



## What are scripts in PaintShop Pro?

### Description

In a previous article ([5 reasons to add PSP to your graphic toolbox](#)), we mentioned that one great feature of PaintShop Pro was its scripting abilities. It can run scripts, it can record scripts, and a user can code scripts. But what are they? Many veteran PaintShop Pro users have never used scripts, either because they don't know about them, or because they are afraid of using them. So, let's shed some light on this fantastic feature.

## What are scripts anyways?

Scripting has been introduced to PaintShop Pro by Jasc with version 8. That means that most current users have the ability to run and record scripts with their current version. The word "script" is used a lot in programming and that word is not specific to PaintShop Pro. The simple definition of "script" is *a list of commands that are executed by a certain program*. It's that simple!

## What can scripts do?

A script can do just about anything you would do manually with your program. It can choose a color, flood fill, write text, draw shapes, make a selection, add a shadow, apply any effect that you would do yourself. But they can also do a lot more.

## Why use scripts?

Computers can usually work faster than we can, so using a script to execute a series of commands means it will be completed faster than if you did it manually. For example, a script can add a drop shadow to 50 separate layers in less than 15 seconds, while the same steps done manually might take several minutes. Scripts are also more precise and can include specific values (like move exactly 159 pixels to the right, or rotate by 36.9 degrees).

## What is a recorded script?

With the arrival of scripts, in version 8, also came the ability to record them. For example, if you need to convert a photo to sepia, and then add 40% noise, you would be able to record those steps, save the sequence as a script and reuse it later in one click instead of several. Some snippets can be added to give the user the option to choose a color or such, but the user input is usually pretty limited.

## Can I record scripts?

Yes. Since PaintShop Pro includes a function to record scripts, this is probably what you will do: if you want to draw a flower, you can start the recording, draw your flower and then save the steps you took for drawing that flower. Then, you can give that script to someone else who can run it and get exactly the same flower.

## Is it hard to record scripts?

Recording scripts is super easy. You click the **Record** button and then you do what you want the script to do, and finally, you save the script. If you want to repeat the exact same steps and those steps do NOT involve selecting an area, or a color, or clicking at a particular place on the image, or working with a particular size, then it would likely work. If you want, for example, your photo to be brightened by 20%, then add the **Time machine** effect and then turned to black and white, those commands would always work so you would be good to go. But beware if you want to draw a rectangle shape, or click inside a selection to change color, or such, then you would need to edit the script.

## Some scripts are not recorded?

Some scripts need extra commands to “think” like you do when you use the program yourself. For example, you can see the image you are working on, you can see where to click to flood fill inside a selection, you can see if the photo is horizontal or vertical and you can adjust your actions accordingly. A recorded script cannot do that because it just repeats the same steps over and over again no matter if the image is appropriate for it, or if the click is in the correct position.

## What are coded scripts?

In order to have a script work with different images, formats, and sizes, additional commands need to be added and those commands cannot be simply recorded. Functions, loops, mathematical calculations, string manipulations, conditionals and many more can be used. They require editing the

code itself by adding lines of codes that are not available while recording, so a good knowledge of coding language is mandatory. Of course, because of all those extra functions available, coded scripts can do much more advanced tasks and mostly, can incorporate user input and work with it.

## Is it hard to code scripts?

Coding script will often require some sort of a recording but since the same commands are often used over and over again, it is easy to copy and paste some snippets from one script to the other. However, there is much more than the simple PSP commands in a script. For example, you can have some calculations (ex: finding the total size of the image based on the starting photo, like in the [Stamps](#) script), loops to repeat the same steps several times (ex: creating paper cards, like in the [Pocket Scrap](#) script), adjusting the size of a font to fit inside a particular shape (ex: like with the [Alpha Beads](#) script), randomizing various settings to give different results every time (ex: combining colors and varying the material settings in the [Colored Pebbles](#) script), and more.

Most of those can be learned in the [Scripting course](#), available in the Campus. After the basics are learned, it is a matter of searching in Python references for new commands.

## My PSP is not in English, will scripts work?

Yes. Scripts are written in Python language and that is English based, even if your PSP is in a different language. Although the language of the coding is English, the message windows could be coded in any language the scripiter chooses. In my scripts, for example, I have a special code that will give you a French message if you are using a French version of PaintShop, and an English message in all other cases. Since I only know French and English, I could not code messages for other languages.

## Can scripts work on all PSP versions?

Scripting became a function of PSP at version 8. This is when a user was able to record and run scripts. This function has stayed with PSP ever since. However, scripts are based on the tools and functions used, and the Python version used by different PSP version. This means that, although most scripts will work in most PSP versions, it is definitely not an absolute, and you will find a lot of exceptions.

## What causes a script not to work in a particular PSP version?

Since scripts are a series of commands, the commands and tools have to be present in the particular PSP version it is run in. For example, if you record a script, in version X2, and it uses the **Pick** tool, it will automatically fail in PSP 9 simply because it does not have that command; back then, it was the **Raster Deform** tool. Lucky for most, if a script is coded with commands from an older version of PSP, the newer versions can still recognize them. For example, if the **Raster Deform** tool is used in a script, the newer versions (which have the **Pick** tool instead) will still be able to understand that command.

A good scripiter will know the differences between the tools and commands and also how they work. Some PSP versions have some particularities that didn't cause any issues before, or work in a different

way than previous versions. Just about every version has its particularities, and depending on the script in question, it might or might not affect its running. Unfortunately, too many people have limited knowledge of the scripting language and are not able to render their script compatible with previous versions so if you want to download or buy a script, make sure it is compatible with your particular version. If in doubt, check with the author.

There is still more to say about scripts and scripting. Check [this article](#) for more information on them. In the meantime, if you have any questions, just post them in the comment section below.

**Date Created**

June 26, 2014

**Author**

myclone