Learning Python beyond the basics involves diving into more specialized areas like web development, data analysis, machine learning, and more. You can become an intermediate or expert Python programmer by focusing on both the breadth and depth of the language and its ecosystem. Below are some resources categorized by skill level and domain.

**Intermediate Level:**

1. **Books:**
   * "Effective Python: 90 Specific Ways to Write Better Python" by Brett Slatkin
   * "Python Algorithms" by Magnus Lie Hetland
2. **Online Courses:**
   * Coursera: "Python 3 Programming Specialization" by the University of Michigan
   * Udemy: "Complete Python Developer in 2021: Zero to Mastery"
3. **Websites:**
   * [Real Python](https://realpython.com/)
   * [Geeks for Geeks Python Programming Language](https://www.geeksforgeeks.org/python-programming-language/)
4. **YouTube Channels:**
   * Corey Schafer's Python Tutorials
   * Sentdex for Python-based data science
5. **Practice:**
   * [LeetCode](https://leetcode.com/) (focus on Python solutions)
   * [Exercism Python Track](https://exercism.io/tracks/python/exercises)
6. **Projects:**
   * Contribute to open-source Python projects on GitHub
   * Build a web application using Flask or Django
   * Automate routine tasks using Python scripts

**Expert Level:**

1. **Books:**
   * "Fluent Python" by Luciano Ramalho
   * "Python Machine Learning" by Sebastian Raschka and Vahid Mirjalili
2. **Online Courses:**
   * Coursera: "Applied Data Science with Python Specialization" by the University of Michigan
   * Udacity: "Machine Learning Engineer Nanodegree"
3. **Papers:**
   * Reading research papers related to Python libraries and algorithms
4. **Websites:**
   * [The Python Software Foundation's Documentation](https://www.python.org/doc/)
   * [PEP Index](https://peps.python.org/)
5. **YouTube Channels:**
   * PyData
   * PyCon
6. **Advanced Projects:**
   * Build and publish your Python package
   * Contribute to high-profile Python libraries (like Pandas, NumPy, or TensorFlow)
7. **Practice:**
   * [Kaggle Competitions](https://www.kaggle.com/)
   * [Project Euler](https://projecteuler.net/) for algorithmic challenges
8. **Certifications:**
   * Microsoft’s Python certification
   * PCEP – Certified Entry-Level Python Programmer

**Domain-Specific Resources:**

1. **Web Development:**
   * Django for Beginners by William S. Vincent
   * Flask Web Development with Python Tutorial series on YouTube by Corey Schafer
2. **Data Science:**
   * "Data Science from Scratch" by Joel Grus
   * DataCamp’s Data Scientist with Python track
3. **Machine Learning:**
   * "Deep Learning" by Ian Goodfellow and Yoshua Bengio and Aaron Courville
   * Fast.ai courses on deep learning
4. **Automation:**
   * "Automate the Boring Stuff with Python" by Al Sweigart
   * Python Automation Cookbook by Jaime Buelta

By leveraging these resources, you can aim to become an intermediate or expert Python developer. Remember, consistency and hands-on practice are key to mastering Python.