

# How to automate Photoshop with Photoshop Scripts

Today, this article will show you how to use Photoshop Scripts. It gives you the ability to change the behavior of a running script - something that Photoshop Actions can't do!

Photoshop is a great tool for image editing, so it's no wonder it is very popular. Previously, **Quantrimang** demonstrated how to design a website and how to sharpen images in Photoshop, so automation is the next logical step when you have mastered the basic concepts.

You may have heard about Photoshop Actions. It provides a simple way for you to automate tasks. Today, this article will show you how to use Photoshop Scripts. Photoshop Scripts are more powerful than Photoshop Actions. It gives you the ability to change the behavior of a running script - something that Photoshop Actions can't do!

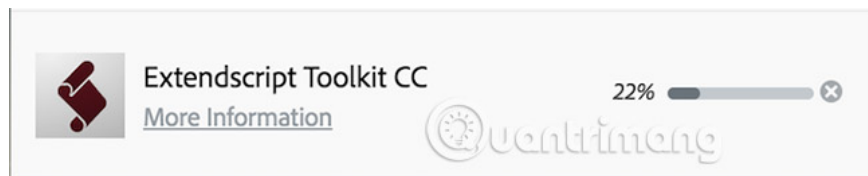
Don't worry if this sounds complicated, because scripts are written in JavaScript, a simple but powerful language.

## How to automate Photoshop with Photoshop Scripts

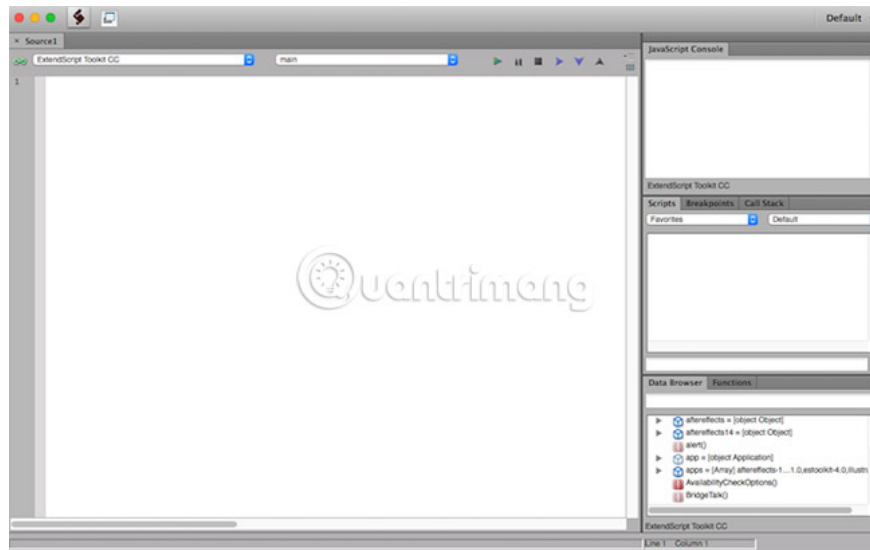
1. First script: Resize the image
2. Run the script
3. Make changing image sizes better

### First script: Resize the image

When writing scripts, you can use ExtendScript Toolkit. You are not required to use this tool, you can use any text editor you like (eg Sublime Text), but there are many benefits to using this toolkit. The biggest advantage is that you can set stops, which is useful when identifying any errors on the code and fixing it.



Open the toolkit after installation. You will see an interface as follows:



Here is the code for your first script - copy and paste this code into the main code window on the left:

```
current_document = app.activeDocument;
new_width = 670;

current_document.resizeImage (
UnitValue (new_width, 'px'),
null,
null,
ResampleMethod.BICUBIC
);
```

Let's see what this code does. The **current\_document** variable stores the active document object from Photoshop. This is accessed using the **app.activeDocument** syntax . If Photoshop is running but no image has been opened, this code will have an exception. Exception is a way for a code to stop executing - this script cannot continue if no image is opened!

Variable **new\_width** stores the width you want to apply to your new photo.

Finally, the **resizeImage** method changes the image size. This must be accessed via the **current\_document** variable . You need to enter a new width (converted into pixels through **BICUBIC's UnitValue**) and **ResampleMethod** methods.

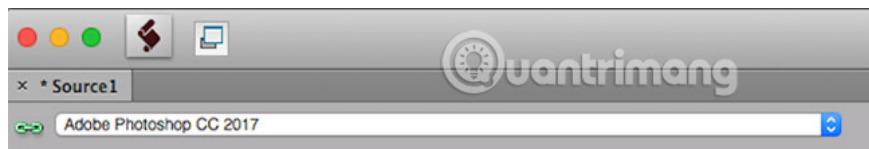
There are 5 main resample methods available to you. All are different in speed and quality, so you can experiment with each one (although Bicubic is often the best choice for most purposes). Here is a quick summary of the differences between methods:

1. **Nearest Neighbor** : Very fast but basic.
2. **Bilinear**: Better than Nearest Neighbor, but slower and not as good as Bicubic.
3. **Bicubic**: Very good results, but quite expensive.
4. **Bicubic Smoother**: An enhanced version of Bicubic is designed to enlarge images.
5. **Bicubic Sharper**: An enhanced version of Bicubic is designed to shrink images.

Remember to make use of them when used in your code.

Now you understand the code. It's time to run it! Make sure Photoshop is open with an image.

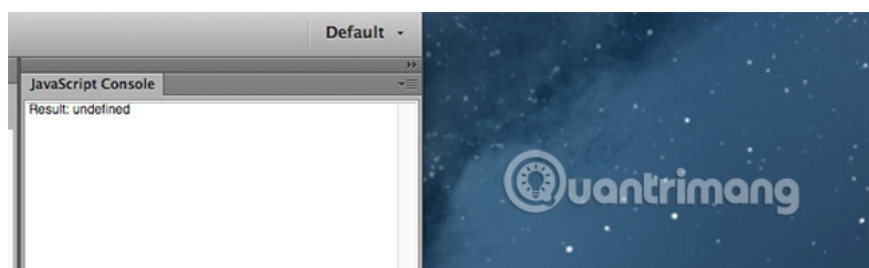
At the top of the ExtendScript Toolkit, change the destination menu (top left) from **ExtendScript Toolkit CC** to **Adobe Photoshop CC 2017** (or any specific version of Photoshop). If you are writing scripts for other Adobe applications, you can change this setting to alternative scripts.



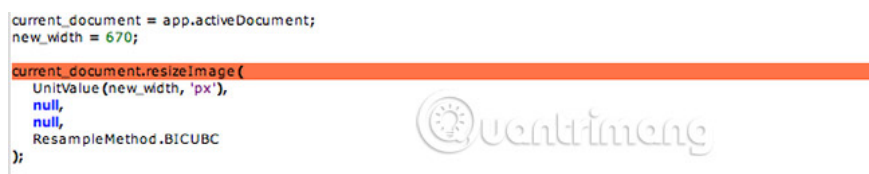
Now press the **Play** button to the right of the destination menu, at the top right of your code.



If everything runs correctly, your document is now resized (don't forget to save it). The JavaScript control panel at the top right of the toolkit will display the results of the script. In this case, the result will be **Result: Undefined**.



If there is a problem (such as an exception), your code will not run and you will get an orange line near the problem location.



This may be a simple typing error, so after stopping the script ( **Top Right Controls> Stop Button** ), double-check and make sure the following factors:

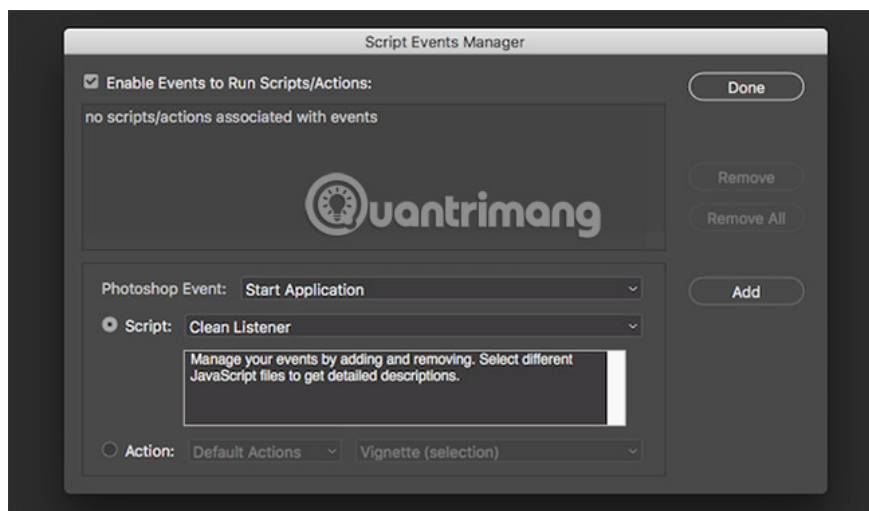
1. Your code is correct and has no spelling errors.
2. Photoshop is running.
3. You have an open image in Photoshop.

## Run the script

Now that your code is running correctly, it's time to add it to Photoshop.

Inside your toolkit, go to **File> Save** and save your script in the correct location and with an appropriate name. By default, this will be the Adobe Scripts folder. Note that the script must be in **.jsx** format .

Inside Photoshop, go to **File> Scripts> Script Events Manager** . Select **Enable Events To Run Scripts / Actions** .



When enabled, this interface allows you to configure existing scripts to run certain actions. There are several main areas, including:

1. **Photoshop Event** : This is the time when the script will run. You can choose from a variety of options, such as when printing, when opening new documents and many other options.
2. **Script**: This is the script to run. There are a few basic scripts that are built-in, but you can also assign one of your own scripts here.
3. **Action**: If you do not use the script, you can choose a basic action to perform the replacement, such as saving to PDF.

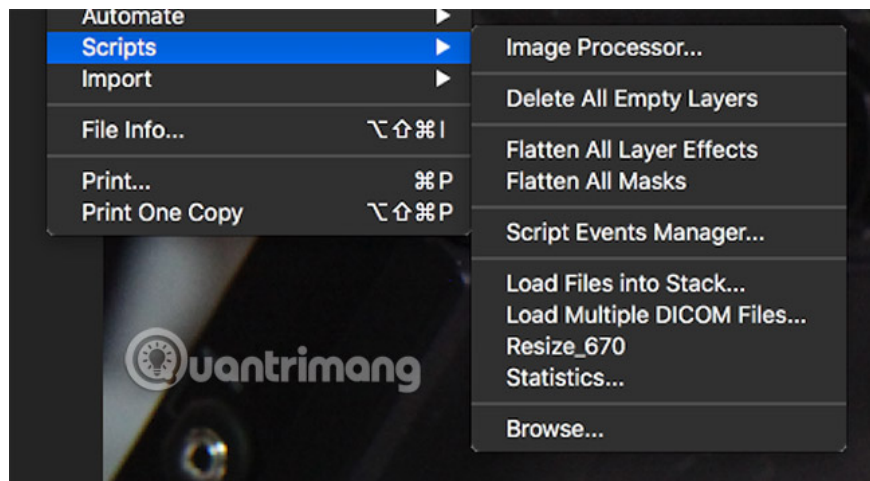
Select **Script**, then select **Browse**. Choose your script. Continue and choose a time to activate your script.

After setting, click **Add** and then **Done**. This menu is also where you can edit or delete any previously configured script.

If you don't want to combine your script with one action, it's even easier to set up. Go to **File> Scripts> Browse** . Navigate to your script and then click **Open**. Your script will run immediately.

If you want to view your script in this script menu, you need to copy that script to the appropriate directory. This can be found in the **Presets> Scripts** folder of Photoshop settings.

After copying, restart Photoshop. When restarted, your script will display as one in **File> Scripts Menu** .



## Make changing image sizes better

Now that you understand Photoshop Scripts, it's time to improve the code.

This code works very well when changing the image size to 670 px (or whatever size you have changed), but it can be even better. Fortunately, JavaScript is a very flexible language!

Please modify the code so that any image smaller than the new size will not be resized. This is the code:

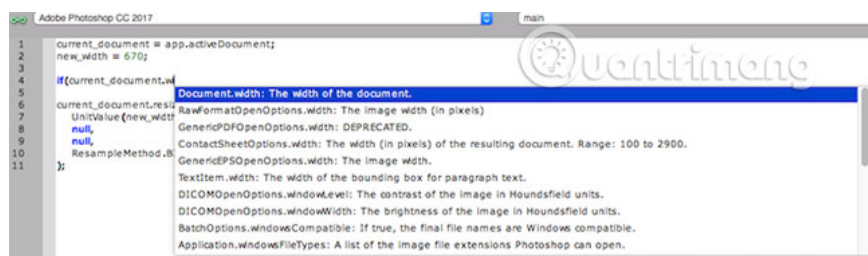
```
current_document = app.activeDocument;
new_width = 670;

if (current_document.width > new_width) {
current_document.resizeImage (
UnitValue (new_width, 'px'),
null,
null,
ResampleMethod.BICUBIC
);
}
```

There is only one change here. An if statement includes the entire image resizing method. If the current image width (accessed through **current\_document.width**) is smaller than the new width, do not change the size.

This ensures that small images are not enlarged and reduce image quality.

When you start entering code into the toolkit, it provides suggestions for valid changes you can make, call methods or properties to access. This is very useful, and you should make the most of it!



This is a final change:

```
current_document = app.activeDocument; // Get the current directory
new_width = 670; // the new width changes the size of the wall

if (current_document.width > new_width) {
// if the image is larger than the new size
current_document.resizeImage (
UnitValue (new_width, 'px'),
null,
null,
ResampleMethod.BICUBICSHARPER
);
}

current_document.activeLayer.autoContrast (); // Apply contrast
current_doc.activeLayer.applySharpen (); // Apply sharpen
```

This code now contains comments, which all code should have, and will help you in the future.

The resample method has changed to **Bicubic Sharper** - this produces slightly better results when reducing the image size.

Finally, contrast and sharpness have been applied in this final step.

Now, you know all you need to automate everything in Photoshop!

Of course, the examples in this article are just basic, but they include key details - you can deploy a specific script depending on your needs!

Have you learned anything new through this article? What is your favorite script? Let us know in the comment section below!

See more:

1. 5 basic photo editing tips on Adobe Photoshop Express
2. 3 tips to improve low-resolution image quality
3. In this way, resizing images in bulk will not take much effort as before

You finished reading the article "**How to automate Photoshop with Photoshop Scripts**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.