DS2500
1/28 - Tues.

Admin

- Lab 2 yday
- Hw2 due 1/31 9pm
  - names of tests: dupe ignores space/case (strip() not sufficient!)
- Hw3 out Fri
- Lab 3 Monday
- DS2500 Semester Project
  - teams
  - proposal due 2/28

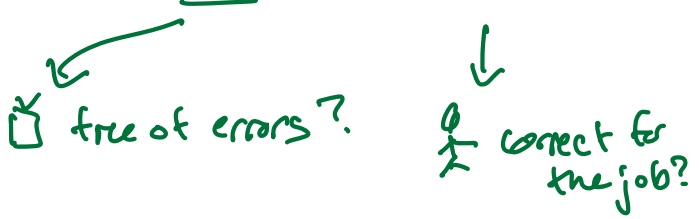
Agenda

1. Debugging Code
2. Interpreting Errors
3. Unit testing

1. Debugging Code

↳ more sophisticated, intricate code

we need to be confident that: our code works + works as expected



→ free from errors: debugging

look for, resolve Python error messages

→ correct for the job? unit testing

verifying correctness of individual functions

(prev: DS2000 - test case for entire program)

This week's dataset: 2023 Boston marathon

- csv file
- a little messy
- name, age, gender, country, finish time, ...,
- top 1000 finishers

privacy / ethics:

- Source: baa.org
  - accuracy ☑
  - collected w/ consent ☑
- personal info
  - if we don't need names, ignore them!
- Who is left out?
  - nonbinary runners?
  - skews male?
  - *same time?

In a moment, we will debug starter code:

- work? *
- work as expected?

* Run the code

If we get an error:

1. DON'T PANIC!
2. Scroll to bottom and read the msg
3. Add prints to find out the problem
   - value of the variable `print(x)`
   - data type of the variable `print(type(x))`

Current goal:

↳ resolve all error messages using this method

2. Interpreting Errors

- **Syntax Error**
  - ↳ typo?
    - missing paren, colon, etc.
    - check the line above
- **Name Error**
  - ↳ typo?
    - variable doesn't exist
    - forgot to import module
- **Type Error**
  - ↳ operation/function doesn't fit the data type
    - "5" + 5
- **Value Error**
  - ↳ value is the problem, not the data type
    - int("5") ✓
    - int("5.5") ✗
- **Index/Key Error**
  - ↳ outside the bounds of a list / dictionary

OH: what type of error?

examples

- lst1 = [4,5]
  lst2 = [6,9]
  lst3 = lst1 - lst2
- lst1 = [4,5]
  print(lst4)
- [4,5] = lst
- x = float("13,789.50")
- dct = {'2': 3}
  print(dct[3])

3. Unit Testing

Debugging: does our code work?

Unit Test: does our code work as expected?

- solving the problem?

DS2000: test case for the whole program

DS2500: test individual functions

- functions are short, concise
- main/driver: creates objects, calls functions
- if our indv functions work, we can put them together w/ confidence

Yoda - pretty sure whole program works no matter what's in data file, what user types in

unit testing demo on Labs

Today's code: look for older runners!

Functions in driver:

- main (no unit tests)
- filter_age (unit tests!)

Filter_age:

- parameters
  - age (int) — example input: 45 — expected:
  - list of ages (ints) — [46, 47, 48] — [x, y, z]
  - list of times (strings) — [x, y, z]
- returns
  - list of times, filtered for every item > age — 200: [45, 10] → [2, 6]; [] → []

in code (tri): compare expected vs. actual

20

[21, 19, 17]

[x, y, z]

[x]