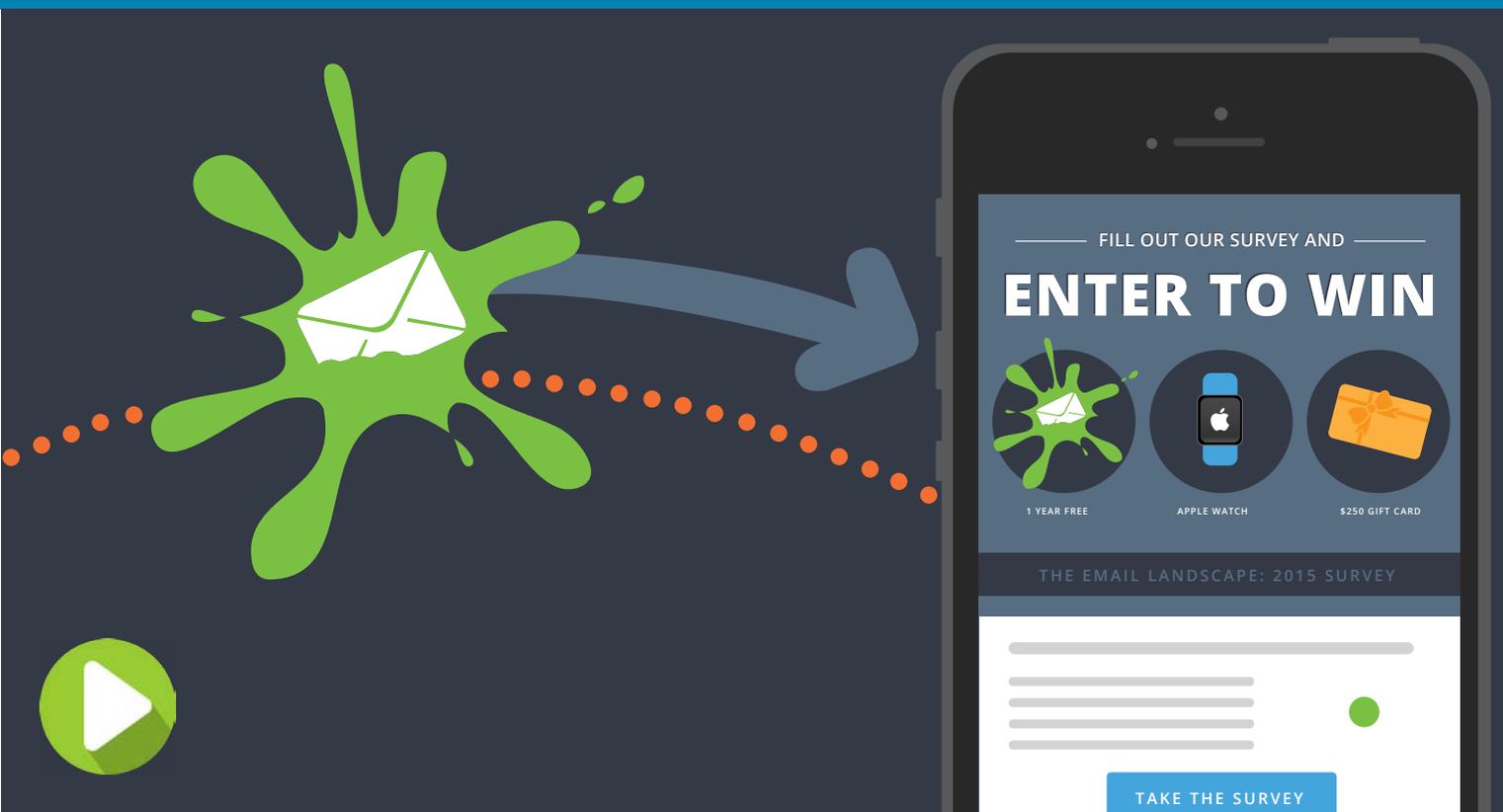
```
@keyframes animatedBackground {
0% { background-position: 650px 0; }
100% { background-position: 0 0; }
}

@-moz-keyframes animatedBackground {
0% { background-position: 650px 0; }
100% { background-position: 0 0; }
}

@-webkit-keyframes animatedBackground {
0% { background-position: 650px 0; }
100% { background-position: 0 0; }
}
```

We're telling the background image to scroll to a pixel value that's either in a positive or negative direction from the starting point of the background.

We can modify this further by adding images/text on top of the scrolling background to create clever animations like display block did earlier in this guide.



# Code Drawbacks of CSS Email Support

Please note; although I've researched this guide fully, people are always stumbling across new techniques and code in email so if you believe any of this to be wrong or outdated, please reach out and let us know.

## REPEATING ANIMATIONS

One of the major drawbacks is the lack of email support for animation-iteration-count. This is the property that enables us to control the amount of times an animation loops for. This means we can either have an animation repeat once or forever. This is a pretty heavy drawback, but there's still lots of amazing animations you can do in emails (as seen in the examples above) with continuous looping.

## DELAYING ANIMATIONS

Another pretty annoying drawback is the lack of support for the animation-delay property. This is the CSS snippet that allows us to delay how long before an animation triggers and starts. This means that all of our animations will start as soon as the email loads. Again, annoying!

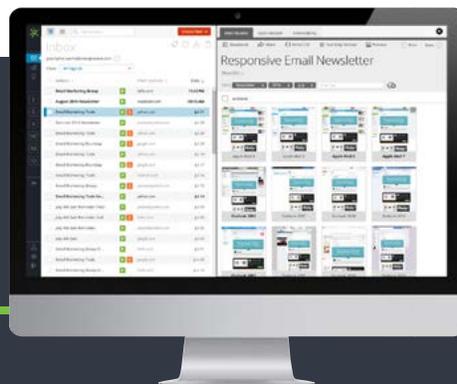
## SCROLL BASED/SCROLL TRIGGERED ANIMATION

I have a lot of people ask me about this but unfortunately, we can't do it. It's a technique we see a lot of on the web but it actually uses a sub-branch of JavaScript, jQuery, to handle the scroll based triggers to then call in the CSS animations. It's a beautiful technique, but we all know the pitfalls of trying to get JavaScript to work in an email!



# Creating the Best Email Experience, in Every Client

CSS animations are a great way to enhance your emails, but this technique is often overlooked by developers, designer and marketers. A delicate balance must be achieved to ensure your email looks great in every inbox, while enabling advanced interaction functionality in the inboxes that support these enhancements. Give everyone the best possible experience in your emails, regardless of the email client, by vigorously testing your emails before you hit "send." Try us free for 7-days to get access to unlimited email, image and spam testing.



Protect your revenue and your bottom line by building and testing your email in the most popular inboxes.

TRY IT **FREE!**

