**Effective Troubleshooting**

Note: This is a general guide to troubleshooting in general, not about a specific process or program. Most of the advice below is applicable to most processes or programs.

Let’s say you have a problem. You’re getting an error message when trying to carry about a process, or you can’t figure out how to set up a program to work the way you want, or your images don’t look as expected, etc.

**Here is a general order of resources to turn to for help**:

1. **Try to solve the problem yourself**.
   1. Before you even start turning to outside resources, try to solve the problem yourself. Just because you’ve received an unexpected outcome doesn’t mean you need outside help.
   2. Some common issues you can solve yourself include:
      1. Making sure all files and folders are where you expect them to be.
      2. Making sure all file paths you’re feeding into a process are correct.
      3. Making sure data copied correctly for one space to another.
      4. Making sure your code doesn’t have any syntax errors.
   3. Always make sure to check the small, simple things first.
2. **Look at the resources you already have access to**.
   1. Always start with what you have close at hand. This could be a software’s user guide, a paper that describes the pipeline you’re running, or even one of the documents within this documentation collective.
   2. If you can’t solve a problem yourself, you may already have a resource that can help.
3. **Read online guides and forums**.
   1. The internet will be a vast and often helpful resource for you.
   2. There are many blog entries, websites, and help guides online on a variety of topics – everything from MRI physics to bash syntax to image registration best practices to general computer issues.
   3. There are also help forums, which are a great resource for programming-centric problems, but other issues as well.
   4. Some helpful hints for effective web searching:
      1. If you are trying to solve an issue that involves an error message, search for the exact test of the error message.
      2. Be exact and specific.
         1. Instead of “how do I set my working directory in python” try “set working directory python3”
      3. If you don’t initially find what you need on your first search, try varying the detail and word usage in your searches.
         1. “printer not connecting to computer” maybe not be as effective as “Mac Mojave can’t connect to printer”.
      4. If you need with specific software, always try the manufacturer’s website first. Many developers also run help forums in addition to including helpful documentation on their websites.
         1. You can also try searching “[software/process name] forum”.
4. **Post on online forums**.
   1. You may encounter a scenario in which no resource can help you. This is rare but no unheard of. In this case, you may need to post a question on a forum. Here are some general guidelines for effective forum posting:
      1. Check the forum thoroughly to make sure your question has not already been asked.
         1. This is a kind of Step 0, but it is good practice to doublecheck that you are not asking a duplicate question.
      2. Give a concise but descriptive title.
         1. For example: “Help with matlab problem” is a bad title. It tells the possible respondents nothing about your issue except that it is related to Matlab.
         2. “How can change colors of histogram in Matlab?” is a much better title.
      3. Contextualize your problem.
         1. In the body of the post, describe what you were doing when the error occurred, what you expected to happen, and what actually happened.
      4. Add supplementary materials when necessary.
         1. Most forums allow for adding in-line code or attachments to posts. Problems often become difficult to describe without visual aids. When in doubt, use these.
      5. Be courteous.
         1. People are more apt to answer a post if they believe the poster appreciates and values their input. It’s always nice to be nice.
5. **Ask your colleagues**.
   1. Some scenarios facilitate asking colleagues for help, such as when the above steps return no answers or when your questions are about in-house software or lab SOPs
   2. Feel free to reach out to your team members in person, via email, or via MS Teams when need be. And remember that it helps to take a similar approach to the guidelines for posting on online forums.
6. **Reach out to authors or developers**.
   1. At times it may be necessary to reach out to the authors of a paper, the developers of a software, or the person/people at the heart of the process/software/model/etc for which you are troubleshooting. Don’t be afraid to reach out of this is the case.
   2. Most people are happy to answer your questions if they have the time and capacity to do so. Just remember that this avenue generally takes longer to resolve because individual researchers or small development teams tend to be busy. Be sure to be respectful of people’s time.