LAYOUT
by Flywheel

HOW TO

How to create SVG animations with CSS

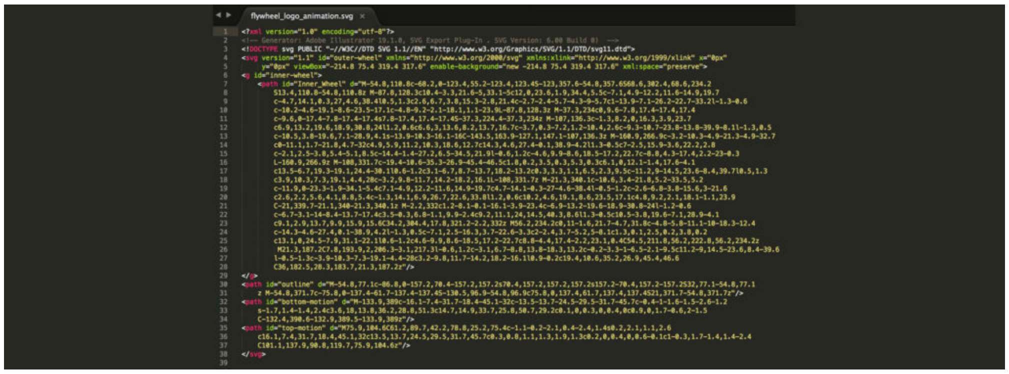
June 1, 2017 by Abbey Fitzgerald

Web animation is all the rage these days, and there's no limit to what you can design with SVGs. If you're ready to try it out on your own site, knowing basic animation concepts will help you build out more complex designs. SVGs are great for animation on the web because they're scalable and resolution independent (meaning they can be scaled without losing quality), unlike JPEGs and PNG files. They are also easy to style because you can use CSS, filters, and interactivity to enhance SVGs.

Before we dive into the tutorial, you should be familiar with CSS transitions and transforms. (If you're not, check out this article!) Some of the same concepts will be used with SVGs.

SVG BASICS

SVG, or Scalable Vector Graphics, use an XML-based text format to describe how the image should appear. SVGs are text files with a bunch of XML inside. If you open one with a code editor, you'll see what I mean. The reason SVGs can be scaled to different sizes without losing quality is because text is used to describe the graphic.



```
<?xml version="1.0" encoding="utf-8"?>
<!-- Generator: Adobe Illustrator 19.1.0, SVG Export Plug-In . SVG Version: 6.00 Build 0)  -->
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg version="1.1" id="outer-wheel" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px"
     y="0px" viewBox="-214.8 75.4 319.4 317.6" enable-background="new -214.8 75.4 319.4 317.6" xml:space="preserve">
<g id="inner-wheel">
    <path id="Inner_Wheel" d="M-54.8,110.8c-68.2,0-123.4,55.2-123.4,123.4S-123,357.6-54.8,357.6S68.6,302.4,68.6,234.2
        S13.4,110.8-54.8,110.8z M-87.8,128.3c10.4-3.3,21.6-5,33.1-5c12,0,23.6,1.9,34.4,5.5c-7.1,4.9-12.2,11.6-14.9,19.7
        c-4.7,14.1,0.3,27,4.6,38.4l0.5,1.3c2.6,6.7,3.8,15.3-2.8,21.4c-2.7-2.4-5.7-4.3-9-5.7c1-13.9-7.1-26.2-22.7-33.2l-1.3-0.6
        c-10.2-4.6-19.1-8.6-23.5-17.1c-4.8-9.2-2.1-18.1,1.1-23.9L-87.8,128.3z M-37.3,234c0,9.6-7.8,17.4-17.4,17.4
        c-9.6,0-17.4-7.8-17.4-17.4s7.8-17.4,17.4-17.4S-37.3,224.4-37.3,234z M-107,136.3c-1.3,8.2,0,16.3,3.9,23.7
        c6.9,13.2,19.6,18.9,30.8,24l1.2,0.6c6.6,3,13.6,8.2,13.7,16.7c-3.7,0.3-7.2,1.2-10.4,2.6c-9.3-10.7-23.8-13.8-39.9-8.1l-1.3,0.5
        c-10.5,3.8-19.6,7.1-28.9,4.1s-13.9-10.3-16.1-16C-143.5,163.9-127.1,147.1-107,136.3z M-160.9,266.9c-3.2-10.3-4.9-21.3-4.9-32.7
        c0-11.1,1.7-21.8,4.7-32c4.9,5.9,11.2,10.3,18.6,12.7c14.3,4.6,27.4-0.1,38.9-4.2l1.3-0.5c7-2.5,15.9-3.6,22.2,2.8
        c-2.1,2.5-3.8,5.4-5.1,8.5c-14.4-1.4-27.2,6.5-34.5,21.9l-0.6,1.2c-4.6,9.9-8.6,18.5-17.2,22.7c-8.8,4.3-17.4,2.2-23-0.3
        L-160.9,266.9z M-108,331.7c-19.4-10.6-35.3-26.9-45.4-46.5c1.8,0.2,3.5,0.3,5.3,0.3c6.1,0,12.1-1.4,17.6-4.1
        c13.5-6.7,19.3-19.1,24.4-30.1l0.6-1.2c3.1-6.7,8.7-13.7,18.2-13.2c0.3,3.3,1.1,6.5,2.3,9.5c-11.2,9-14.5,23.6-8.4,39.7l0.5,1.3
        c3.9,10.3,7.3,19.1,4.4,28c-3.2,9.8-11.7,14.2-18.2,16.1L-108,331.7z M-21.3,340.1c-10.6,3.4-21.8,5.2-33.5,5.2
        c-11.9,0-23.3-1.9-34.1-5.4c7.1-4.9,12.2-11.6,14.9-19.7c4.7-14.1-0.3-27-4.6-38.4l-0.5-1.2c-2.6-6.8-3.8-15.6,3-21.6
        c2.6,2.2,5.6,4.1,8.8,5.4c-1.3,14.1,6.9,26.7,22.6,33.8l1.2,0.6c10.2,4.6,19.1,8.6,23.5,17.1c4.8,9.2,2.1,18.1-1.1,23.9
        C-21,339.7-21.1,340-21.3,340.1z M-2.2,332c1.2-8.1-0.1-16.1-3.9-23.4c-6.9-13.2-19.6-18.9-30.8-24l-1.2-0.6
        c-6.7-3.1-14-8.4-13.7-17.4c3.5-0.3,6.8-1.1,9.9-2.4c9.2,11.1,24,14.5,40.3,8.6l1.3-0.5c10.5-3.8,19.6-7.1,28.9-4.1
        c9.1,2.9,13.7,9.9,15.9,15.6C34.2,304.4,17.8,321.2-2.2,332z M56.2,234.2c0,11-1.6,21.7-4.7,31.8c-4.8-5.8-11.1-10-18.3-12.4
        c-14.3-4.6-27.4,0.1-38.9,4.2l-1.3,0.5c-7.1,2.5-16.3,3.7-22.6-3.3c2-2.4,3.7-5.2,5-8.1c1.3,0.1,2.5,0.2,3.8,0.2
        c13.1,0,24.5-7.9,31.1-22.1l0.6-1.2c4.6-9.9,8.6-18.5,17.2-22.7c8.8-4.4,17.4-2.2,23.1,0.4C54.5,211.8,56.2,222.8,56.2,234.2z
         M21.3,187.2C7.8,193.9,2,206.3-3.1,217.3l-0.6,1.2c-3.1,6.7-8.8,13.8-18.3,13.2c-0.2-3.3-1-6.5-2.1-9.5c11.2-9,14.5-23.6,8.4-39.6
        l-0.5-1.3c-3.9-10.3-7.3-19.1-4.4-28c3.2-9.8,11.7-14.2,18.2-16.1l0.9-0.2c19.4,10.6,35.2,26.9,45.4,46.6
        C36,182.5,28.3,183.7,21.3,187.2z"/>
</g>
<path id="outline" d="M-54.8,77.1c-86.8,0-157.2,70.4-157.2,157.2s70.4,157.2,157.2,157.2s157.2-70.4,157.2-157.2S32,77.1-54.8,77.1
    z M-54.8,371.7c-75.8,0-137.4-61.7-137.4-137.4S-130.5,96.9-54.8,96.9c75.8,0,137.4,61.7,137.4,137.4S21,371.7-54.8,371.7z"/>
<path id="bottom-motion" d="M-133.9,389c-16.1-7.4-31.7-18.4-45.1-32c-13.5-13.7-24.5-29.5-31.7-45.7c-0.4-1-1.6-1.5-2.6-1.2
    s-1.7,1.4-1.4,2.4c3.6,18,13.8,36.2,28.8,51.3c14.7,14.9,33.7,25.8,50.7,29.2c0.1,0,0.3,0,0.4,0c0.9,0,1.7-0.6,2-1.5
    C-132.4,390.6-132.9,389.5-133.9,389z"/>
<path id="top-motion" d="M75.9,104.6C61.2,89.7,42.2,78.8,25.2,75.4c-1.1-0.2-2.1,0.4-2.4,1.4s0.2,2.1,1.1,2.6
    c16.1,7.4,31.7,18.4,45.1,32c13.5,13.7,24.5,29.5,31.7,45.7c0.3,0.8,1.1,1.3,1.9,1.3c0.2,0,0.4,0,0.6-0.1c1-0.3,1.7-1.4,1.4-2.4
    C101.1,137.9,90.8,119.7,75.9,104.6z"/>
</svg>
```

Visual information in an SVG file is calculated and rendered by the browser or whatever graphics software you are using to modify it. They aren't as complex as a JPG or other image file formats, which allows them to be designed and edited manually in a text editor, if you choose to do so. They can be altered and styled with CSS, making them awesome to design with on the web.

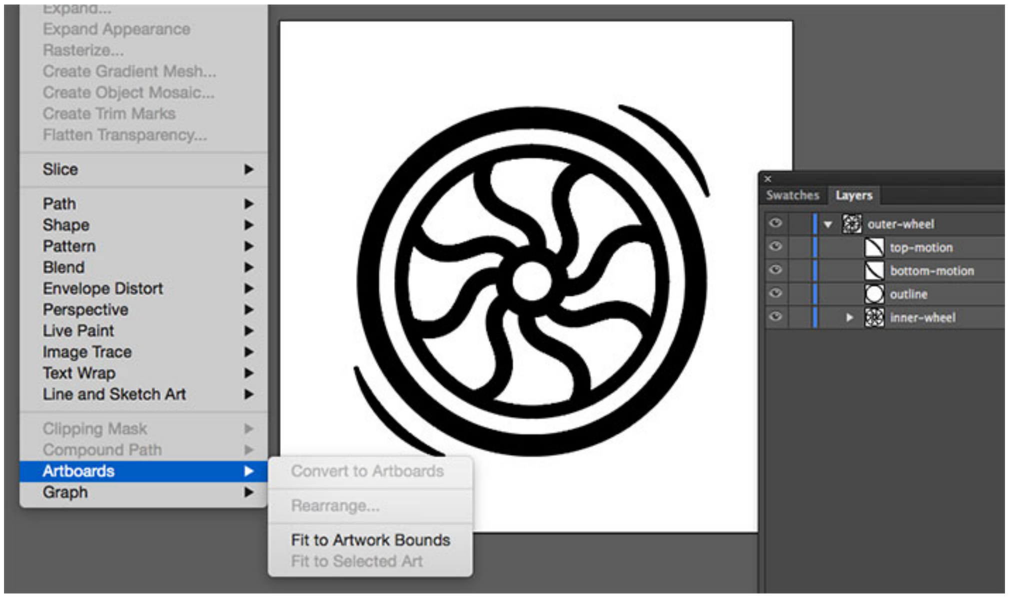
Browser support for SVG graphics is pretty universal; IE8 may give some issues, but modern browsers handle SVGs wonderfully. There may be minor bugs here or there, but typically it is smooth sailing.

HOW TO CREATE AN SVG GRAPHIC

Adobe Illustrator is my program of choice when creating SVGs. In all reality, they can be created from scratch since it's all XML, but it's probably easier to use a design program, especially for more complex SVGs.

TIPS WHEN DESIGNING IN ADOBE ILLUSTRATOR

Like with any other Adobe Illustrator project, the bounding box is where the illustration is created, so it's important to make sure everything is contained. The best way to do that is to select the artwork you want to have in your SVG and then go to Object > Artboards > Fit to Artwork Bounds.



This is a very important step. There have been a few occasions where I have not done this, and the image looked very small on the site because of the extra white space. Making everything fit perfectly in bounds solved the problem.

It may not be the most fun thing to do when you are in design mode, but being mindful and consistent with naming of layers will help you when you are doing future animation work with the graphic. "Layer One, Layer Two" will make it hard to remember what layers contained what part of the graphic. The reason for this is that Illustrator will use these names to generate the IDs in the SVG code.

Layer groups are awesome, especially for more complex graphics. They're also used to create groups in SVG files. When you're working with them, I highly recommend making sure the layer groups contain only related shapes, to keep your file clean and easy to edit later on.

In this file, you'll see the outer part of the wheel is named "outer-wheel" and the inner part has a name of "inner-wheel." It's basic, but it works.

This is optional, but running it through a CSS optimizer is helpful. There's a good chance that you can reduce the file size by doing this. One great option is SVG Optimizer by Peter Collingridge. SVG OMG is also an option worth looking into.

EXPORTING AN SVG GRAPHIC IN ADOBE ILLUSTRATOR

Now that you have your final SVG image, it needs to be exported so that it can be used on the web. Go to File > Save As > SVG. You may also go to File, Export, .SVG, depending on which version of Illustrator you are using. This will be saved as flyweel_wheel.

Once you do this, there will be a dialog box with a few options (if you don't see all of the following, go to "More Options"):

Profile: SVG 1.1

Type: This controls how you want to handle any fonts in your design and embeds fonts as SVG. We don't have any, so we'll leave this set as SVG.

Subsetting: This option embeds character details into the SVG file when needed. This allows the file to display fonts which may not be present on the user's system. It's possible to only include the glyph(s) used in the SVG (which reduces file size) when "Only Glyphs Used" (if you use a special font for your artwork) is selected.

Image Location: This controls the information that can be stored for the rasterized image data inside the SVG file as a data URI with the "Embedded" option. (Not super relevant in this case, but this allows for a link or embedded images, which would increase file size.)

CSS Properties: Presentation attributes allow for CSS styles to be put right in the SVG if needed. Depending on your use case, this may or may not be optimal. Presentation Attributes specifies things like `fill: blue;` rather than what is seen traditionally with inline styles: `style="fill: blue;"`. Presentation attributes are typically easier to override in CSS.

More Advanced Options: This group of checkboxes allows you to change a variety of settings, including how many decimal places you go to in the various numbers. One should be sufficient here. The advanced options are mostly necessary if there is a lot of text involved in your file. The option to output fewer `<tspan>` elements can dramatically reduce the size of your exported SVG.

In some cases, text might be drawn along a custom path. With the "Use <textPath> element for Text on path" option, it is exported as text on a path. The "Responsive" option is also important. If it is not checked, the SVG file will have a hard-coded width and height.

HOW TO ANIMATE SVGS WITH CSS

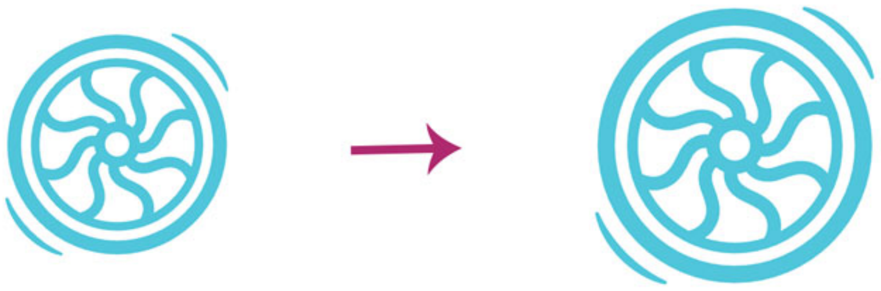
Now that there is an actual SVG to work with, we can create a simple animation to see how this all works. The CSS transform property and translate method will help the wheel move. Animating with CSS is awesome because there aren't any plugins or libraries that need to be installed; all you need is HTML and CSS to get started.

SVGs can be animated the same way that HTML elements can, using CSS keyframes and animation properties or using CSS transitions. More complex animations usually have some kind of transformation applied — a translation, rotation, scaling, or skewing.

BASIC ANIMATION

The following is a simple animation that makes the wheel grow on hover:

```
svg {
   height: 20%;
   width: 20%;
   fill: #50c6db;
}

svg:hover {
   transform: scale(1.25);
   transition-duration: 1.5s;
}
```



SPINNING WHEEL ANIMATION

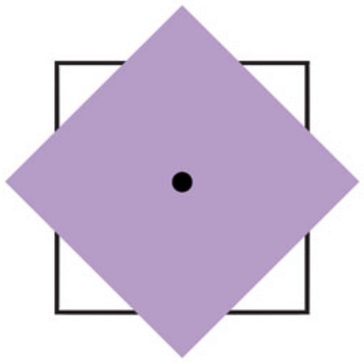
Here's where the hard work in Adobe Illustrator pays off. The effort of effectively naming the layers will be put to good use. With the outer and inner layer groups, animation can be controlled and customized, which is a huge advantage over animating raster graphics.

The wheel SVG will function as a loading spinner graphic. The outer wheel will spin and the inner will grow slightly as the color moves from light to dark.

SVG elements are pretty predictable, for the most part, but there are some positioning things that may be slightly more tricky. If you're used to other HTML elements, they respond to transform and transform-origin in the same. One thing to note is that they don't follow the box model, meaning margin, border, padding, etc. This makes positioning and transforming these elements slightly more challenging.

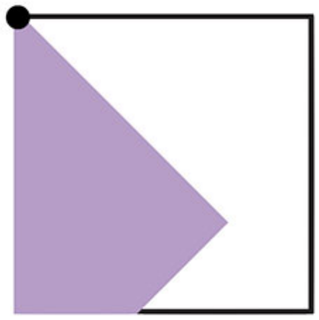
TRANSFORM-ORIGIN BASICS

The transform-origin property is used to change the position of the origin of transformation of an element. The transform origin of an HTML element is at (50%, 50%), which is the element's center.



This has been rotated 45 degrees with the transform origin of 50% and 50%.

An SVG element's transform origin is positioned at the (0, 0) point, which is the top-left corner of the canvas.



This has been rotated 45 degrees with the default transform origin of 0 and 0.

How does the SVG element rotate around its own center? An adjustment is needed with the the transform-origin property. This can be set using either a percentage value or an absolute value (pixels, ems, or rems). This value will be set relative to the element's bounding box.

If we were to set the transform origin of the `<rect>` to the center using percentage values, it would be done like this:

```
rect {
  transform-origin: 50% 50%;
}
```

MAKING THE WHEEL

First, getting the rotation keyframes set up is key. This will be a spinner graphic, so full rotation is needed. There will also be a fade-in effect applied as well. This example can be found on Codepen.

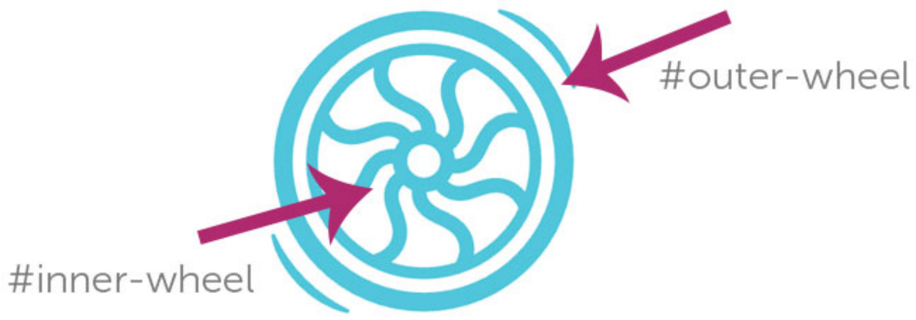
```
@keyframes spin {
    from {transform:rotate(0deg);}
    to {transform:rotate(360deg);}
}

@keyframes fadeIn {
    0% {
        opacity: .35;
    }

    50% {
        opacity: .5;
    }

    75% {
        opacity: .75;
    }

    100% {
        opacity: .25;
    }
}
```

Next, it's important to create an SVG wrapper.

```
.svg-container {
  height:100%;
  width:100%;
  max-height:15cm;
  max-width:15cm;
  margin: 0 auto;
}
```

General SVG styles were created and this is where the transform-origin is specified.

```
svg {
  top: 50%;
  left: 50%;
  position: absolute;
  max-height:15cm;
  max-width:15cm;
  width: 20%;
  height: 20%;
  fill: #50c6db;
  transform: translate(50%, 50%);
}
```



This is where the layer naming becomes very helpful. The spinning animation is applied to whole SVG graphic because it was specified on `#outer-wheel`, but the inner wheel has a fading effect that was only meant for that inner part of the graphic. By targeting only the `#inner-wheel`, the fade in animation was applied.

```
#outer-wheel {
  animation: spin 4s infinite linear;
}

#inner-wheel {
  animation: fadeIn 2s infinite linear;
}
```

Hopefully this gives you a good introduction to SVGs and basic animation techniques. As you use them more, these basic ideas will help you create more complex animations.

Want more design tips? Subscribe to Layout

We'll send the freshest design and WordPress content straight to your inbox, once a week.

Your email address

YES PLEASE!

7 COMMENTS

jake says:

June 8, 2017 at 11:08 am

hey how about after Fx.. i think that molight be he proper way to do it..

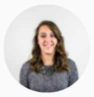
Reply

hanz says:

June 15, 2017 at 3:00 pm

I was expecting a relevant live example not a spinning gif :/

Reply



Morgan Ryan says:

June 15, 2017 at 3:24 pm

Hey Hanz, you can see the live example in the Codepen included in the article! Here's the link: https://codepen.io/abbeyjfitzgerald/pen/xddddL

Reply

pro pack apk says:

January 17, 2018 at 11:41 am

Thanks for another informative website.

Reply

revilodesign says:

February 14, 2018 at 8:36 am

Thanks for the instructions. You explained that very well

Thanks Oli

Reply

stuatrt says:

May 17, 2018 at 5:15 pm

this is a cool tutorial do you prefer to animate via css or js?

Reply

J3theWebDev says:

November 18, 2018 at 12:43 pm

Hello!

You should always aim to try an animate SVG's with CSS as it's significantly more stable, faster, and looks better :)

Plus it might actually be easier than you think!

Reply

JOIN THE DISCUSSION

Name

Email

Website

ADD COMMENT

SUBSCRIBE TO THE LAYOUT

Your email address

Top Posts

How to get started with Local by Flywheel for your WordPress development environment

30 awesome agency websites

How to make your WordPress site mobile friendly

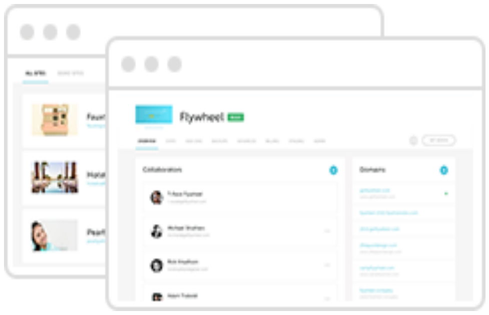
33 JavaScript libraries and frameworks to check out in 2018

The ultimate Instagram dimensions guideline for 2018 + free infographics and templates

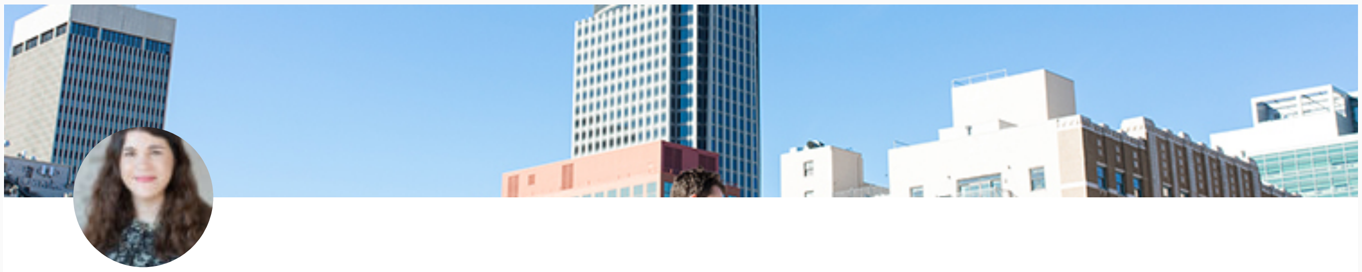
What's Flywheel?

Flywheel is a beautifully simple WordPress hosting and site management platform built for designers and creative agencies.

Learn more about the platform



More articles



How to add a responsive Google Map in WordPress

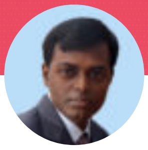
No matter where you're going, Google Maps will show you the way. Being able to instantly get directions has become essential while navigating through cities. And as the most popular map solution, adding Google Maps to your WordPress site could not be easier. This user-friendly mapping technology will provide business information, contact information, and directions...

Read full article

How to show different sidebars for posts and pages in WordPress

By default, WordPress shows the same sidebar on all posts and pages. But sometimes, you may want to display a completely different sidebar for your most important post or sidebar items related to an entire post category. If you've ever wondered how to do that, you're in luck! In this article, I'll show you how...

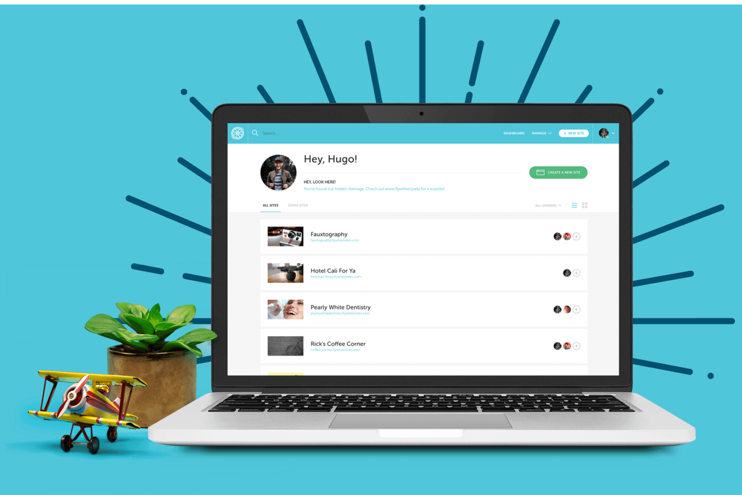
Read full article >



The beginner's guide to CSS specificity

Have you ever tried to override a CSS rule, only to find it not working as it should have? Or noticed when one element is targeted multiple times with different CSS selector combinations, only one rule is applied at a time? This is all because of CSS specificity rules. CSS specificity rules can be one...

Read full article >



FLYWHEEL

Delightful managed WordPress hosting

The #1 choice of over 70,000 digital creatives

TOUR THE PLATFORM

Follow Flywheel:

LAYOUT

Inspiration
WordPress
Business
How-to
Freebies

RESOURCES

Free ebooks
Discounts
Design Trends e-course

PRODUCTS

COMPANY

WordPress hosting
Reseller hosting
Enterprise hosting
Free local development

About
Careers
Life at Flywheel

FLYWHEEL