

MoviePy v2.0 have introduced breaking changes, see [Updating from v1.X to v2.X for more info.](#)



[Home](#) > [The MoviePy User Guide](#) > [Creating your own effects](#)

Creating your own effects

In addition to the existing effects already offered by MoviePy, we can create our own effects to modify a clip however we want.

Why creating your own effects?

For simple enough tasks, we've seen that we can [Modify a clip appearance and timing using filters](#). Though it might be enough for simple tasks, filters are kind of limited:

- They can only access frame and/or timepoint
- We cannot pass arguments to them
- They are hard to maintain and re-use

To allow for more complex and reusable clip modifications, we can create our own custom effects, that we will later apply with [`with_effects\(\)`](#).

For example, imagine we want to add a progress bar to a clip, to do so we will not only need the time and image of the current frame, but also the total duration of the clip. We will also probably want to be able to pass parameters to define the appearance of the progress bar, such as color or height. This is a perfect task for an effect!

Creating an effect

In MoviePy, effects are objects of type [`moviepy.Effect.Effect`](#), which is the base [`abstract class`](#) for all effects (kind of the same as [`Clip`](#) is the base for all [`VideoClip`](#) and [`AudioClip`](#)).

So, to create an effect, we will need to inherit the [`Effect`](#) class, and do two things:

[Skip to main content](#)

- Create an `__init__` method to be able to receive the parameters of our effect.
- Implement the inherited `apply()` method, which must take as an argument the clip we want to modify, and return the modified version.

In the end, your effect will probably use `time_transform()`, `image_transform()`, or `transform()` to really apply your modifications on the clip. The main difference is, because your filter will be a method or an anonymous function inside your effect class, you will be able to access all properties of your object from it!

So, let's see how we could create our progress bar effect:

```
"""Let's write a custom effect that will add a basic progress bar
at the bottom of our clip."""

from moviepy import VideoClip
from moviepy.decorators import requires_duration

# Here you see a decorator that will verify if our clip have a duration
# MoviePy offer a few of them that may come handy when writing your own effects
@requires_duration
def progress_bar(clip: VideoClip, color: tuple, height: int = 10):
    """
    Add a progress bar at the bottom of our clip

    Parameters
    -----

        color: Color of the bar as a RGB tuple
        height: The height of the bar in pixels. Default = 10
    """

    # Because we have define the filter func inside our global effect,
    # it have access to global effect scope and can use clip from inside filter
    def filter(get_frame, t):
        progression = t / clip.duration
        bar_width = int(progression * clip.w)

        # Showing a progress bar is just replacing bottom pixels
        # on some part of our frame
        frame = get_frame(t)
        frame[-height:, 0:bar_width] = color

    return clip.transform(filter, apply_to="mask")
```

[Skip to main content](#)

Note

When creating an effect, you frequently have to write boilerplate code for assigning properties on object initialization, `dataclasses` is a nice way to limit that.

If you want to create your own effects, in addition of this documentation we strongly encourage you to go and take a look at the existing ones (see `moviepy.video.fx` and `moviepy.audio.fx`) to see how they works and take inspiration.

Previous

◀ [Modifying clips and apply effects](#)

Next

[Compositing multiple clips](#) ▶

© Copyright 2024, Zulko - MIT.

Built with the PyData Sphinx Theme 0.13.0.

Created using Sphinx 6.2.1.