# Adding an Interface

# Also basic telnet connectivity

|  |
| --- |
| ###libraries to import **import** getpass **import** telnetlib   HOST="localhost" ## this is whatever IP address you identify user=input('enter your telnet username: ') ##defined on the router  password= getpass.getpass() # defined on the router   tn=telnetlib.Telnet(HOST) #variable created to reference telnetlib library   tn.read\_until(b"Username: ") ## use the b to turn the string into byte code  tn.write(user.encode('ascii') +b"\n") # this converts the username string into the ASCII text ( readable by the router ) and adds a line break  If password:  tn.read\_until(b"Password: ") # asks for password (just like telnet) and converts input into byte string  tn.write(password.encode('ascii' +b"\n") #same explanation as the username parameter  tn.write(b"enable\n") ## you have to enter everything into byte code , this brings you enable  tn.write(b"value \n" ### this is the password you defined to enter enable mode  tn.write(b"conf t \n") ##enter config mode  tn.write(b"int value \n" ## what ever interface you wanna work on  tn.write (b"ip address value \n")###whatever you desire to to enter  tn.write (b"router ospf (process ID) \n") ## enables ospf tn.write (b"network (network statement) \n") ## adds a network statement  tn.write (b"end\n") # breaks you out of config mode  tn.write (b"exit\n") ##exits out of the router  **print** (tn.read\_all().decode('ascii)) ## returns back to normal string |