PyBites Cheat Sheet - *Tips to Become a Better Python Developer*

```
for i, tip in enumerate(pybites, 1):
```

BEST PRACTICE	FEATURED ON PYBITES
1. (Re)Read and Apply the PEP8 Style Guide. This leads to cleaner, more maintainable code, easier for others to read and contribute, and hey ... you don't get frowned upon.	- 5 min Guide to PEP8 - Indentation Tips, Naming Things
2. Isolate Your Environment. Install your packages in a virtual env. This keeps your work environment clean and dumping a pip freeze in a requirement.txt makes project dependencies transparent to other developers.	- The Beauty of Virtualenv
3. Don't Reinvent the Wheel. Time again we find good examples of Python's batteries included approach! You can rotate a string with deque.rotate, build powerful iterators with itertools, compare lists with difflib, have Pythonic containers using collections, etc. Studying the stdlib makes you a better Python developer.	- Collections.deque, Counter, itertools - difflib, shelve - PyPI has > 100K packages
4. Learn Important Concepts. OOP, the Python Data Model, iteration, generators, decorators, context managers.	- Iteration, OOP primer, Data Model
5. Learn from Common Python Mistakes. Errors should not be silenced, avoid from module import *, watch out for mutable default arguments, to name a few.	- Learning from Python Mistakes - Error Handling, Mutability
6. Write Pythonic Code. Some of this comes with experience, however a lot of these idioms are easy to grasp so you should form a habit of adding them to your toolkit, for example tuple unpacking, enumerate, zip, with, string formatting.	- Beautiful, idiomatic Python - String formatting, Dict Ordering
7. Learn from the Pros. "I know the basics, how do I get to the next level?" Glad you asked: we created a Resources list to go from Beginner to Pro. It made to the top of /r/LearnPython and was featured on the Python Bytes podcast :)	- From Beginner to Pro Resources
8. Keep up2date with New Trends. Python's ecosystem is fast-paced and ever changing. There is a lot of active development and we think Python3 is totally awesome :)	- 3.6 New Features - PyBites Weekly Twitter Digest
9. Always Be Coding part I) Challenge Yourself! We learned so much raising and solving code challenges, in just a few months we already learned to code a simple game, leverage the Twitter API, do a sentiment analysis and more.	- PyBites Code Challenges posts - PyBites Challenges on Github
10. Always Be Coding part II) Build Cool Stuff. No inspiration? Start scratching your own itch: what problems can you solve for yourself, friends or family? It's when you build concrete products that you get stuck, debug, read others source code, search StackOverflow a lot and get some DevOps skills deploying your code. Really no better way to learn!	- Create a Simple Web Scraper - How to build a Twitter Bot - How to Build a Simple API

--
If you liked and found these tips useful, let us know!
Follow us on Twitter, Github and/or Facebook, we'd love to hear from you.
Thanks for being part of our community. We look forward to learning and sharing more Python with you.

Keep Calm and Code in Python!

Brought to you by two Passionate Pythonistas,
Bob and Julian @ PyBites