launch_code

Live Coding

Technical Assessments

1. Online Coding/Logic tests
    a. Before the interview
2. Technical Questions
    a. During the interview
3. Personal Projects, and Portfolios
    a. During the interview
4. Live Coding
    a. During the interview
5. Company Assigned Work
    a. After the interview

Live Coding

Live Coding: A programming problem to be solved "live" in front of the interviewers.

Format: Whiteboard, online in a shared document (Collabedit.com), on a computer.

My Personal Opinion/Disclaimer

Live coding performance doesn't necessarily correlate with job performance!

You are rarely, if ever, asked to code in front of a group of people away from your coding environment on the job.

...It Still Happens

Live Coding is still regularly used in the interview process.

You should walk into every interview expecting them to ask you to live code.

You can’t hide. Live coding will clearly show your interviews that you CAN code.

Environment

You DON’T have your normal tools. No IDE, no Google, no Stack Overflow, no examples of previous work.

You DO have access to another programmer. Your interviewer, or someone in the room should know what you’re doing. You can talk with them. Share your ideas, ask for feedback. Draw them into your problem solving process.

Live Coding

Solving a small programming puzzle on a whiteboard in the interview.

1. Effort
2. Ability to explain logic
3. Communication
4. Ability to work under pressure
5. Work with others
6. Right vs Wrong

Preparation & Resources

Perfect practice makes perfect. Find a problem, and practice solving it on paper, without your computer. Talk through the logic, explain your solution outloud. Finally test yourself by running your code in your IDE.

1. Hackerrank.com
2. Codewars.com
3. Checkio.org
4. Exercism.io

Best Practices

1. Clarify/Planning the problem
2. Break it down into small solvable chunks
3. Logic (pseudocode, comments, or spoken)
4. Code (syntax)
5. Test (Be the compiler, keep track of your variables and output)
6. Check in with interviewers

Example

Compare the Triplets (Hackerrank.com)

Provided Codestub

```
import sys

a0,a1,a2 = input().strip().split(' ')

a0,a1,a2 = [int(a0),int(a1),int(a2)]

b0,b1,b2 = input().strip().split(' ')

b0,b1,b2 = [int(b0),int(b1),int(b2)]
```

Clarification/Planning

We have 6 variables in memory in 2 chunks. We need to compare the first variable of the first chunk to the first variable in the second chunk and so forth to determine which is larger. If we find a variable is bigger we need to increment that chunks counter. At the end we need to print out the counters.

I will reorganize the “chunks” into lists, the lists will always be the same size so I can compare them with one for loop, and I will need 2 counter variables.

Break the problem down/Pseudocode

1. Create lists
2. Create counter variables
3. For loop with comparison statement
4. Print statement

Code

```
a_list = [a0,a1,a2]

b_list = [b0,b1,b2]

a_score = 0

b_score = 0
```

Continued...

```
for i in range(len(a_list)):
    if a_list[i] > b_list[i]:
        a_score += 1
    elif a_list[i] < b_list[i]:
        b_score += 1
print('{} {}'.format(a_score, b_score))
```

Testing

Input: a0 = 5, a1 = 6, a2 = 7, b0 = 3, b1 = 6, b2 = 10

5 > 3 -> a_count = 1

6 == 6 -> neither count gets incremented

7 < 10 -> b_count = 1

Output: 1 1

Checkin

Walk them through a test case. Ask them if they have questions, or have a test case that would break your solution. Then you can brainstorm fixes with them.

Q & A

Questions!

Paul Matthews

paul@launchcode.org