

## CSCI 150 HW: Python tracing & conditionals practice

*Due: Wednesday, September 18*

You should consider the code in each exercise separately from the other exercises.

1. To receive full credit for this exercise, you should do the following:
  1. **Trace:** First, trace the execution of each Python program below, using the template shown in class. That is, scratch work goes on the left, and keep track of the values of any variables on the right. You should complete the exercise *without* using the Python interpreter. (You are welcome to refer to your notes, read Python documentation, look at examples from class, *etc.*; just don't actually run any code.)
  2. **Check:** Now use a Python notebook to run the code and examine the values of all the variables. Do the final values of the variables agree with what you wrote down in step 1?
  3. **Evaluate:** If your answer to part 1 was different than the actual output, keep experimenting with it, consult the textbook or Python documentation, ask a friend or TA or professor, *etc.* until you can explain why the code works the way it does *and* what your misunderstanding(s) were in part 1. (You do not need to do anything for step 3 if the output agrees exactly with what you wrote in step 1.)

```
t = 5
nn = 2 + t
zebra = "fish"
if 3 * nn < t + 18:
    fish = "zebra"
    if zebra == fish:
        fish = zebra
        t = t + 1
    elif zebra == "zebra":
        t = t - 1
    else:
        zebra = fish
        t = t + 2
nn = nn + 3

if t * 2 > nn:
    t = nn + 1
else:
    t = nn - 1
```

2. We are having a party with amounts of tea and candy. Write a function called `party` to decide the outcome of the party: either `bad`, `good`, or `great`. Usually, a party is good if both tea and candy are at least 5. However,

if either tea or candy is at least double the amount of the other one, the party is not just good, but great. In all cases, if either tea or candy is less than 5, the party is always bad.

Your `party` function should take two inputs called `tea` and `candy`, both of type `int`, and should output a `str`.

For example,

- `party(3, 7)` should output "bad".
- `party(6, 5)` should output "good".
- `party(6, 12)` should output "great".