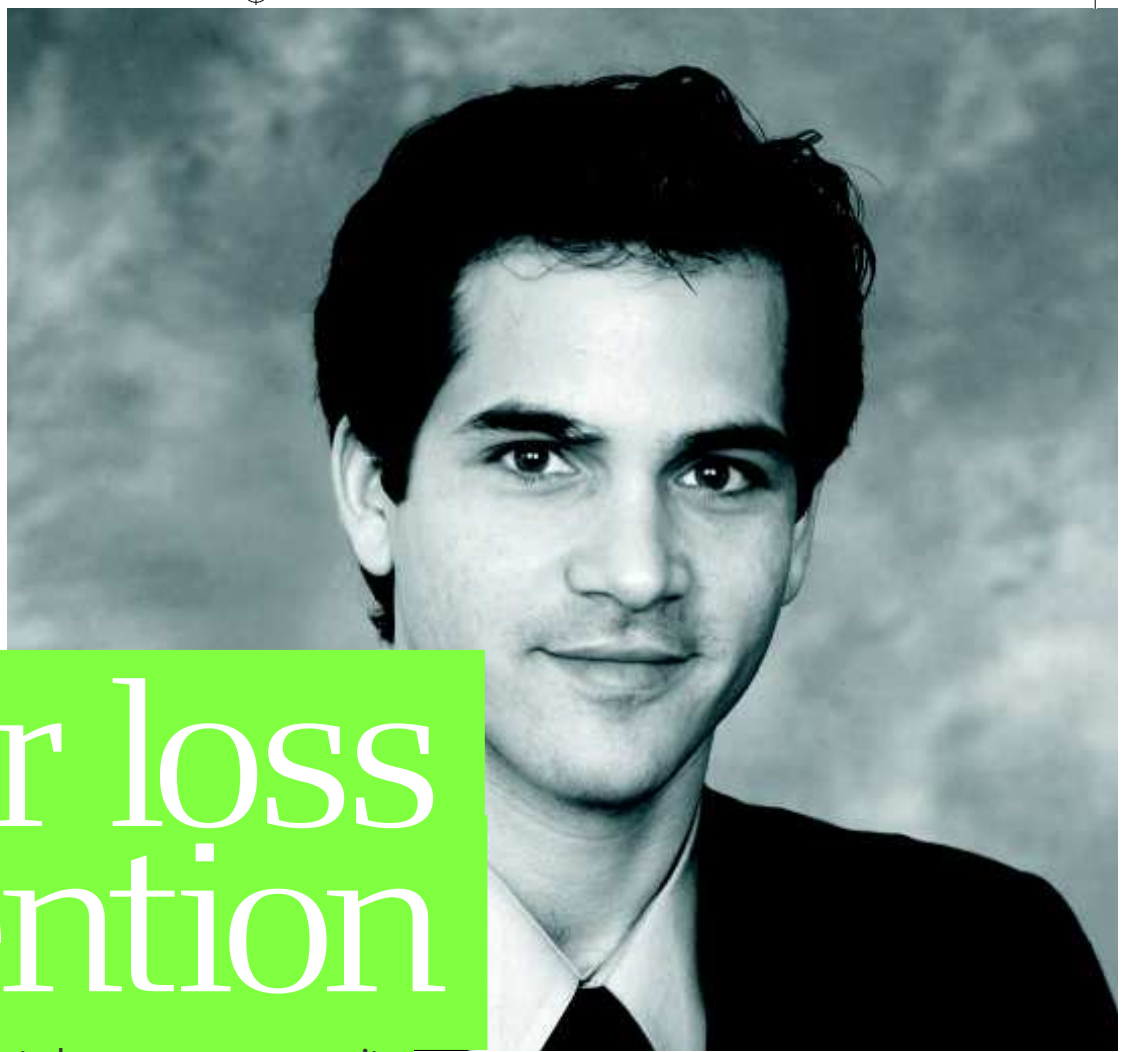


Anthony Calzadilla on...

Visitor loss prevention

Set up a time-delayed animation to keep users on your site, says Optimum7's Anthony Calzadilla



We're always trying to maximise the amount of time a visitor spends on our websites. Google Analytics shows us the average time someone spends on a page before they exit. With this information, we can create a time-delayed animation that will play after a specific period. Then it will kick into action to try to regain visitors' attention by using subtle animation to draw them back in.

To determine your top exit pages, log into Google Analytics. Find and click the **Content** link on the left-hand navigation. A sub-menu drops down. Click the **Top Exit Pages** link. A list is displayed of the pages with the highest number of visitors exiting. Click the first one on the list to drill down into the details. Take note of the average **time on page**. This will give you an idea of how long visitors are on this specific page before they leave.

In this case, our objective is to have the visitors click our **Get a Quote** button. We'll create an animation to entice users to interact with it. We'll apply a timed delay to the animation, because we don't want to interfere with them – we just want to try to reel them back in before they jump ship.

To persuade our users to stay and click on our button, we'll brighten the background colour and enlarge it by 200 per cent. That will surely cause the visitor to investigate. Here's the CSS3 version of a timed delay animation. First, we'll set up the basic styles of the button:

```
#call_to_action {
width: 300px;
height:150px;
```

```
font-size: 1em;
text-align:center;
}
```

Next, set up the keyframe animation named **call_to_action**, which will scale the button to 200 per cent. The background colour will fade from beige to yellow during the animation.

```
@-webkit-keyframes call_to_action {
0% {-webkit-transform: scale(1); background-color:beige;}
100% {-webkit-transform: scale(2); background-color:yellow;}
}
```

Now we fire off the animation. Delay the animation 45 seconds, run the **call_to_action** animation infinitely with easing.

```
#call_to_action {
-webkit-animation-delay:45s;
-webkit-animation-name: call_to_action;
-webkit-animation-duration: 5s;
-webkit-animation-direction:alternate;
-webkit-animation-iteration-count: infinite;
-webkit-animation-timing-function: ease-in-out;
}
```

Unfortunately, this CSS3 version only works in Chrome and Safari at the moment, meaning that the majority of web users won't see the animation. But fortunately, the same functionality can easily be replicated using jQuery:

```
// When the browser and its contents finish
loading, start the delay timer
$(window).load(function () {
// In 45 seconds start the animation
$.doTimeout( 45000, function(){
// increase the width / height of button and
change the background colour to yellow
$('#call_to_action').animate({
width:600px,
height:300px,
backgroundColor: "yellow"
}, 5000);
});
});
```

To make the timed delay animation even easier here, we've used the **doTimeout** jQuery plug-in. Doing so makes setting up **setInterval** functions a breeze. ●

Anthony Calzadilla is creative director of internet marketing company Optimum7 (www.optimum7.com) and blogs at www.anthonycalzadilla.com

“It will kick into action and try to regain visitors' attention” Anthony Calzadilla