It is very important to put your RPi in a case if you power on the RPi while holding it in your hand, because skin conducts electricity. Static electricity can potentially damage semiconductors on the board and kill the RPi.

You also need a microphone so that you can give commands to Melissa, who now resides in the RPi. Microphones that connect via a USB cable are available on Amazon; you can check for compatibility with the RPi in the product's description before purchasing it.

Another thing you need is either earphones or speakers to connect to your RPi so you can hear Melissa's responses to your queries.

Making Melissa Better Each Day!

This section talks about how you can add new and more complex functionalities to make Melissa better each day. The preferred way to do this is to fork the official repo, open an issue for discussion under new features (this will save you some time if someone else is already working on the issue and will let people decide if they want to help you on a complicated issue), create a new branch for working on your module, and start to work on it. After you have finished building your feature, open a pull request referencing the issue you created earlier. After testing your feature, you can merge it into the official codebase.

The goal is to obtain as many e-karma points as you can by contributing to Melissa's repository. Remember, 1 green box = 1 e-karma point. You can help improve this community-driven, completely open source initiative and make it one of the best virtual assistants in the world. Let's go through some sample features/subprojects that you can own and that would make a big impact on Melissa's functionality.

Windows Compatibility

Melissa currently supports only OS X and Linux systems. It would be great to include compatibility for Windows as well. That would require going through the entire install methodology in Melissa's repository and documenting changes. It would also require changing the code where the `sys.platform()` function is called and adding references to the Windows platform.

This is one of the easiest yet most important issues that needs to be taken care of in Melissa's repository. It would help you run Melissa on Windows IoT on your Raspberry Pi.

Tests

No code repository is truly professional unless it has tests built for it. Melissa currently does not have any tests for her health check. These tests would be run by contributors before submitting their pull requests. They might include checking for errors, trailing blank spaces, PEP-8 guidelines conformation, and much more.

You could make a package of tests by creating a `test` directory in the root folder of the project. Or perhaps a test package is not be a good idea at the beginning, but tests should be put in a different directory of their own.