



Pyladies 2024-09-17

Python up your next Coding interview

Tips and tricks
to confidently pass it

By Luca Lodi





Gemini



Create an image of 2 happy people having
an interview in front of a computer in a
hyperrealistic photography style

◆ Gemini



Create an image of 2 happy people having an interview in front of a computer in a hyperrealistic photography style



I'm showing limited results for images of people at this time. Can I help with anything else?





Pyladies 2024-09-17

Python up your next Coding interview

Tips and tricks
to confidently pass it

By Luca Lodi



About me



Luca Lodi

Software Engineer
@ Pinterest

Why ?

Luca

Interviewed

with

30+ companies

Many loops ...

2.5y ago

Interviewer

For *Interns / Mid / Senior / Staff*

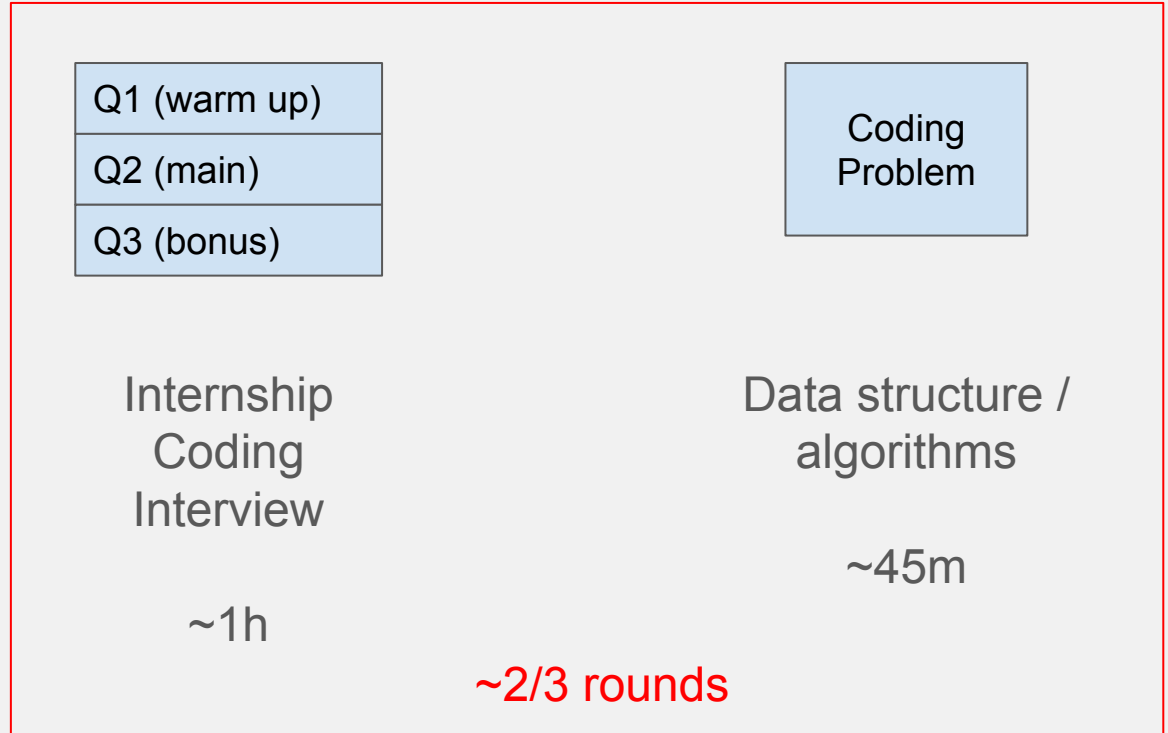
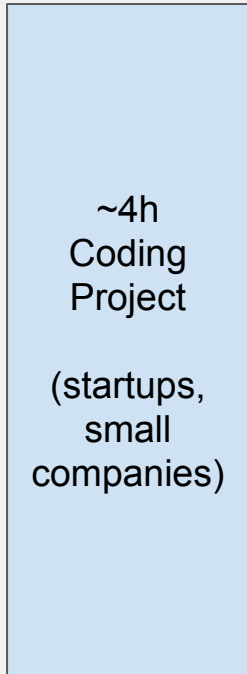
25+ interviews

this year



Interview structure

Many interview types



How to *pass* an interview ?



DOs

How to ~~pass~~ an interview ?



DOs

How to *fail* an interview ?



DON'Ts

Most failures are because

1

Stay silent



DON'Ts

2

Not write code



DON'Ts

3

**Not manage
time / morale**



DON'Ts

Time management

45m

4 candidates

The iterative

Ask questions

Understand the
problem

Implement code

Run + Debug

Implement code

Run + Debug

Improve / refactor
code

Waterfall

Understands
problem /
Sketch solution

Implement code

Test
Run + Debug

Shoot first, Ask later

Jump into code

Run + Debug

Write test cases,
Not working

The perfectionist

Insist on finding an
optimal solution

Loose track of time

Write no code

4 candidates

The iterative

Ask questions



Understand the problem

Implement code

Run + Debug



Implement code

Run + Debug

Improve / refactor code



Waterfall

Understands problem / Sketch solution



Implement code

Test
Run + Debug

Shoot first, Ask later

Jump into code



Run + Debug

Write test cases,
Not working

The perfectionist

Insist on finding an optimal solution



Loose track of time



Write no code



**What
companies
look for ?**

Evaluation criteria

Coding:

**Code that
works**

Code organization:

**Readable,
tested code**

Problem solving:

**Unblock
yourself**

Collaboration:

**Ask,
explain**

An iceberg floating in a blue ocean under a blue sky with white clouds. The tip of the iceberg is above the water, while the much larger, jagged base is submerged. To the right of the iceberg, a list of software development practices is presented in colored boxes, with the top two above the water and the bottom four below it.

Code that works

Tested code

Readable code

Troubleshoot issues

Manage time

Collaborate

Iterate & improve code



Prepare !



If you are new to coding



LeetCode

Or really *anything* similar



Learn basic data structures



Is HashMap the solution to every problem?

All Images Videos News Books Finance Web



Medium · Varsha Das

50+ likes · 6 months ago

Is HashMap the solution to every problem? Let's figure out!

We think that the only solution to almost every coding problem, especially a Leetcode problem—**HashMap**—is the collective favourite data ...

People also ask

Why not use HashMap for everything?





Real or mock
interviews

Just apply !



DOs

Use Python

Sketch solution == valid code

```
"""  
Iterate over the number in the list  
    if even  
        increase total by the number  
"""
```

```
for number in my_list:  
    if number % 2 == 0:  
        total += number
```

Battery included: sort

```
sorted_people_by_name = sorted(  
    people,  
    key=lambda p: p['name']  
)
```

Battery included: group by

```
import itertools
grouped_by_city = itertools.groupby(
    data,
    key=lambda p: p['city']
)
```

Battery included: dict comprehension

```
city_tot_salaries = {  
    city: sum(  
        p['salary']  
        for p in people  
    )  
    for city, people in grouped_by_city  
}
```

Battery included: list comprehension

```
city_tot_salaries = {  
    city: sum(  
        p['salary']  
        for p in people  
    )  
    for city, people in grouped_by_city  
}
```



DOs

Ask questions

Do ask questions

- **Structure**
- **Examples**
- **What is allowed ?**

**To the
recruiter**

- **Clarify assumptions**
- **Can I google ?**
- **Help / hint**

**To the
interviewer**



DOs

Iterate, quick



optimal
solution
not coded

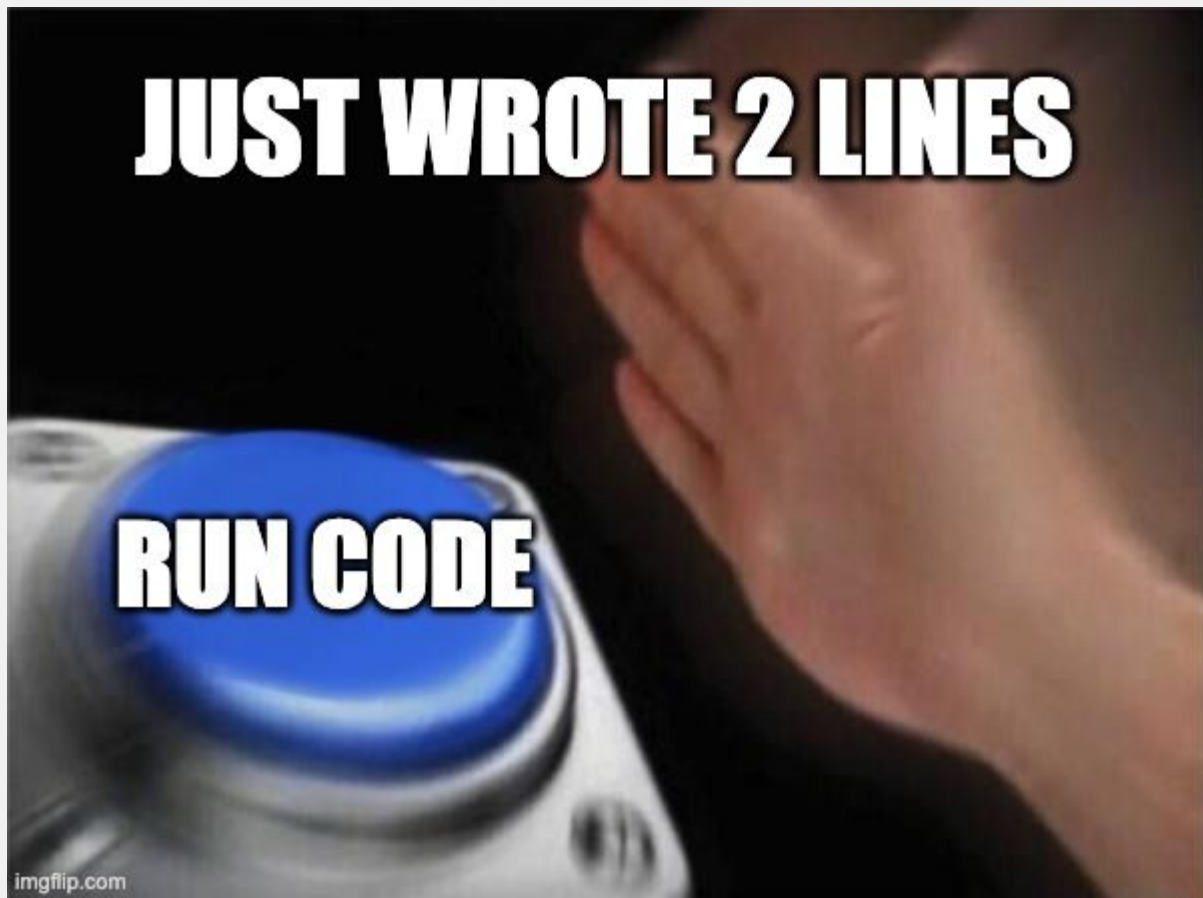


coded
brute force
solution

Start **simple**,

Improve if
you have time

Optimize later



Run code
frequently,
to discover
errors early



DOs

Debug early, or leave time for it

May cause panic if found late

SyntaxError

IndentationError

NameError: name `'ii'` is not defined

Typos

**Logic
issues**

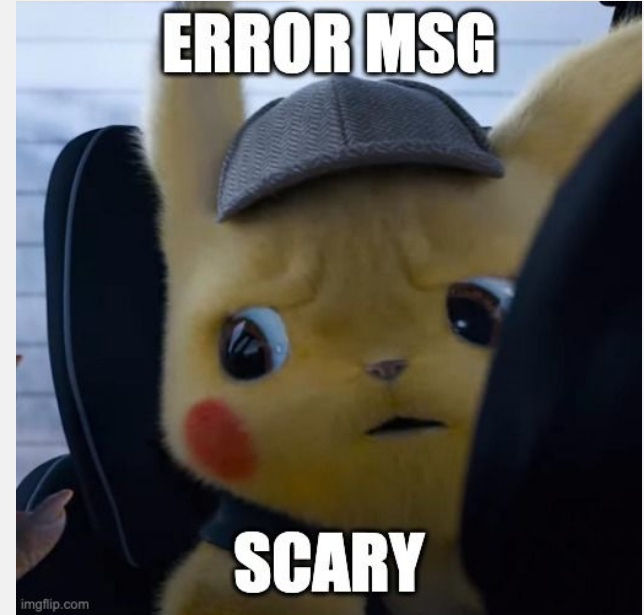
KeyError: `'c'`

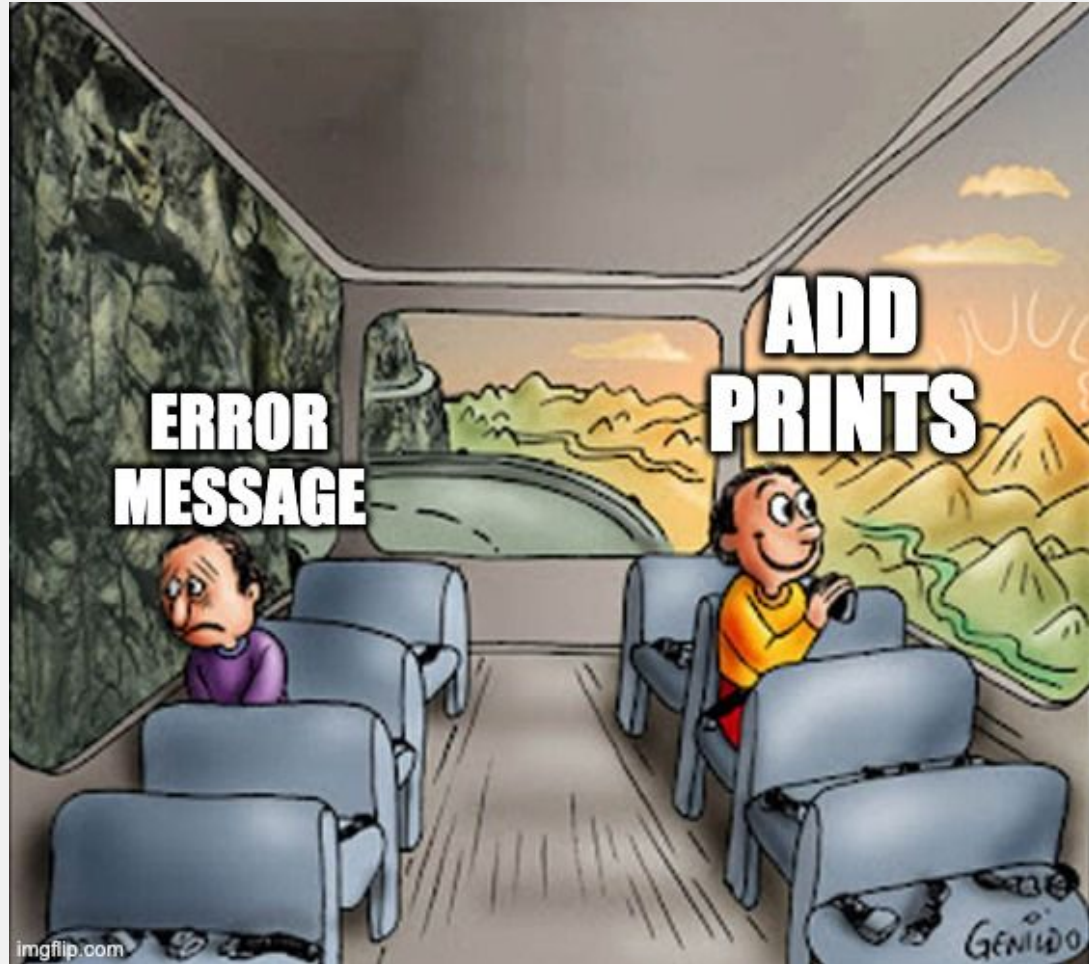
IndexError: `list` index out
of `range`

TypeError: unsupported operand `type(s)` for `/:` `'str'`
and `'int'`



**Start from
the error msg**





**Debug
by
print**



DOs

Test the code

**Test cases
are not always
provided**

Simple: prints

```
print(my_function([3, 45, 87, 129]))
```

```
data = [3, 45, 87, 129]  
print(my_function(data))  
# expect 42
```

Better: asserts

```
data = [3, 45, 87, 129]
expected_result = 42
result = my_function(data)

print(result, expected_result)
assert expected_result == result, "Got a
    different result than expected"
```

Better: Multiple test cases

```
def assert_equals(expected_result, result):  
    assert expected_result == result, f"Got a different  
    result than expected. Got {result}, but expected  
    {expected_result}"  
    print(f"The result matches the expected one  
    ({expected_result})")
```

```
data1 = [3, 45, 87, 129]  
assert_equals(42, my_function(data1))
```

```
data2 = [3, 13, 23]  
assert_equals(10, my_function(data2))
```

The result matches the expected one (42)
The result matches the expected one (10)

Easier troubleshooting

```
data1 = [3, 45, 87, 129]
assert_equals(42, my_function(data1))

data2 = [3, 13, 23]
assert_equals(11, my_function(data2))
```

The result matches the expected one (42)

Traceback (most recent call last):

File ~/solution.py:13

assert_equals(11, my_function(data2))

File ~/solution.py:5 in assert_equals

assert expected_result == result, f"Got a different result than expected.
Got {result}, but expected {expected_result}"

AssertionError: Got a different result than expected. Got 10, but expected 11



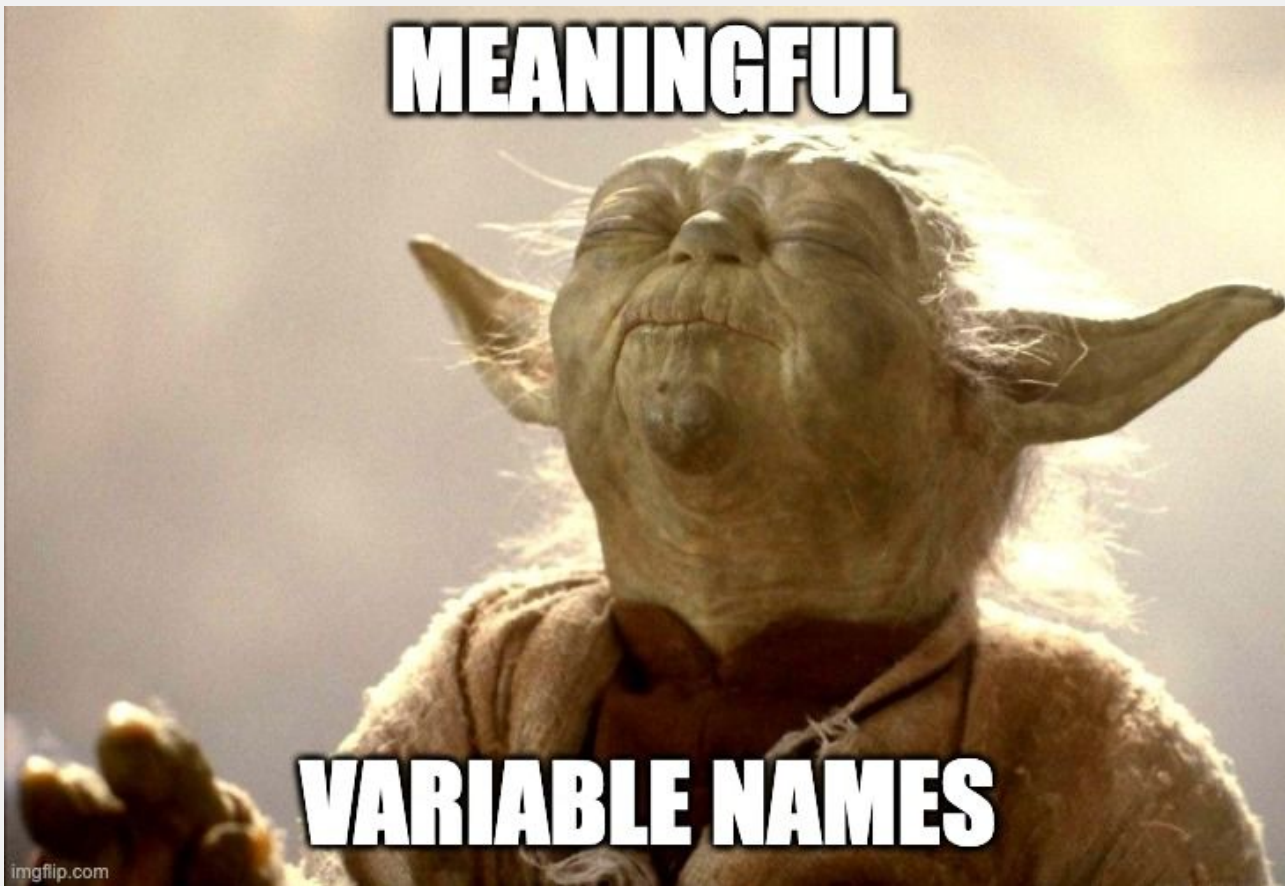
DOs

Readable code



**Focus on defining
the function's
inputs and outputs**

MEANINGFUL



VARIABLE NAMES



Refactor **last
if finished early**

TL;DR

Cheatsheet

Prepare

Take test interviews

Leetcode !

Prepare your “test”
framework

Frame the problem

Ask clarifying questions

Sketch your solution:
Easy / brute force first

Optimize if you have time

Solve the problem

Incremental, iterative
coding

Test as you go

Reserve time to
troubleshoot issues

Finalize your answer

Troubleshoot issues

Add more test cases

Refactor code

Questions



Thank you



beary berry much !