

# Additional Python Resources

If you're having trouble with a particular concept or simply want to have access to more information, try one of the following links.

## DOCUMENTATION

- [Official Python 3 Documentation](#) - "official"/technical explanation of what a particular function/operator does, examples of correct syntax, what the various libraries are, etc.

## TEXTBOOKS/TUTORIALS

- [Dive Into Python](#) - another survey of Python syntax, datatypes, etc.
- [Think Python](#) by Allen Downey - a good general overview of the Python language. Includes exercises.
- [The Official Python Tutorial](#) - self-explanatory
- [Learn Python the Hard Way](#) - (note: for Python 2) another free online text
- [Reserved Keywords in Python](#) - don't use these as variable names
- [PEP 8](#) - Style Guide for Python Code - learn what is good and bad style in Python
- [CheckIO](#) - learn Python by exploring a game world
- [Invent with Python](#) - develop your Python skills by making games or hacking ciphers
- [Codecademy](#) - (note: for Python 2) learn Python by building web apps and manipulating data; interactive tutorial sequence
- [Python Tutor](#) - interactive tutorial sequence of exercises

## DEBUGGING

- [Python Tutor](#) - an excellent way to actually visualize how the interpreter actually reads and executes your code
- [DiffChecker](#) - compares two sets of text and shows you which lines are different
- [Debugging in Python](#) - steps you can take to try to debug your program

## SOFTWARE

- [Python Tools for Visual Studio](#) - Visual Studio plug-in enabling Python programming

## OTHER Q&A

- [Stack Overflow](#) - a large Q&A forum for programming concepts (not just Python). Try searching here before you post on the edX forum, and you may find that someone has already answered your question.

## MORE PRACTICE PROBLEMS

- [Python Challenge](#) - a series of puzzles you can try to test your Python abilities
- [Project Euler](#) - additional programming challenges you can try once your Python knowledge becomes stronger; problems are sorted by increasing difficulty
- [Coding Bat](#) - problems you can solve within an online interpreter
- [Codewars](#) - improve your skills by training on real code challenges

MIT OpenCourseWare  
<https://ocw.mit.edu>

6.0001 Introduction to Computer Science and Programming in Python  
Fall 2016

For information about citing these materials or our Terms of Use, visit: <https://ocw.mit.edu/terms>.