

## Diff:

### Differences between given skeleton and solution

In order to make the sample solution easier to understand, the differences between it and the given skeleton source code were highlighted with the help of the program `diff`.

### Legend:

- Gray: unchanged text (only excerpts).
- Green: new lines
- Yellow: changed lines
- Red: deleted lines

Note: Files not listed have not been changed.

This document was created with the help of [diff2html](#) erstellt.

```
diff -u ../course13-communication-with-external-hardware/exercise/code/01_doc_strings.py ../course13-communication-with-external-hardware/exercise/solution/01_doc_strings.py
```

../course13-communication-with-external-hardware/exercise/code/01_doc_strings.py	../course13-communication-with-external-hardware/exercise/solution/01_doc_strings.py
1	1
2	2
3 """	3 """
4	4
5	5
6 from ipyindex import IPS	6 funcs = [bin, hex, oct, ord, chr, int]
7	7
8 funcs = [] ## ← insert functions here	8
9	9 for f in funcs:
10	10
11 for f in funcs:	11
12	12
13 # print name of function:	13
14 print(f.__name__)	14 print(f.__name__)
15	15
16 # print docstring	16 print(f.__doc__)
17 # ...	17
18	18
19	19
20 # print delimiter	20
21 print("-"*10, "\n")	21
22	22
23	23
24 ## optional: try out the function interactively	24
25	25
26 # IPS()	26

```
diff -u ../course13-communication-with-external-hardware/exercise/code/02_chat_client.py ../course13-communication-with-external-hardware/exercise/solution/02_chat_client.py
```

../course13-communication-with-external-hardware/exercise/code/02_chat_client.py	../course13-communication-with-external-hardware/exercise/solution/02_chat_client.py
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15 # convert unicode string to byte array (with utf8 encoding)	15 # convert unicode string to byte array (with utf8 encoding)
16 bin_msg = bytes(msg, "utf8")	16 bin_msg = bytes(msg, "utf8")
17 XXX() # send the data	17 s.send(bin_msg)
18 print("sent data:", bin_msg)	18 print("sent data:", bin_msg)
19	19
20 def receive():	20 def receive():
21 XXX = XXX(XXX)	21 data = s.recv(1024)
22 print("message from server", repr(XXX))	22 print("message from server", repr(data))
23	23
24 HOST = 'localhost' # The 'remote' host	24 HOST = 'localhost' # The 'remote' host
25 PORT = 50007 # The same port as used by the server	25 PORT = 50007 # The same port as used by the server

```
diff -u ../course13-communication-with-external-hardware/exercise/code/03_light_on.py ../course13-communication-with-external-hardware/exercise/solution/03_light_on.py
```

../course13-communication-with-external-hardware/exercise/code/03_light_on.py	../course13-communication-with-external-hardware/exercise/solution/03_light_on.py
1	1
2	2
3	3
4	4
5 # Create instance of the corresponding class (pass interface)	5 # Create instance of the corresponding class (pass interface)
6 # port name: e.g. "COM4" on Windows, "/dev/ttyUSB0" on Unix	6 # port name: e.g. "COM4" on Windows, "/dev/ttyUSB0" on Unix
7 #AC = ArduinoCommunicator(...)	7 #AC = ArduinoCommunicator("COM4")
8	8 AC = ArduinoCommunicator("/dev/ttyUSB0")
9	9
10	10
11 # call the appropriate method (check the source code of the class)	11 # call the appropriate method (check the source code of the class)
12 #AC.???	12 AC.light_on(30)
13	13
14	14
15 # start interactive shell	15 # start interactive shell

```
diff -u ../course13-communication-with-external-hardware/exercise/code/03_robot.py ../course13-communication-with-external-hardware/exercise/solution/03_robot.py
```

../course13-communication-with-external-hardware/exercise/code/03_robot.py	../course13-communication-with-external-hardware/exercise/solution/03_robot.py
4	4
5 # Create instance of the corresponding class (pass interface)	5 # Create instance of the corresponding class (pass interface)
6 # port name: e.g. "COM4" on Windows, "/dev/ttyUSB0" on Unix	6 # port name: e.g. "COM4" on Windows, "/dev/ttyUSB0" on Unix
7 RC = RobotCommunicator('COM4')	7 #RC = RobotCommunicator('COM4')
8	8 RC = RobotCommunicator("/dev/ttyUSB0")
9	9
10	10
11 # call the appropriate methods (check the source code of the class)	11 # call the appropriate methods (check the source code of the class)
12 # drive forward	12 RC.forward(200)
13 # make sound	13 RC.sound()
14 # read and print analog value	14 res = RC.read_analog()
15 # drive backward	15 print(res)
	16 RC.backward(200)
16	17
17	18
18 # start interactive shell	19 # start interactive shell
diff -u ../course13-communication-with-external-hardware/exercise/code/04_measurement.py ../course13-communication-with-external-hardware/exercise/solution/04_measurement.py	
../course13-communication-with-external-hardware/exercise/code/04_measurement.py	../course13-communication-with-external-hardware/exercise/solution/04_measurement.py
26 cmd2 = "MEAS:VOLT:DC? 10, 0.003\n"	26 cmd2 = "MEAS:VOLT:DC? 10, 0.003\n"
27	27
28	28
29 # transfer commands and query result (add 3 lines)	29 # transfer commands and query result
30	30
	31 s.write(cmd1)
	32 s.write(cmd2)
	33 res = s.readline()
31	34
32 XXX = XXX(XXX) # convert from string to float	35 res = float(res) # convert from string to float
33 print("Voltage: %s V" % res)	36 print("Voltage: %s V" % res)
34	37
	38
35 s.close()	39 s.close()