1. **Libraries and Environment Overview**  
   Video link [https://vimeo.com/858795196](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzYxMDQ3MDk0LCJuYmYiOjE3NjEwNDcwOTQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJid2k0NW95Y2VqNWRyeTZrNWQxOCIsInRva2VuIjoiYndpNDVveWNlajVkcnk2azVkMTgiLCJzZW5kX2F0IjoxNzYxMDU4ODAwLCJlbWFpbF9pZCI6ODQ4MTQ0OCwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3ZpbWVvLmNvbS84NTg3OTUxOTY_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.noCP8inWc2fDjahtC1GLJW10e1Y9XqHijgw7PInFJjM)  
   Length is 2 minutes
2. **PYPI**  
   Video link [https://vimeo.com/858801119](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzYxMDQ3MDk0LCJuYmYiOjE3NjEwNDcwOTQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJid2k0NW95Y2VqNWRyeTZrNWQxOCIsInRva2VuIjoiYndpNDVveWNlajVkcnk2azVkMTgiLCJzZW5kX2F0IjoxNzYxMDU4ODAwLCJlbWFpbF9pZCI6ODQ4MTQ0OCwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3ZpbWVvLmNvbS84NTg4MDExMTk_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.BgHivhhlNKWO3v4DK5kAkM_gpkNPrVNK58mUD0kOxhg)  
   Length is 1 minute
3. **Importing Libraries**  
   Video link [https://vimeo.com/858803462](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzYxMDQ3MDk0LCJuYmYiOjE3NjEwNDcwOTQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJid2k0NW95Y2VqNWRyeTZrNWQxOCIsInRva2VuIjoiYndpNDVveWNlajVkcnk2azVkMTgiLCJzZW5kX2F0IjoxNzYxMDU4ODAwLCJlbWFpbF9pZCI6ODQ4MTQ0OCwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3ZpbWVvLmNvbS84NTg4MDM0NjI_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.a8gblewhRPmohesb0tENQodz1FqPXg-DG2m1uuSoaeU)  
   Length is 5 minutes
4. **Library Imports and Names**  
   Video link [https://vimeo.com/858809369](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzYxMDQ3MDk0LCJuYmYiOjE3NjEwNDcwOTQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJid2k0NW95Y2VqNWRyeTZrNWQxOCIsInRva2VuIjoiYndpNDVveWNlajVkcnk2azVkMTgiLCJzZW5kX2F0IjoxNzYxMDU4ODAwLCJlbWFpbF9pZCI6ODQ4MTQ0OCwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3ZpbWVvLmNvbS84NTg4MDkzNjk_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.ggcLpVYmtzII2FKYv4DGSsZ08d38tS4nZttyZzkZK1o)  
   Length is 5 minutes
5. **sys.path**  
   Video link [https://vimeo.com/858815036](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzYxMDQ3MDk0LCJuYmYiOjE3NjEwNDcwOTQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJid2k0NW95Y2VqNWRyeTZrNWQxOCIsInRva2VuIjoiYndpNDVveWNlajVkcnk2azVkMTgiLCJzZW5kX2F0IjoxNzYxMDU4ODAwLCJlbWFpbF9pZCI6ODQ4MTQ0OCwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3ZpbWVvLmNvbS84NTg4MTUwMzY_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.IV43NmwCyl9CnyXrBF2hsk5XG3epxLJ5MvXucL6cmGA)   
   Length is 6 minutes
6. **$PYTHONPATH**  
   Video link [https://vimeo.com/858818566](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzYxMDQ3MDk0LCJuYmYiOjE3NjEwNDcwOTQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJid2k0NW95Y2VqNWRyeTZrNWQxOCIsInRva2VuIjoiYndpNDVveWNlajVkcnk2azVkMTgiLCJzZW5kX2F0IjoxNzYxMDU4ODAwLCJlbWFpbF9pZCI6ODQ4MTQ0OCwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3ZpbWVvLmNvbS84NTg4MTg1NjY_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.zxiPeGYDqALjgDJpLOGKrMp_eUrDJXMuYi57stpAW9s)  
   Length is 4 minutes
7. **Virtual Environments**  
   Video link [https://vimeo.com/858844605](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzYxMDQ3MDk0LCJuYmYiOjE3NjEwNDcwOTQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJid2k0NW95Y2VqNWRyeTZrNWQxOCIsInRva2VuIjoiYndpNDVveWNlajVkcnk2azVkMTgiLCJzZW5kX2F0IjoxNzYxMDU4ODAwLCJlbWFpbF9pZCI6ODQ4MTQ0OCwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3ZpbWVvLmNvbS84NTg4NDQ2MDU_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.N_GPI9IUNawfd5FsbUo28eV9ALX1pkhHe3YNg_xXKKo)  
   Length is 9 minutes

**Collateral Material (programs used in the videos):**  
  
In collateral, I try to provide the Python programs shown in the videos. In other words, I try to make it easy for you to reproduce the examples from the videos.  
  
[Lesson9 Collateral Programs](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9..LeEHp1c7Oo8VhR592RZpYDloIyz68YURRruJ0cGBDEQ)  
  
  
  
  
**Additional Content:**  
  
**PYPI**  
  
[Python Package Index](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzYxMDQ3MDk0LCJuYmYiOjE3NjEwNDcwOTQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJid2k0NW95Y2VqNWRyeTZrNWQxOCIsInRva2VuIjoiYndpNDVveWNlajVkcnk2azVkMTgiLCJzZW5kX2F0IjoxNzYxMDU4ODAwLCJlbWFpbF9pZCI6ODQ4MTQ0OCwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3B5cGkub3JnLz9fX3M9bXpwbTZucTNjbXR6eDAzdGs1ZGcifQ.UGpwOdCOMY-nO59muYobOvnmE5q7U0bpqEHOz6OGCRE)  
  
  
**Library Imports**  
  
[The import Statement](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzYxMDQ3MDk0LCJuYmYiOjE3NjEwNDcwOTQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJid2k0NW95Y2VqNWRyeTZrNWQxOCIsInRva2VuIjoiYndpNDVveWNlajVkcnk2azVkMTgiLCJzZW5kX2F0IjoxNzYxMDU4ODAwLCJlbWFpbF9pZCI6ODQ4MTQ0OCwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3JlYWxweXRob24uY29tL2xlc3NvbnMvaW1wb3J0LXN0YXRlbWVudC8_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.uLmK2wXxHAAxMpUZwsWm7yrn5s7qv2-ouWl9_FJHrgc)  
Real Python article that details different aspects of library imports. The content is similar to the course videos. Briefly mentions Python namespaces.  
  
  
**sys.path and $PYTHONPATH**  
  
[The Module Search Path](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzYxMDQ3MDk0LCJuYmYiOjE3NjEwNDcwOTQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJid2k0NW95Y2VqNWRyeTZrNWQxOCIsInRva2VuIjoiYndpNDVveWNlajVkcnk2azVkMTgiLCJzZW5kX2F0IjoxNzYxMDU4ODAwLCJlbWFpbF9pZCI6ODQ4MTQ0OCwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3JlYWxweXRob24uY29tL2xlc3NvbnMvbW9kdWxlLXNlYXJjaC1wYXRoLz9fX3M9bXpwbTZucTNjbXR6eDAzdGs1ZGcifQ.jrq92Kdu0zWrKhTjdUAav93LwU1dZ_L3--N_p2KCjIA)  
Real Python article detailing different aspects of sys.path and $PYTHONPATH (mostly sys.path). One interesting point that the article mentions is that sys.path will first search in the script directory (and not in your current working directory). In most of my testing, the current working directory and the script directory are the same. Interestingly, when I tested this I saw different behavior between MacOS and Linux. For MacOS, it would behave in the way the Real Python article described (i.e. the directory containing the script would be added to sys.path). In Linux, it would first add the directory containing the script and then right after this specify the current working directory. Note, the article also shows a script that dynamically modifies sys.path. I would avoid doing this.  
  
  
**Virtual Environments**  
  
[Python Virtual Environments: A Primer](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzYxMDQ3MDk0LCJuYmYiOjE3NjEwNDcwOTQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJid2k0NW95Y2VqNWRyeTZrNWQxOCIsInRva2VuIjoiYndpNDVveWNlajVkcnk2azVkMTgiLCJzZW5kX2F0IjoxNzYxMDU4ODAwLCJlbWFpbF9pZCI6ODQ4MTQ0OCwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3JlYWxweXRob24uY29tL3B5dGhvbi12aXJ0dWFsLWVudmlyb25tZW50cy1hLXByaW1lci8_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.TQvTA0cRW0XkZgbIkV6shNIupdg3WHcFAVJ78k6DY0o)  
Real Python article that goes into a very, very deep dive into Virtual Environments. More details than you (probably) ever want are here.  
  
  
**PIP**  
  
[PIP User Guide](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzYxMDQ3MDk0LCJuYmYiOjE3NjEwNDcwOTQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJid2k0NW95Y2VqNWRyeTZrNWQxOCIsInRva2VuIjoiYndpNDVveWNlajVkcnk2azVkMTgiLCJzZW5kX2F0IjoxNzYxMDU4ODAwLCJlbWFpbF9pZCI6ODQ4MTQ0OCwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3BpcC5weXBhLmlvL2VuL3N0YWJsZS91c2VyX2d1aWRlLz9fX3M9bXpwbTZucTNjbXR6eDAzdGs1ZGcifQ.mS_yglX3O28SGw2ls-agNd5wkn-e--Wq-76TX_9M_Lk)  
  
[PIP VCS Support](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzYxMDQ3MDk0LCJuYmYiOjE3NjEwNDcwOTQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJid2k0NW95Y2VqNWRyeTZrNWQxOCIsInRva2VuIjoiYndpNDVveWNlajVkcnk2azVkMTgiLCJzZW5kX2F0IjoxNzYxMDU4ODAwLCJlbWFpbF9pZCI6ODQ4MTQ0OCwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3BpcC5weXBhLmlvL2VuL3N0YWJsZS90b3BpY3MvdmNzLXN1cHBvcnQvP19fcz1tenBtNm5xM2NtdHp4MDN0azVkZyJ9.-r8SldVEKbZc-cyBFj1cfK8TR60wasQgjXOP5mJa9z4)  
  
  
  
  
**Exercises**

Reference code for these exercises is posted on GitHub at:

[https://github.com/twin-bridges/learning\_python/tree/main/lesson9/exercises](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzYxMDQ3MDk0LCJuYmYiOjE3NjEwNDcwOTQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJid2k0NW95Y2VqNWRyeTZrNWQxOCIsInRva2VuIjoiYndpNDVveWNlajVkcnk2azVkMTgiLCJzZW5kX2F0IjoxNzYxMDU4ODAwLCJlbWFpbF9pZCI6ODQ4MTQ0OCwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL2dpdGh1Yi5jb20vdHdpbi1icmlkZ2VzL2xlYXJuaW5nX3B5dGhvbi90cmVlL21haW4vbGVzc29uOS9leGVyY2lzZXM_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.pN67f3BjQsPQHmxo4xjYvQ3MyqVqFEKAuSPswxCm_w8)  
  
  
1a. Create a short Python script where you "import sys" and then print out your sys.path variable. Manually inspect your sys.path output and determine which locations Python uses for libraries (on your system). Note, do not use "IPython" for this exercise (as it modifies the standard sys.path).  
  
1b. Modify your system's $PYTHONPATH environment variable to include the ["bin" directory](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzYxMDQ3MDk0LCJuYmYiOjE3NjEwNDcwOTQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJid2k0NW95Y2VqNWRyeTZrNWQxOCIsInRva2VuIjoiYndpNDVveWNlajVkcnk2azVkMTgiLCJzZW5kX2F0IjoxNzYxMDU4ODAwLCJlbWFpbF9pZCI6ODQ4MTQ0OCwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL2dpdGh1Yi5jb20vdHdpbi1icmlkZ2VzL2xlYXJuaW5nX3B5dGhvbi90cmVlL21haW4vYmluP19fcz1tenBtNm5xM2NtdHp4MDN0azVkZyJ9.WV1-CfKYjWI8eWBurTp8OMCpoSGyUmHjbEEp7Duq6uU) in the base of this course's repository. Modify the script from exercise1a such that you also "import my\_lib" which is a simple Python module (located [here](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzYxMDQ3MDk0LCJuYmYiOjE3NjEwNDcwOTQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJid2k0NW95Y2VqNWRyeTZrNWQxOCIsInRva2VuIjoiYndpNDVveWNlajVkcnk2azVkMTgiLCJzZW5kX2F0IjoxNzYxMDU4ODAwLCJlbWFpbF9pZCI6ODQ4MTQ0OCwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL2dpdGh1Yi5jb20vdHdpbi1icmlkZ2VzL2xlYXJuaW5nX3B5dGhvbi9ibG9iL21haW4vYmluL215X2xpYi5weT9fX3M9bXpwbTZucTNjbXR6eDAzdGs1ZGcifQ.IThc7mzUKcBumHkk5Ufl6csMKNzSe4Ugb81pTZg5do4)). Verify the import works properly and visually check that your printed sys.path includes the new directory. You should see "Hello world" on standard out (as my\_lib.py prints out this message when imported).

2a. Relying on the $PYTHONPATH export specified in exercise1b, create a script that imports "func1" and "func2" from "my\_lib2" ([my\_lib2](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzYxMDQ3MDk0LCJuYmYiOjE3NjEwNDcwOTQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJid2k0NW95Y2VqNWRyeTZrNWQxOCIsInRva2VuIjoiYndpNDVveWNlajVkcnk2azVkMTgiLCJzZW5kX2F0IjoxNzYxMDU4ODAwLCJlbWFpbF9pZCI6ODQ4MTQ0OCwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL2dpdGh1Yi5jb20vdHdpbi1icmlkZ2VzL2xlYXJuaW5nX3B5dGhvbi9ibG9iL21haW4vYmluL215X2xpYjIucHk_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.lntjR8eY-NdqRLDCIC3C17JGsNpkYgpr9QP1GpP1nWM) is also located in "{ github\_repo }/bin"). Note, your script should directly call func1() and func2() without needing to specify the library name in the function call. Make sure your script executes properly--func1() should print out "Hello" and func2() should print out "World".  
  
2b. Repeat exercise2a except your calls should now be of the form my\_lib2.func1() and my\_lib2.func2(). Make sure your script executes properly.  
  
2c. Repeat exercise2b except use "as test\_lib" and change the name of library (in other words, instead of using my\_lib2.func1() and my\_lib2.func2() to call func1 and func2, your function calls should be test\_lib.func1() and test\_lib.func2(). Make sure your script executes properly.  
  
  
3a. Create a new virtual environment named "my\_venv". Pick a convenient directory on your system to contain this virtual environment (for example, ~/CODE or ~/VENV).  
  
3b. Activate your new virtual environment.  
  
3c. Execute "pip list" on the virtual environment and verify that you only have "pip" and "setuptools" installed.  
  
3d. Upgrade pip to the latest version of pip.  
  
3e. Re-run pip list and verify your version of pip has been upgraded.  
  
3f. Use pip to install the rich library. Use pip list to see which libraries were installed in addition to rich.  
  
3g. Execute "pip freeze" and look at the pip freeze output.  
  
3h. Deactivate your virtual environment.  
  
  
4a. Using your my\_venv from exercise3, pip install the [Netmiko library](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzYxMDQ3MDk0LCJuYmYiOjE3NjEwNDcwOTQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJid2k0NW95Y2VqNWRyeTZrNWQxOCIsInRva2VuIjoiYndpNDVveWNlajVkcnk2azVkMTgiLCJzZW5kX2F0IjoxNzYxMDU4ODAwLCJlbWFpbF9pZCI6ODQ4MTQ0OCwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL2dpdGh1Yi5jb20va3RieWVycy9uZXRtaWtvP19fcz1tenBtNm5xM2NtdHp4MDN0azVkZyJ9.AmcixL_M1k_QW5ZtyR8rCK854l_k-zjWkD03SsWhhqc" \t "_blank) from GitHub.  
  
4b. pip install the Netmiko master branch from github.  
  
4c. pip install the Netmiko 4.1.2 tag.  
  
4d. pip install the Netmiko code from Github using this specific commit: 679be2be58a975e874fd97616c7014f0726460c1  
  
4e. Use "pip freeze" and from this output create a requirements.txt file (using the current state of your environment from step 4d).  
  
  
5a. Create a new virtual environment named "test1\_venv". Activate this new virtual environment.  
  
5b. pip install all of the libraries using the requirements.txt file created in Exercise 4e.  
  
  
  
  
**CLASS OUTLINE**

1. **Libraries and Environment Overview**
   1. Libraries Overview [0:10]
      1. How do we find libraries that can assist us? [1:04]
2. **PYPI**
   1. What is PYPI ([pypi.org](http://pypi.org/))?  Python package index. [0:05]
   2. Library information: release history, home page, how active, and license information  [0:16]
3. **Importing Libraries - What Happens**
   1. How do we use these libraries in our code? [0:10]
      1. import ‘re’ statement [0:17]
   2. What happens when we import a library? [0:37]
      1. Python looks to see if the library is builtin [0:45]
      2. Python then looks in sys.path. sys.path is a list of locations Python tries to find the library in [1:00]
      3. Once found, Python will load the entire contents of the file, every line [2:02]
         1. Imported files may have other imports [2:18]
         2. Recursively resolving imports with a way to handle circular imports [3:35]
   3. Key Points [3:50]
4. **Library Imports and Names**
   1. “import re” will cause all names in the “re” library to be prefixed with “re”. [0:17]
   2. Alternatively, we can import using the “from” syntax [0:45]
      1. The names will all be imported directly into your program [1:04]
   3. Note, when using the  “from re import y” syntax, Python will still process the entire “re” library: line-by-line, top to bottom (including resolving imports recursively). This only changes the naming, not the import process [1:45]
   4. You can also change the names imported things are known as [2:35]
   5. You really should (almost) never do:  “from re import \*” [3:22]
      1. You don’t know if something in the library will trample names in your program [3:55]
      2. Makes it really hard when searching for names in your program for where they originated [4:05]
5. **sys.path**
   1. Python first search for “built-in” libraries and then will look for libraries using sys.path - a list of locations to check [0:44]
   2. Use the Python REPL, not “ipython” because it modifies sys.path [2:04]
   3. Example: re library [2:27]
   4. Example: math library [2:45]
   5. Example: “rich”, where libraries that pip installs would be located [3:08]
   6. What happens if we create a file named “math.py” in our current directory, our math.py file will be loaded instead of the normal library [3:40]
   7. We can modify sys.path and add additional locations to search [5:16]
6. **$PYTHONPATH**
   1. $PYTHONPATH is an environment variable that allows us to modify our sys.path [0:31]
      1. We can now import libraries that are in this directory [1:35]
   2. What about on Windows? We have sys.path as we would expect. [1:55]
      1. Look online for various ways in Windows to do this. [2:20]
   3. On Windows (and potentially MacOS), I would likely use an IDE (VSCode or PyCharm) [2:50]
      1. And there is a very good chance in your IDE that you can set the PYTHONPATH environment variable [3:00]
7. **Virtual Environments**
   1. Why? [1:00]
      1. We don’t really want to install/change libraries used by the operating system's Python [1:53]
      2. We don’t want operating system changes to alter our Python environment [2:33]
   2. How? [3:01]
      1. Pick a Python you want for the virtual environment [3:12]
      2. Execute the "venv" module [3:20]
      3. Name the virtual environment [3:25]
      4. Activate the virtual environment [3:53]
      5. Exit the virtual environment with deactivate [6:10]
   3. After you create a virtual environment - it will be essentially blank (as far as which libraries are installed) [6:22]
   4. How do I get these different Python versions installed in the first place? [6:52]
      1. MacOS/Windows - just install them [7:05]
      2. Linux - use system installer tools like apt and yum, or you can install from source code. [7:27]
      3. Another solution for installing different Python versions is pyenv [7:58]
   5. It is very common to have multiple virtual environments [9:05]