



Technical Interviewing

Steps to Victory

1. Hack your mindset
2. Use UPER
3. Teach the problem



Hack your Mindset

RELAX.

Pretend the interviewer is a coworker who has just come to you for help on a difficult problem.

"Hey, Beej! I'm stuck on this problem--can you help me out?"



Hack your Mindset

- Teaching a problem is the right level of detail for an interview.
- Makes you think out loud.
- Shows you can systematically solve problems.
- Causes you to act like you're part of the team.



UPER in an Interview

UPER

- Understand: teach the problem to the interviewer
- Plan: teach the plan to the interviewer
 - Conceptually! As little code as possible!
- Execute: code it up (or go over your code if you wrote it earlier)
- Reflect: look on ways to improve it, or future directions.



Sample Interview

"Can you show me how you'd reverse a linked list?"

"This problem was to reverse a linked list. Can you tell me how you solved it?"



Sample Interview

```
def reverse_list(head):  
    prev = None  
    cur = head  
    while cur is not None:  
        old_next = cur.next  
        cur.next = prev  
        prev = cur  
        cur = old_next  
    return prev
```



Sample Interview

```
def reverse_list(head):  
    prev = None  
    cur = head  
  
    while cur is not None:  
        old_next = cur.next  
  
        cur.next = prev    # Reverse the pointer  
  
        prev = cur  
        cur = old_next  
  
    return prev
```



Make Your Code Look Sharp

You dress well for the interview

You make sure your resume looks good

Give your code the same treatment

- Good comments
- Good use of whitespace
- Consistent spacing around punctuation and operators
- Consistent indentation
- Good variable and function names



Summary

- Teach the problem
- UPER is your best friend
- Make your code look sharp



Questions?

Ask away!

