

Hello, and welcome to Halo!

In this short tutorial, we'll be going over how to run multiple Halo spinner objects chronologically and use them within the greater context of your Python program.

First, we're going to want to make sure we have Python 3 installed. Open up your command line, and enter the following code:

```
$ python --version
```

Make sure you see **Python 3.x.x**, where the x is any digit. This means you have some version of Python 3 installed on your machine. If you don't have Python 3, simply go to

Next, let's make sure we install Halo. In your command line, enter the following code:

```
$ pip install halo
```

Now we're ready to create our context manager script!

Navigate to where you want your file to be on your computer. In my case, this file will be in C:\Halo\examples.

Create a new Python file with the following command:

```
$ touch context_manager.py
```

Now open this file in your favorite text editor to begin coding!

At the top of the file, copy and paste the following code:

```
import os
import sys
import time
```

This imports the custom packages we need to run the spinners correctly. After these imports, add the following line of code:

```
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))
```

This will add our spinners to the terminal interface so we can see them there.

Next, let's go ahead and import the Halo package. Copy and paste in the following line:

```
from halo import Halo
```

Ok, now we're ready to create our spinners contextually! Copy in the following lines of code:

```
with Halo(text='Unpacking', spinner='dots', color='red'):
    # Run time consuming work here
    time.sleep(3)

with Halo(text='Adding presets', spinner='triangle', color='red'):
    # Run time consuming work here
    time.sleep(3)

with Halo(text='Configuring textures', spinner='bouncingBall', color='red'):
    # Run time consuming work here
    time.sleep(3)
```

In the code above, the `text` is what will be displayed in the terminal while the spinner is operating. The `spinner` parameter is the type of spinner being displayed, and the `color` is obviously the color of the spinner.

This code is creating a spinner object three times, and with each spinner it will run the code inside the scope until that code is finished, then the spinner disappears. This is why there is a comment to run time consuming work inside the loop, as the spinner will keep running until it is over. The sleep function is a place-saver that is allowing the spinner to run for a set 3 seconds to demonstrate it.

This context management within Halo is extremely powerful, as you can show the user that your program is loading or configuring and keep them updated as your Python program is running.

Ok, let's get this program running shall we? Go back to your command line, and in the directory where you have your Python file enter the following command:

```
$ python context_manager.py
```

And enjoy as you see your spinners run one after the other!