Tracing Ifs

Computer Science - Week 3  
Jul 29, 2022 - Version 1.0.3

Please make sure that all members of the group place their UD **email** AND **name** below.

Choose roles following the [instructions here](https://blockpy.cis.udel.edu/assignments/reading/bakery_appendix_pogil).

You should work in groups of 3. If you cannot find 3 group members, then work in groups of 2.

| **Role** | **Name** | **Email** |
| --- | --- | --- |
| **Manager** | Zach Phillips | phillizr@udel.edu |
| **Speaker** | Mason Davis | mcd@udel.edu |
| **Recorder** | Nick O’Haire | nohaire@udel.edu |

# 1) Code Reading

The Manager should download the following Python code and open it in Thonny: <https://gist.githubusercontent.com/acbart/7b8b1e0d2687defe428abb15cc4490f5/raw/567253df715d134d603767912b0d93f37f3f3a89/tracing_ifs.py>

`

Together, your group should read through the code.

1. What does this code do? Explain in plain english. Focus on explaining the entire code’s purpose, not the little details of what each line does. Aim to write at least 3 sentences.

| The first thing the program does is ask the user to input what day of the week it is. It then converts that day of the week into a number using the convert\_date function. It then uses the first two functions to decide if the day of the week is early in the week or late in the week. It then returns a string based on if it is either early or late in the week or if it is not due yet. |
| --- |

2. How many [**scopes**](https://python-sneks.github.io/pages/v3_0/22-scope/) are there in the program? Remember, IF statements do NOT have their own scope, but functions DO.

| 1 global scope, 6 function scopes. So there are 7 total scopes. |
| --- |

3. How many **bodies** are there in the program? Remember, IF statements and Functions BOTH have their own bodies. Hint: You probably forgot one special body that every program has!

| 1 global body, 6 functions bodies, 10 IF statement bodies. So there are 17 total bodies. |
| --- |

4. Write out each function name and then list the variables defined inside that functions’ scope:

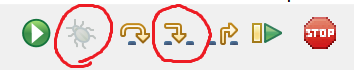
| Is\_early’s variable is day\_of\_week  Is\_late’s variable is day\_of\_week  Convert\_date’s variable is date  Is\_valid\_day’s variable is date  Get\_today’s variable is date  Main’s variables are today and day\_of\_week |
| --- |

# 2) Code Tracing

5. Assuming the user typed in the text Wednesday, write out the order that the lines are executed in the boxes below. Some boxes are already filled out for you.

Before you get started, here are some hints:

* You may not need all the boxes.
* Only write a line number when it is **started**, not when it is **returned** to.
* You can assume the function call jumps to the first line of the body, not the header.
* Use the Debugger in Thonny to complete this problem! That’s the “bug” button that lets you “Step Into” code. Alternatively, you can use the [Python Tutor website](http://www.pythontutor.com/visualize.html#code=today%20%3D%20%22Tuesday%22%0Adef%20is_early%28day_of_week%3A%20int%29%20-%3E%20bool%3A%20%0A%20%20%20%20return%20day_of_week%20%3C%202%20%0A%20%0Adef%20is_late%28day_of_week%3A%20int%29%20-%3E%20bool%3A%20%0A%20%20%20%20return%20day_of_week%20%3E%3D%204%20%0A%20%0Adef%20convert_date%28date%3A%20str%29%20-%3E%20int%3A%20%0A%20%20%20%20if%20date.lower%28%29%20%3D%3D%20'monday'%3A%20%0A%20%20%20%20%20%20%20%20return%200%20%0A%20%20%20%20elif%20date.lower%28%29%20%3D%3D%20'tuesday'%3A%20%0A%20%20%20%20%20%20%20%20return%201%20%0A%20%20%20%20elif%20date.lower%28%29%20%3D%3D%20'wednesday'%3A%20%0A%20%20%20%20%20%20%20%20return%202%20%0A%20%20%20%20elif%20date.lower%28%29%20%3D%3D%20'thursday'%3A%20%0A%20%20%20%20%20%20%20%20return%203%20%0A%20%20%20%20elif%20date.lower%28%29%20%3D%3D%20'friday'%3A%20%0A%20%20%20%20%20%20%20%20return%204%20%0A%20%20%20%20else%3A%20%0A%20%20%20%20%20%20%20%20return%205%20%0A%20%20%20%20%20%20%0Adef%20is_valid_day%28date%3A%20str%29%20-%3E%20bool%3A%20%0A%20%20%20%20return%20date.endswith%28'day'%29%20%0A%20%0Adef%20get_today%28%29%20-%3E%20str%3A%20%0A%20%20%20%20date%20%3D%20input%28%22What%20is%20the%20current%20day%20of%20the%20week%3F%22%29%20%0A%20%20%20%20if%20not%20is_valid_day%28date%29%3A%20%0A%20%20%20%20%20%20%20%20return%20%22Unknown%20day%22%20%0A%20%20%20%20return%20date.title%28%29%20%0A%20%0Adef%20main%28%29%3A%20%0A%20%20%20%20today%20%3D%20get_today%28%29%20%0A%20%20%20%20print%28%22Today%20is%22,%20today%29%20%0A%20%20%20%20day_of_week%20%3D%20convert_date%28today%29%20%0A%20%20%20%20if%20is_early%28day_of_week%29%3A%20%0A%20%20%20%20%20%20%20%20print%28%22You%20have%20plenty%20of%20time,%20relax!%22%29%20%0A%20%20%20%20elif%20is_late%28day_of_week%29%3A%20%0A%20%20%20%20%20%20%20%20print%28%22Well,%20it's%20too%20late%20now.%20Better%20give%20up!%22%29%20%0A%20%20%20%20else%3A%20%0A%20%20%20%20%20%20%20%20print%28%22It's%20not%20even%20due%20yet!%20It%20can%20wait.%22%29%20%0A%20%20%20%20%20%20%0Amain%28%29%0Astatus%20%3D%20is_early%28convert_date%28today%29%29%0Aprint%28status%29&cumulative=false&curInstr=0&heapPrimitives=false&mode=display&origin=opt-frontend.js&py=3&rawInputLstJSON=%5B%5D&textReferences=false) to run and step through the code.
* Seriously, use Thonny. If you guess on this one, you are almost guaranteed to get it wrong.



| **1** | **2** | **5** | **8** | **22** | **25** | **31** | **42** | **32** | **26** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **27** | **23** | **29** | **33** | **34** | **9** | **11** | **13** | **14** | **35** |
| **3** | **37** | **6** | **40** | **43** | **9** | **11** | **12** | **3** | **44** |
|  |  |  |  |  |  |  |  |  |  |

# 3) Thinking Critically

6. For each function in the program, give two unit tests. Remember that a unit test is specified by arguments and the expected return value.

| Function | Arguments | Expected |  | Function | Arguments | Expected |
| --- | --- | --- | --- | --- | --- | --- |
| is\_early | 0 | True |  | convert\_date | Tuesday | 1 |
| is\_early | 4 | False |  | convert\_date | Saturday | 5 |
| is\_late | 0 | False |  | is\_valid\_day | Wednesday | True |
| is\_late | 4 | True |  | is\_valid\_day | Wed | False |

7. For each of the functions listed above, how many unit tests would be ideal at a **minimum**? Explain your reasoning for why that minimum number of tests is reasonable.

| is\_early:  3 would be the minimum number of tests, we would test with a number less than 2, greater than 2, and then 2 itself. |
| --- |
| is\_late:  3 would be the minimum number of tests, we would test with a number less than 4, greater than 4, and then 4 itself. |
| convert\_date:  7 would be the minimum number of tests 6 for the days of the week and one weekend, an empty string, and a random word. |
| is\_valid\_day:  3 would be the minimum number of tests, we would test a string that ends with “day”, and a string that doesn’t end with “day”, and an empty string. |

# 4) Reflect and Review

Discuss among yourselves: what did you learn from this activity? What was surprising or interesting? If you didn’t learn anything, what do you think we were trying to teach you? How could this activity be improved?

| Nick: I learned that IF statements can have their own bodies within function bodies. How if statements can skip bodies and execute the next body of an elif or else function was something that was interesting. |
| --- |
| Mason: I learned in the code tracing section how to better understand the order of lines called in a function. I think you were trying to teach us more about the details about if statements. I would improve this activity by having us try and write our own if statements to solve basic problems. |
| Zach: I learned more about the different types of bodies that exist and how to count them. I found the program tracing interesting, and the Python Tutor website was helpful and insightful. |

# Final Submission

When your team is happy with your answers for all the questions, download this file as a Word Document (docx) and upload the file to the appropriate assignment on Canvas.

Only one member of your team needs to submit.