Hierarchy of Online Resources

While there are many online resources about programming, not all of the them are created equal. This list of resources is in approximate order of reliability.

1. **The Python Tutorial** - This section of the official documentation surveys Python's syntax and standard library. It uses examples, and is written using less technical language than the main documentation. Make sure you're reading the Python 3 version of the docs!

2. **The Python Language and Library References** - The Language Reference and Library Reference are more technical than the tutorial, but they are the definitive sources of truth. As you become increasingly acquainted with Python you should use these resources more and more.

3. **Third-Party Library Documentation** - Third-party libraries publish their documentation on their own websites, and often times at https://readthedocs.org/. You can judge the quality of a third-party library by the quality of its documentation. If the developers haven't found time to write good docs, they probably haven't found the time to polish their library either.

4. **The websites and blogs of prominent experts** - The previous resources are primary sources, meaning that they are documentation from the same people who wrote the code being documented. Primary sources are the most reliable. Secondary sources are also extremely valuable. The difficulty with secondary sources is determining the credibility of the source. The websites of authors like Doug Hellmann and developers like Eli Bendersky are excellent. The blog of an unknown author might be excellent, or it might be rubbish.

5. **StackOverflow** - This question and answer site has a good amount of traffic, so it's likely that someone has asked (and someone has answered) a related question before! However, answers are provided by volunteers and vary in quality. Always understand solutions before putting them into your program. One line answers without any explanation are dubious. This is a good place to find out more about your question or discover alternative search terms.

6. **Bug Trackers** - Sometimes you'll encounter a problem so rare, or so new, that no one has addressed it on StackOverflow. You might find a reference to your error in a bug report on GitHub for instance. These bug reports can be helpful, but you'll probably have to do some original engineering work to solve the problem.

7. **Random Web Forums** - Sometimes your search yields references to forums that haven't been active since 2004, or some similarly ancient time. If these are the only resources that address your problem, you should rethink how you're approaching your solution.