Hello, and welcome to Halo!

In this short tutorial, we'll be going over how to create and run your first custom spinner right in the command line.

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 first spinner!

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 custom_spinner.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:

```
from __future__ import unicode_literals
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 spinner to the terminal interface so we can see it 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 the actual spinner object! Copy in the following lines of code to get the framework for our first spinner object:

```
spinner = Halo(
    text='Custom Spinner',
    spinner={
        'interval': 100,
        'frames': ['-', '+', '*', '+', '-']
    }
)
```

In the code above, the `text` is what will be displayed in the terminal while the spinner is operating. The `frames` are the characters that will be cycled through to create the spinner, and the `interval` is how quickly the spinner cycles through the characters you've provided.

So now that we've got our spinner all set up, let's provide some code to make it actually run! Copy and paste the following code into your Python file below the spinner object:

```
try:
    spinner.start()
    time.sleep(5)
    spinner.succeed('It works!')
except (KeyboardInterrupt, SystemExit):
    spinner.stop()
```

Here the program is telling the computer to try to `start` the spinner, and we are telling it to `sleep` to allow the spinner to run for an allotted amount of time (in our case 5 seconds). Upon completion, we'll get a `success` message. If the spinner could not run, or a command is entered to stop it, the `except` keyword will catch it and `stop` the spinner.

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 custom_spinner.py
```

And enjoy as you see your spinner come to life!