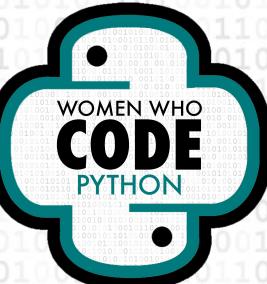


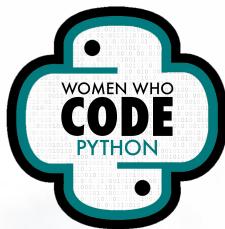
# Welcome everyone!

- You can find these slides on GitHub here:  
<https://github.com/WomenWhoCode/WWCodePython>
- Please make sure your chat is set to “All panelists and attendees”.
- Some housekeeping rules:
  - Everyone will be muted throughout the webinar, but there will be opportunities for participation!
  - Please share your thoughts on the chat and/or ask questions in the Q&A.
  - The entire team is here today. Please reach out to us with any technical questions!



# WELCOME WOMEN WHO CODE





# Women Who Code Python

**Intro to Data Structures  
with Python:  
Ace the Technical Interview**



**Session #1: Introduction**

WOMEN WHO  
**CODE**<sup>®</sup>

# MEET YOUR TEAM



**Rishika Singh**  
Track Lead



**Jasmeen Rajpal**  
Evangelist

WOMEN WHO  
**CODE**

# OUR MISSION

Inspiring women to  
excel in technology  
careers.

WOMEN WHO  
**CODE**



# OUR VISION

A world where women are representative as technical executives, founders, VCs, board members and software engineers.

WOMEN WHO  
**CODE**



# OUR TARGET

Engineers with two or more years of experience looking for support and resources to strengthen their influence and levelup in their careers.



# CODE OF CONDUCT

**WWCode is an inclusive community**, dedicated to providing an empowering experience for everyone who participates in or supports our community, regardless of gender, gender identity and expression, sexual orientation, ability, physical appearance, body size, race, ethnicity, age, religion, socioeconomic status, caste, creed, political affiliation, or preferred programming language(s).

Our events are intended to inspire women to excel in technology careers, and anyone who is there for this purpose is welcome. We do not tolerate harassment of members in any form. Our **Code of Conduct** applies to all WWCode events and online communities.

Read the full version and access our incident report form at [womenwhocode.com/codeofconduct](http://womenwhocode.com/codeofconduct)

# 230,000 Members

70 networks in 20 countries  
Members in 97+ countries  
10K+ events  
\$1025 daily Conference tickets  
\$2M Scholarships  
Access to [jobs](#) + [resources](#)  
Infinite connections



# OUR MOVEMENT

As the world changes, we can be a connecting force that creates a sense of belonging while the world is being asked to isolate.



# Upcoming Events

WED  
**10**  
MAR

 Databases with Python: Session on Firebase  *Featured*  
6:00 PM – 8:00 PM (EST) |  Zoom

[Register](#)

THU  
**11**  
MAR

 Intro to Data Structures with Python: Ace the Technical Interview (Session #2:  
Python-Specific Data Structures)  *Featured*  
8:00 PM – 9:30 PM (EST) |  Zoom

[Register](#)

THU  
**25**  
MAR

 Intro to Data Structures with Python: Ace the Technical Interview (Session #3:  
Arrays & Matrices)  *Featured*  
8:00 PM – 9:30 PM (EDT) |  Zoom

[Register](#)

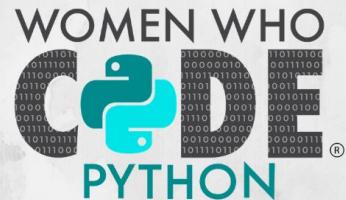
WED  
**31**  
MAR

 Databases with Python: Session on MongoDB  *Featured*  
7:00 PM – 8:00 PM (EDT) |  Zoom

[Register](#)



# Stay Connected



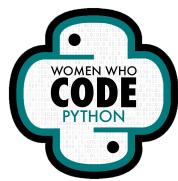
WOMEN WHO  
CODE  
PYTHON®

JOIN US ON SOCIAL MEDIA!

@WWCODEPYTHON

[WOMENWHOCODE.COM/PYTHON](http://WOMENWHOCODE.COM/PYTHON)

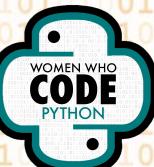


# Today's Agenda



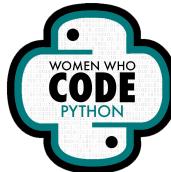
1. Series Information
2. The Coding Challenge
  - a. What and Why?
  - b. The 3 Types
3. How to Approach a Coding Challenge
4. Do's and Don'ts for Technical Interviews
5. Sample Interview Questions
6. Resources & Preparation
  - a. Leetcode
  - b. Hackerrank
  - c. Repl (for practice)

# Series Information



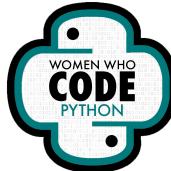
# Series Format

- Biweekly sessions
- Concepts & theory + applications
  - 1st half: slide presentation
  - 2nd half: Colab notebook
- Opportunity for live Q&A
- 1 real-life technical interview question per data structure



# Series Topics

- **Module #1:** Built-in/Python-specific
  - Lists, Dictionaries, Tuples, Sets
- **Module #2:** Linear, user-defined
  - Arrays & Matrices
  - Stacks & Queues
  - Linked Lists
- **Module #3:** Non-linear, user-defined
  - Trees
  - Heaps
  - Graphs
  - Maps & Hash Tables

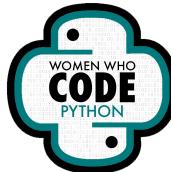


# The Coding Challenge



# The Coding Challenge

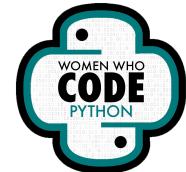
- What is a coding challenge?
  - tests that are sent to candidates by a company with the intent of screening technical skills/coding proficiency
- Why the coding challenge?
  - helps companies identify the best developers by looking at their work sample, allowing them to make a data-driven decision about whether the candidate should be invited onsite or not



# Coding Challenge - Types

Coding challenges come in 3 primary forms:

- Skill building exercises
- Take home interviews
- Live interview challenges

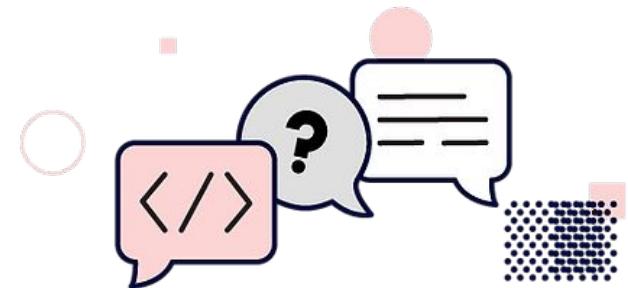


# How to Approach a Coding Challenge



# Understand & Analyze the Problem

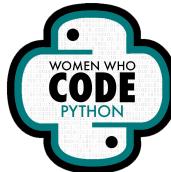
- Read the question several times and understand it completely
- Think about the question and analyze it carefully
- Take a paper and write down everything. What is given (input) and what you need to find out (output)?



# Understand & Analyze the Problem

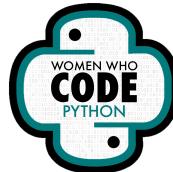
*While going through the problem you need to ask a few questions to yourself...*

1. Did you understand the problem fully?
2. Would you be able to explain this question to someone else?
3. What and how many inputs are required?
4. What would be the output for those inputs?
5. Do you need to separate out some modules or parts from the problem?
6. Do you have enough information to solve that question? If not, then read the question again or clear it to the interviewer.



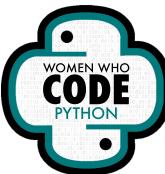
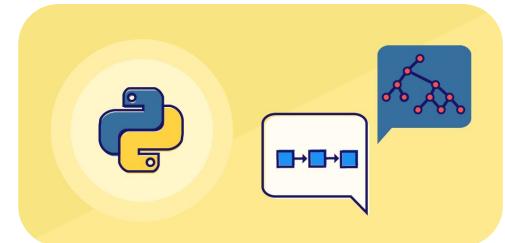
# Sample Input/Output

- Consider some simple inputs or data and analyze the output
- Now consider some complex and bigger input and identify what will be the output and how many cases you need to take for the problem
- Consider the edge cases as well
- You will get clarity on how many cases your code can handle!
- Sometimes, the company will have test cases to run your code on



# Break Down the Problem

- Break down the problem into smaller chunks and then try to solve each part of the problem
- Make a flow chart or a UML for the problem in hand (consider this for whiteboarding or take-home challenges)
- Divide the problem into subproblems or smaller chunks
  - Solve the subproblems and make independent functions for each subproblems
  - Connect the solutions of each subproblems by calling them in the required order, or as necessary
- Wherever it's required, use classes and objects depending on the problem's complexity



# Pseudocode → Replace with Real Code

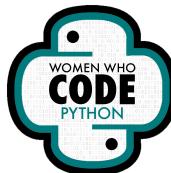
- Start with writing some pseudocode.  
Reading pseudocode gives a clear idea of what your code is supposed to do
- Once you have written the pseudocode, it's time to translate this into actual code
  - Replace each line of your pseudocode into real code in the language you are working on
- If you have divided your problem into subproblems, then write down the code for each subproblem

## Pseudocode

```
if the score is 90 or above  
    grade is an "A"  
else  
    if the score is 80 or above  
        grade is a "B"  
    else  
        if the score is 70 or above  
            grade is a "C"  
        else  
            grade is an "F"
```

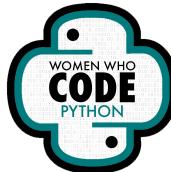
## Actual Python code

```
if score >= 90:  
    grade = "A"  
else:  
    if score >= 80:  
        grade = "B"  
    else:  
        if score >= 70:  
            grade = "C"  
        else:  
            grade = "F"
```



# Optimize & Simplify

- Always try to improve your code
- Look back, analyze it once again and try to find a better or alternate solution
- Writing the correct solution to your problem is not the final thing you should do
- Explore the problem completely with all possible solutions and then write down the most efficient or optimized solution for your code

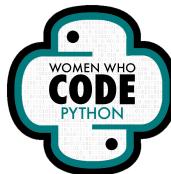


# Do's and Don't's of Technical Interviews



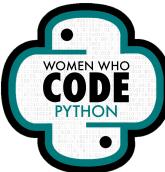
# The Do's

- Have questions ready to ask the interviewer
  - How is the work culture?
  - What is your favorite part of working here?
- Practice interviewing with your friend or someone else
- Research the position before interviewing
- Think out loud and ask questions when answering questions

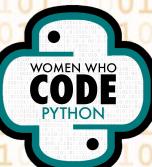


# The Dont's

- Ignore time considerations
- Let one bad round beat you down
- Assume you'll know what kinds of questions will be asked
- Get too stuck on a solution
- Use Google while interviewing
- Be late
- Be dishonest about your resume and previous experiences



# Resources & Preparation



# Resources

All these resources provides challenges for several different domains

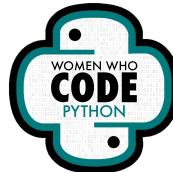
- Algorithms
- SQL
- Functional Programming
- AI etc..

Leet Code: <https://leetcode.com/>

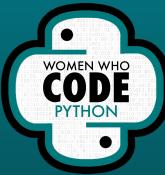
Hacker Rank: <https://www.hackerrank.com/>

Codewars: <https://www.codewars.com/>

Codechef: <https://www.codechef.com/>



# Q&A Time!



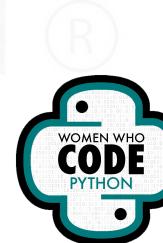
# Sample Interview Questions

Easy: FizzBuzz → Repl.it link:

<https://repl.it/join/ygqbfltx-rishikasingh2>

Problem Statement:

*Write a Python program which iterates the integers from 1 to 50. For multiples of three, print “Fizz” (instead of the number), and for multiples of five, print “Buzz”. For numbers which are multiples of both three and five, print “FizzBuzz”.*



# Sample Interview Questions

Medium: [Longest Palindrome Substring](#)

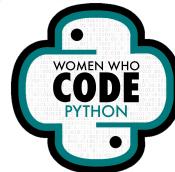
Problem Statement:

*Given a string s, return the longest palindrome substring in s.*

*Input:* s = “cbbd”

*Output:* “bb”

Source: leetcode.com



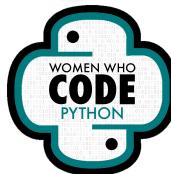
# Sample Interview Questions

Hard: [Longest Valid Parentheses](#)

Problem Statement:

*Given a string containing just the characters '(' and ')', find the length of the longest valid (well-formed) parentheses substring.*

Source: leetcode.com

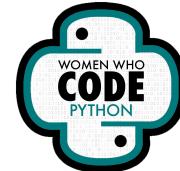
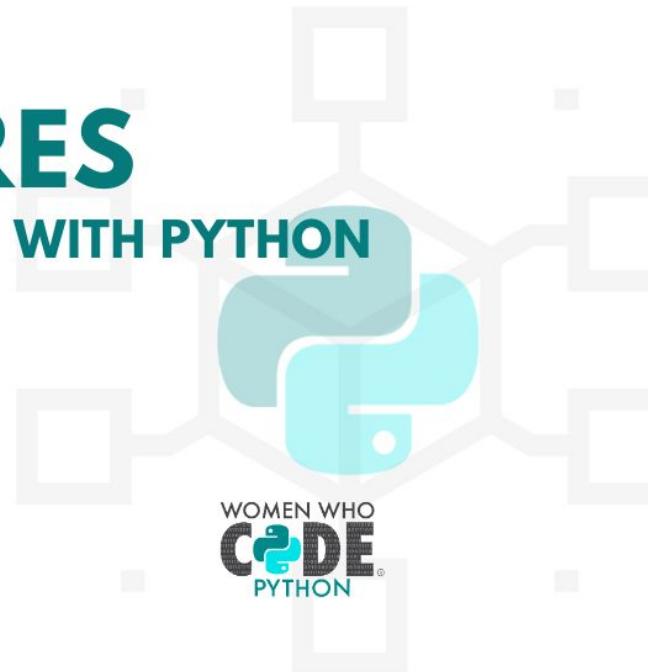


# Next Session!

INTRO TO  
**DATA  
STRUCTURES**  
WITH PYTHON

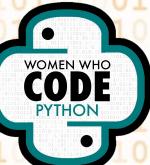
ACE THE  
TECHNICAL  
INTERVIEW

THU. MARCH 11TH  
@ 8:00PM EST



# Questions?

Join our Slack channel:  
**#intro-data-structures-stdy-grp**



# Thank You!

