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!


Women Who Code: Python
Beginner Python Study Group
Session 5: Programming Logic
WOMEN WHO
CODE
PYTHON

Meet us!

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.

WOMEN WHO CODE

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**

WOMEN WHO CODE

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

WOMEN WHO CODE

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.

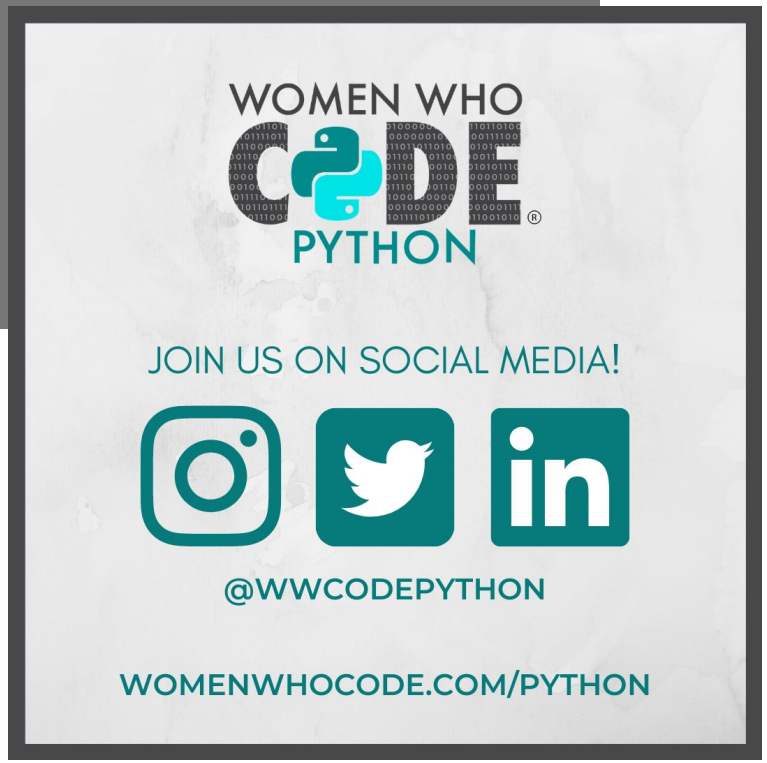
WOMEN WHO CODE

Upcoming Events!

FRI 30 OCT	👷 AMA with Megan, Data Scientist at Microsoft👷 4:00 PM – 5:00 PM (EDT) | Zoom	Register
WED 04 NOV	✨ Beginner Python Study Group ✨ Session 6: Open Q&A/Review Session *Recurring* 8:00 PM – 9:30 PM (EST) | Zoom	Register
THU 12 NOV	🚗 Introduction to Autonomous Vehicles 🚗 11:00 PM (EST) | Zoom	Register
WED 18 NOV	✨ Beginner Python Study Group ✨ Session 7: Writing Your Own Python Module *Recurring* 8:00 PM – 9:30 PM (EST) | Zoom	Register
WED 02 DEC	✨ Beginner Python Study Group ✨ Session 8: Mini Project *Recurring* 8:00 PM – 9:30 PM (EST) | Zoom	Register

WOMEN WHO CODE PYTHON

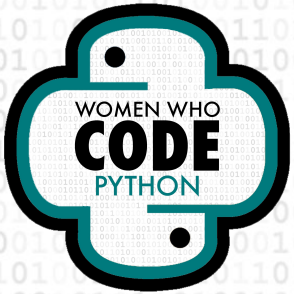
Stay Connected

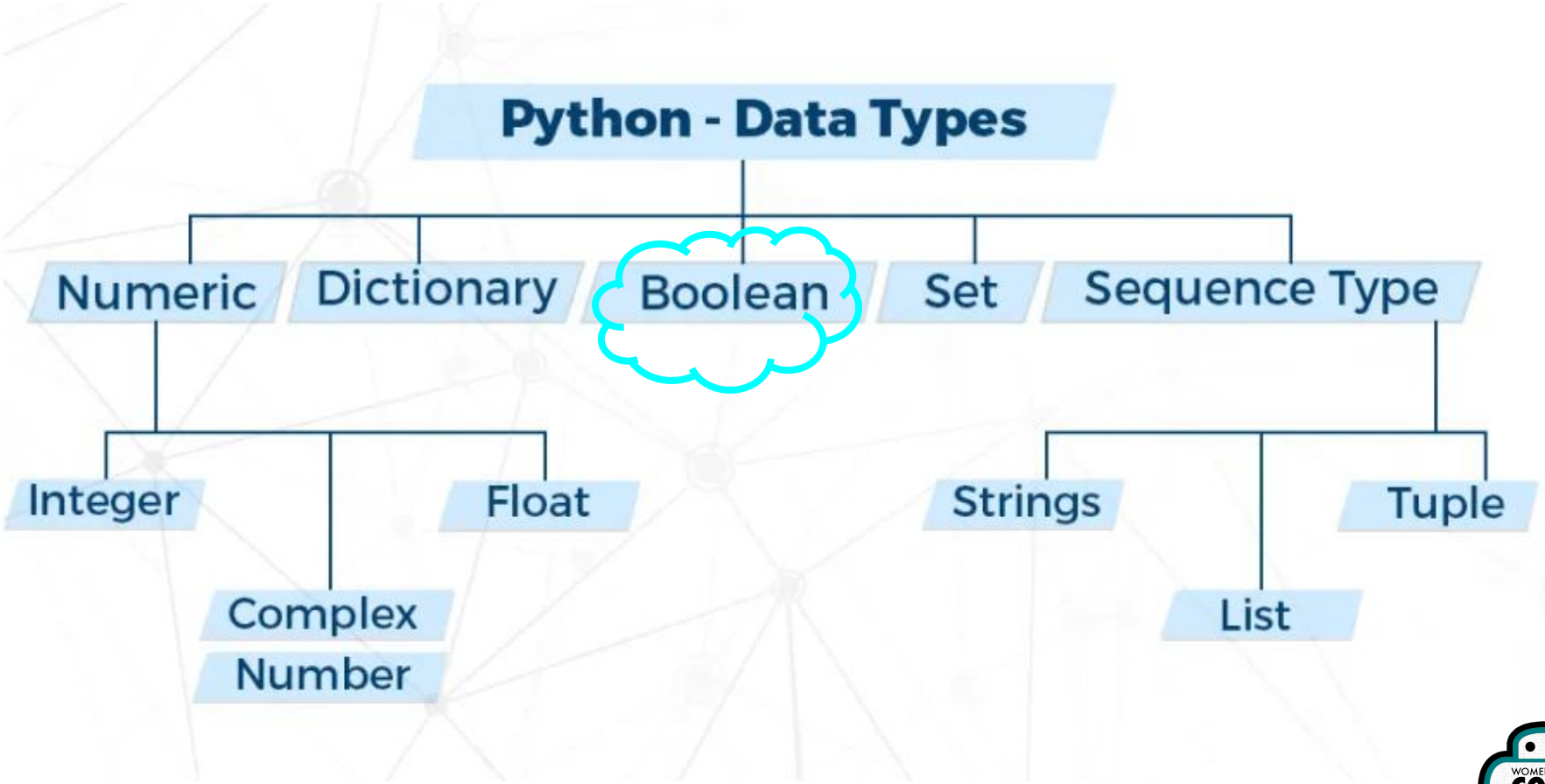


1. Recap of Session #4 🤓
2. Programming Logic
    a. Elements of Flow Control
    b. Flow Control Statements
        i. if statement
        ii. else statement
        iii. elif statement
        iv. while loop
        v. for loop
3. Google Colab - Live Coding! 👩‍💻
4. Wrap-Up 🔃

Session #4 Recap

- Visit https://www.youtube.com/user/WomenWhoCode for recording when available
- Slides & Code: https://github.com/WomenWhoCode/WWCodePython

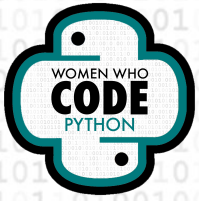



Python - Data Types
Numeric
Dictionary
Boolean
Set
Sequence Type
Integer
Complex Number
Float
Strings
List
Tuple

Boolean Values & Operators

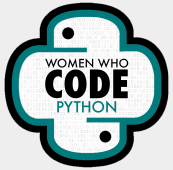
- Values:
  - *True*
  - *False*
- Operators:

Operator	Meaning
==	Equal to
!=	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to



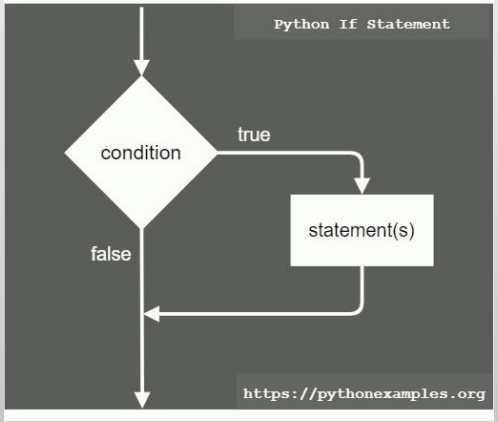
Elements of Flow Control

- Conditions
- Blocks of Code



Conditions

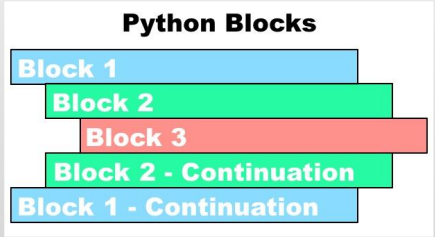
- Boolean expressions → conditions
- Condition: more specific name in the context of flow control statements
- Always evaluate down to a Boolean value (*True/False*)
- Decides what to do based on whether its condition is True/False
- Almost every flow control statement uses a condition



WOMEN WHO CODE PYTHON

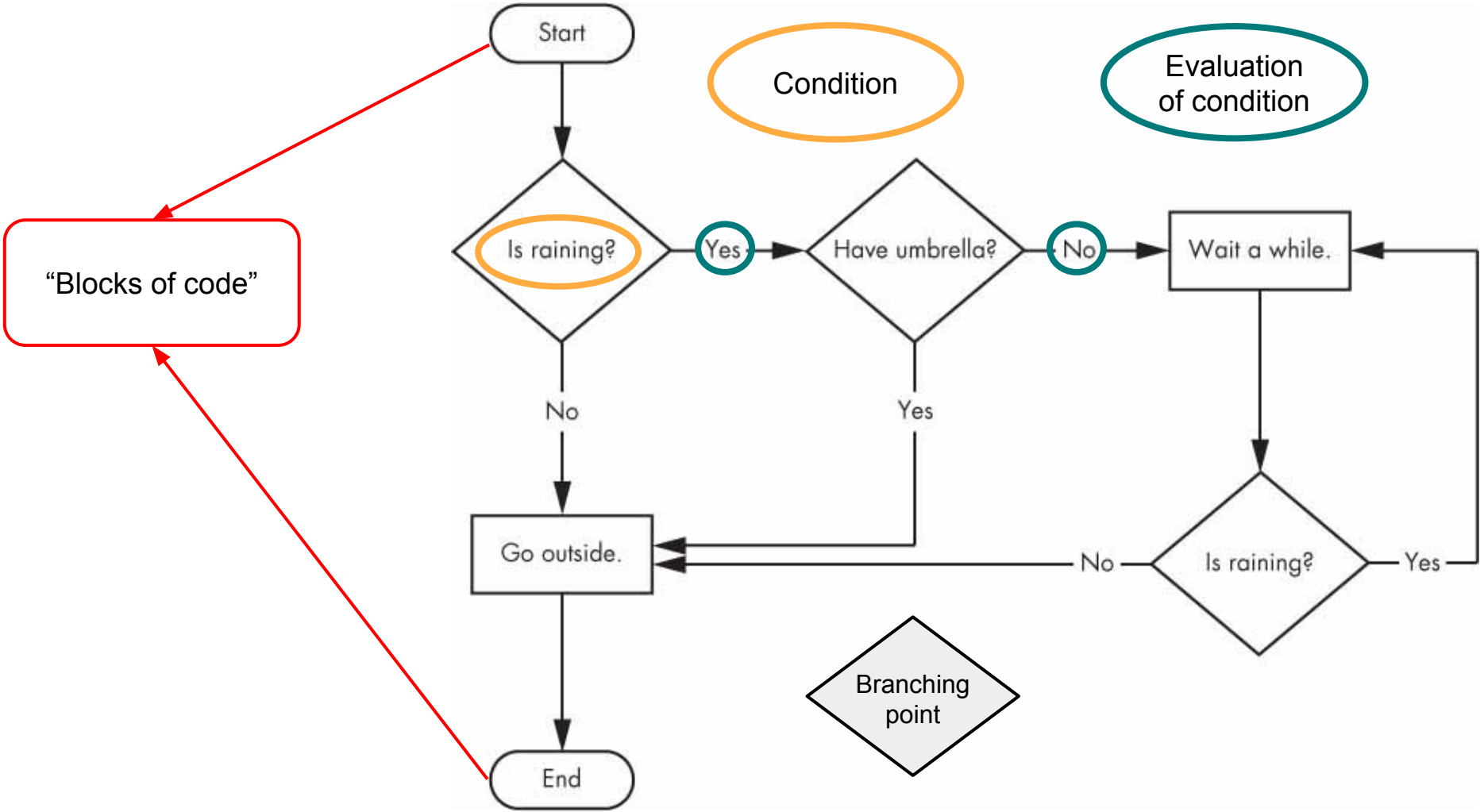
Blocks of Code

- Lines of Python code grouped together in blocks
- Block begins and ends from indentation
- 3 rules for blocks:
  - Begin when the indentation increases
  - Can contain other blocks
  - End when the indentation decreases to zero or to a containing block's indentation



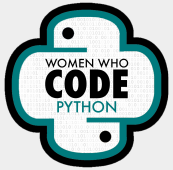

```
def addition_test(a, b):
    result = a + b
    """
    This is a block
    of commented lines.
    They aren't parsed and interpreted by the compiler.
    """
return result
```

WOMEN WHO CODE PYTHON


Start
Condition
Evaluation of condition
"Blocks of code"
Is raining?
Yes
Have umbrella?
No
Wait a while.
No
Yes
Go outside.
No
Is raining?
Yes
Branching point
End

Flow Control Statements

- Conditions
  - if
  - else
  - elif
- Loops
  - while
  - for

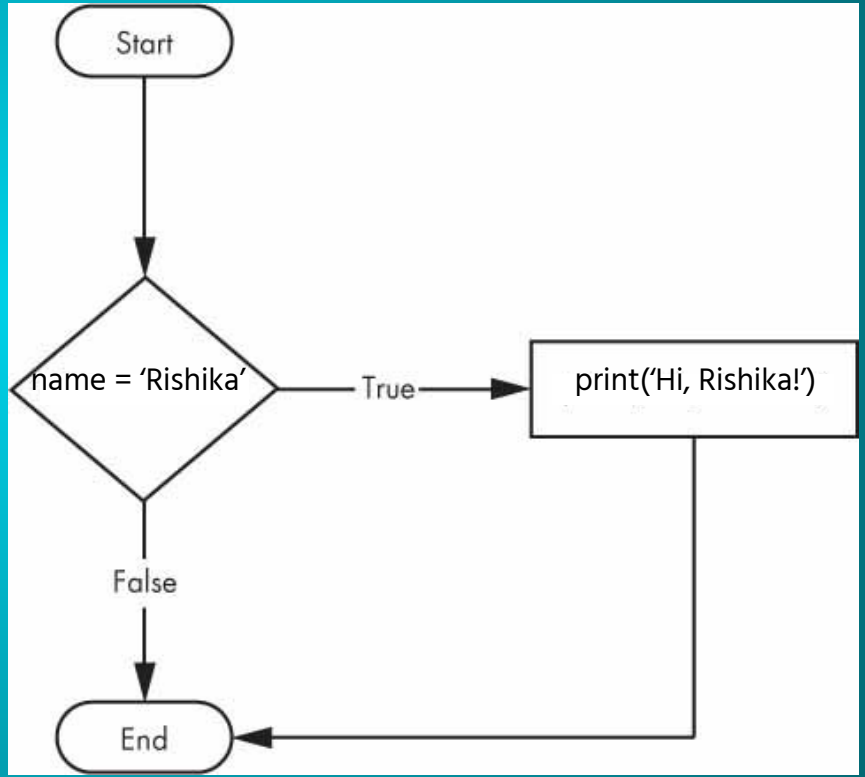


if statement

- Most common type of flow control statement
- Clause will execute if the condition is True, and is skipped if it is False
- Plain English: *“If this condition is true, execute the code in the clause.”*
- Consists of the following:
  - The if keyword
  - A condition (evaluates to True/False)
  - A colon
  - The if clause

Example: if statement

```
if name == 'Rishika':
    print('Hi, Rishika!')
```

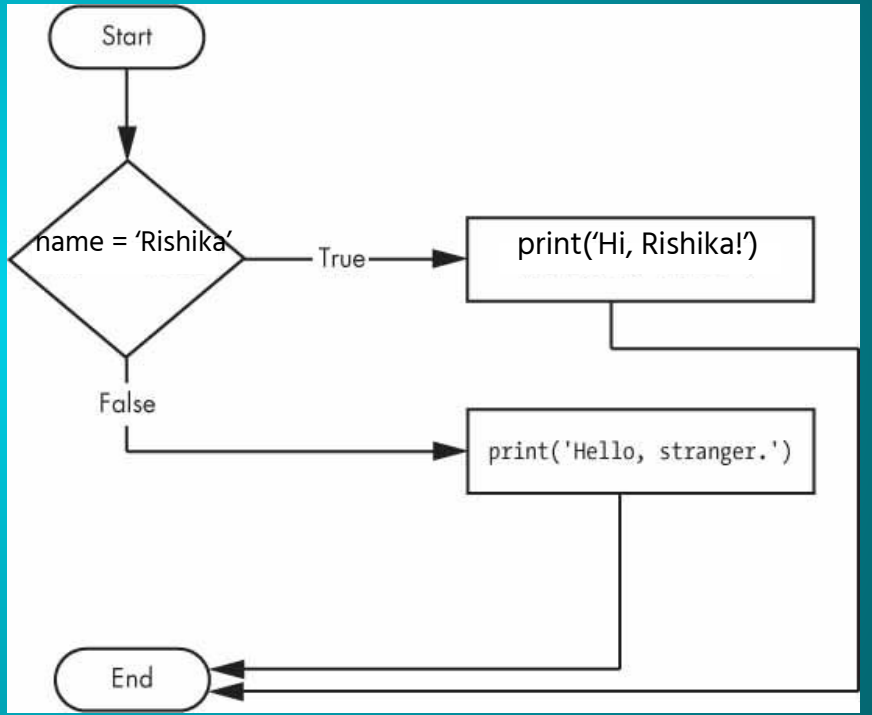


else statement

- Optionally follows an if statement
- Executed only when if statement's condition = False
- Plain English:
  - If this condition is true, execute this code. Or else, execute that code.
- *No condition*
- Consists of the following:
  - The else keyword
  - A colon
  - The else clause

Example: else statement

```
if name == ‘Rishika’:
    print(‘Hi, Rishika!’)
else:
    print(‘Hello, stranger.’)
```

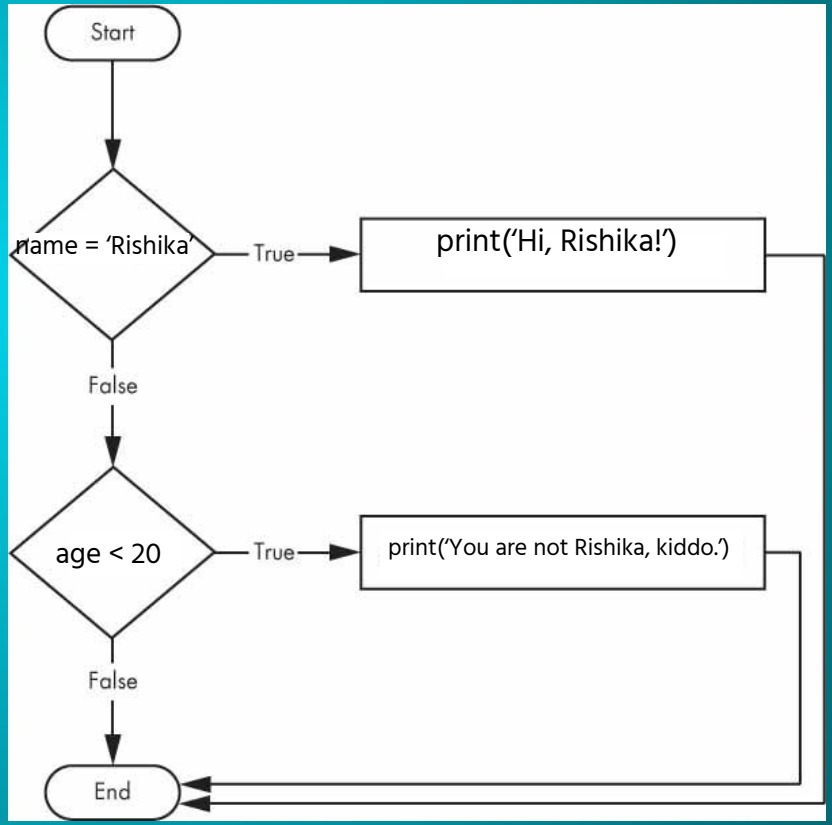


elif statement

- You want one of *many* possible clauses to execute
- elif = ‘else if’
- Always follows an if or another elif
- Condition that is checked if all previous conditions = False
- Consists of the following:
  - The elif keyword
  - A condition (True/False)
  - A colon
  - The elif clause

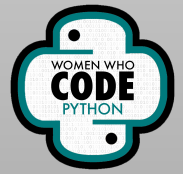
Example: elif statement

```
if name == 'Rishika':
    print('Hi, Rishika!')
elif age < 20:
    print('You are not Rishika, kiddo.')
```



while loop

- Execute code over and over again (executed as long as statement's condition = True)
- Consists of the following:
  - The while keyword
  - A condition (True/False)
  - A colon
  - The while clause
- Similar to if statement, different in how they behave
  - After if statement, execution continues on
  - After while loop, jumps back to the start



Example: while loop

```
spam = 0

if spam < 5:

    print('Hello, world.')

    spam = spam + 1
```

Output :“hello world” is printed **1 time only**

Example:

```
>>> spam = 0
>>> if spam < 5:
...     print("hello world")
...     spam = spam + 1
...
hello world
>>>
```

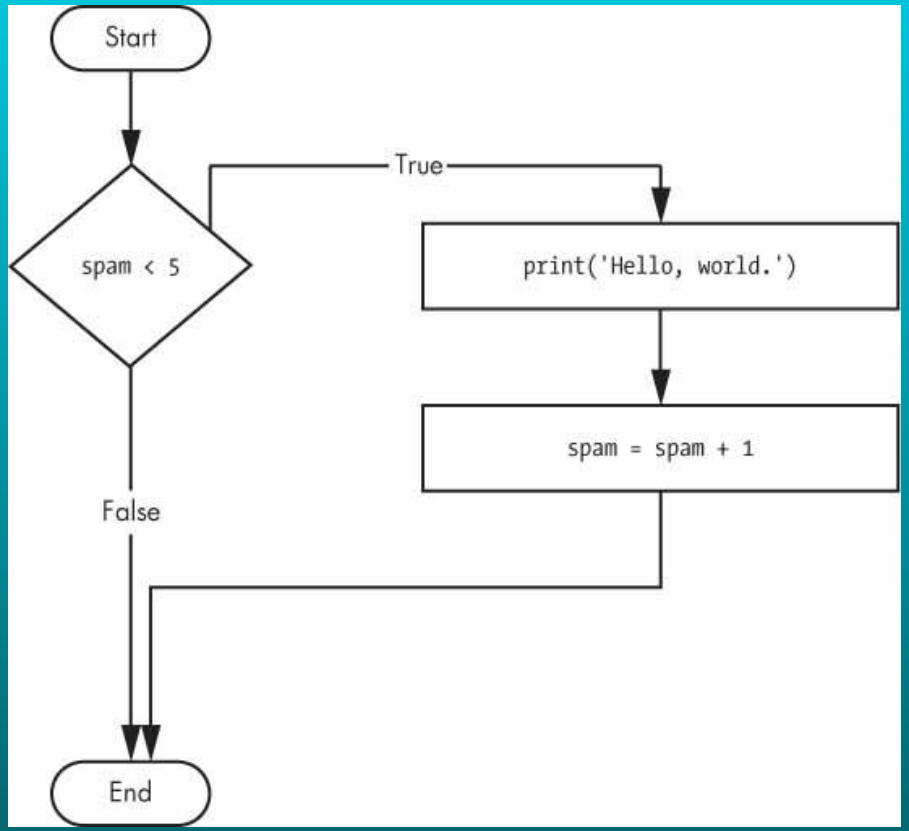
```
spam = 0

while spam < 5:

    print('Hello, world.')

    spam = spam + 1
```

Output: “hello world” is printed **5 times**

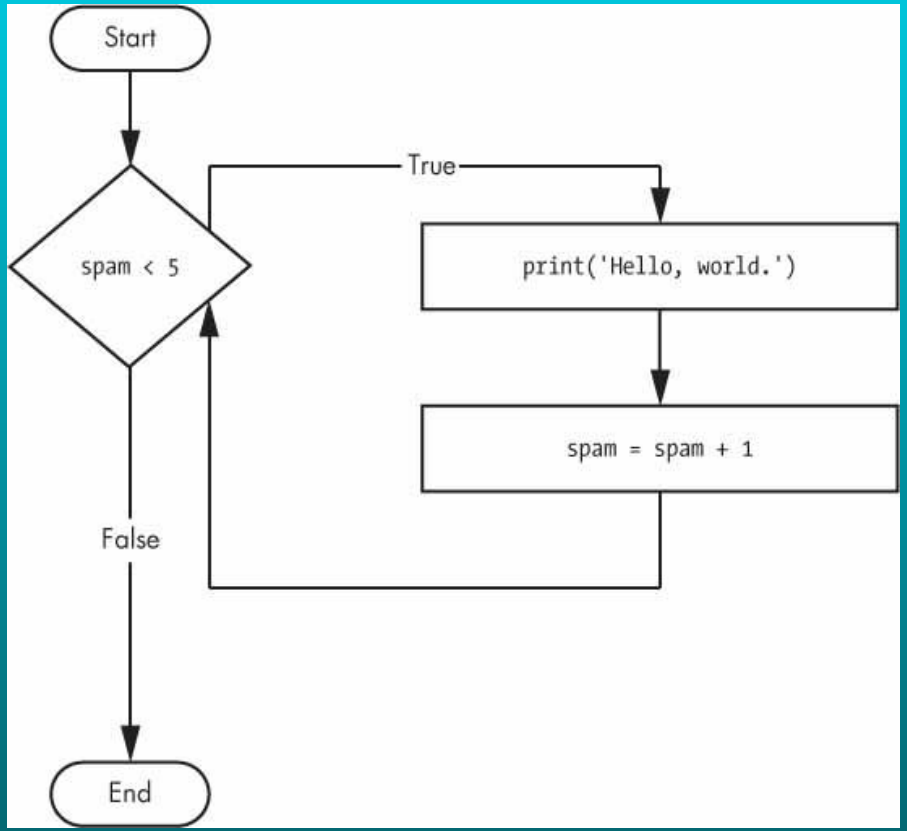
Example:

```
>>> spam = 0
>>> while spam < 5:
...     print("hello world")
...     spam = spam + 1
...
hello world
hello world
hello world
hello world
hello world
>>>
```

POLL TIME!


Start
spam < 5
True
print('Hello, world.')
spam = spam + 1
False
End

(A)


Start
spam < 5
True
print('Hello, world.')
spam = spam + 1
False
End

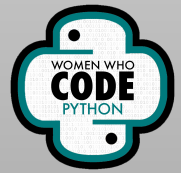
(B)

while loop: Additional statements

- break statement
  - Used directly inside the loop
  - Immediately exit the while loop's clause
  - Only need the break keyword
- continue statement
  - Also inside the loop
  - Immediately jump back to start of the loop and reevaluate the loop's condition

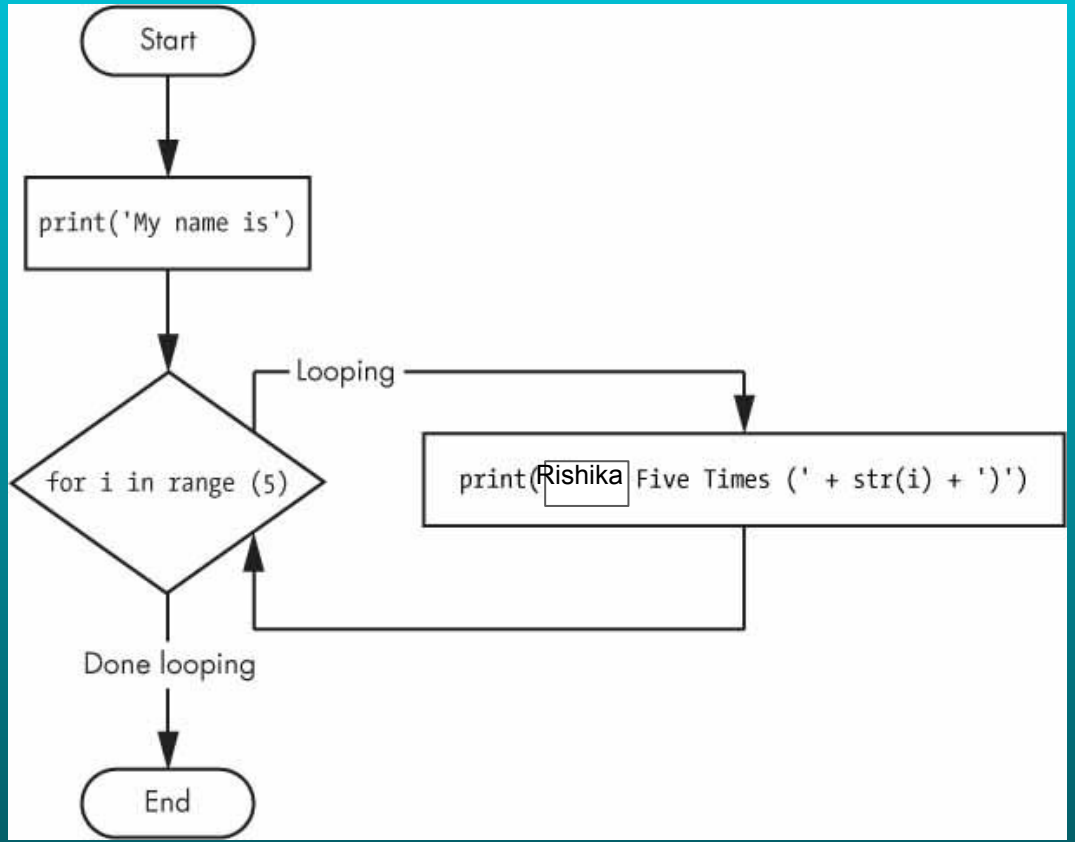
for loop

- Execute code for a specific number of times
  - while loop keeps looping
- Consists of the following:
  - The for keyword
  - A variable name
  - The in keyword
  - 2 options:
    - A call to the range() with up to 3 integers passed to it OR
    - An iterable that has already been declared
  - A colon
  - The for clause



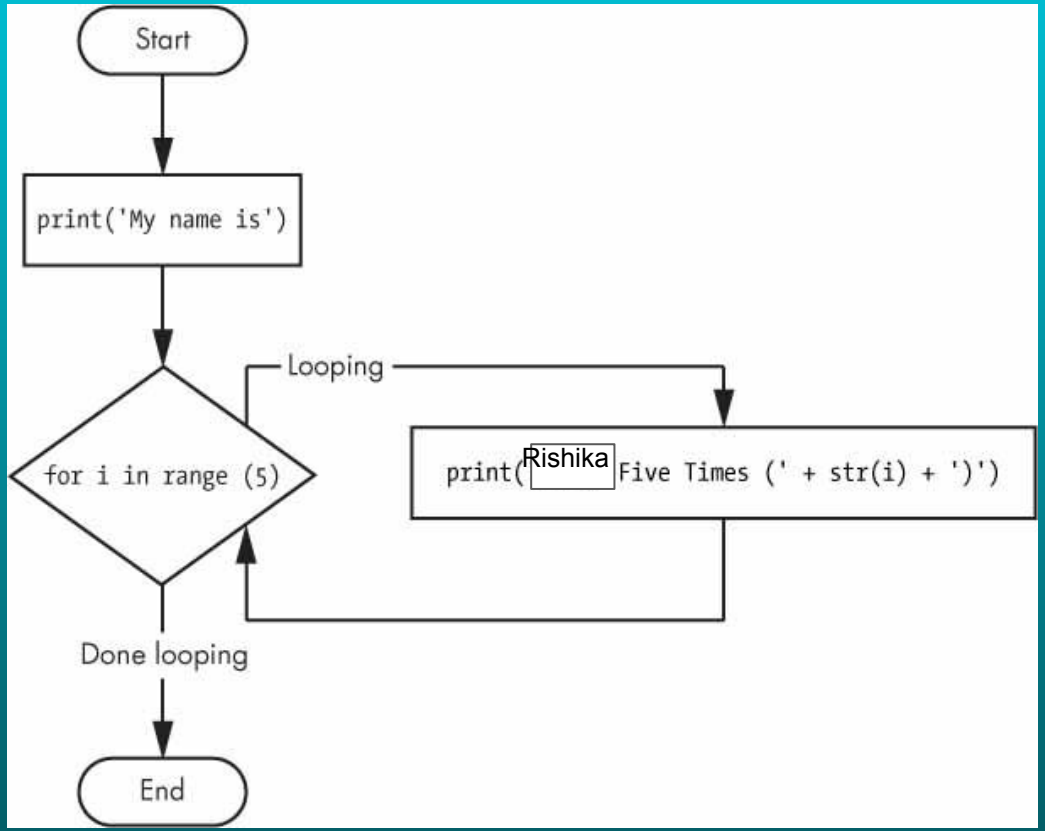
Example: for loop

```
print('My name is ')
for i in range(5):
    print('Rishika Five Times (' + str(i) + ')')
```

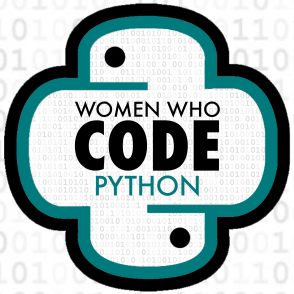


Equivalent while loop

```
print('My name is ')
i = 0
while i < 5:
    print('Rishika Five Times (' + str(i) + ')')
    i = i + 1
```

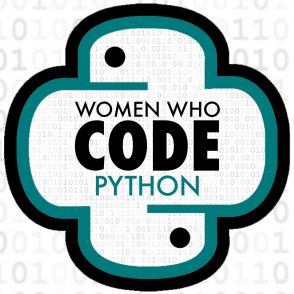


Q&A Time!



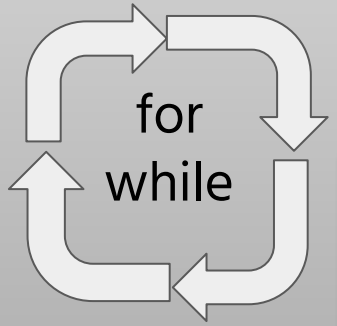
Time for Live Coding!

Google Colab link:
https://colab.research.google.com/drive/1unqDzoYdxrrF1sm1Ii_v4bd6ndQZilce?usp=sharing



Wrap-Up

- Programming Logic & Flow Control
- Next session: Open Q&A / Review Session
  - Google Form! https://forms.gle/RkNr4aZtT1cYzPhV8



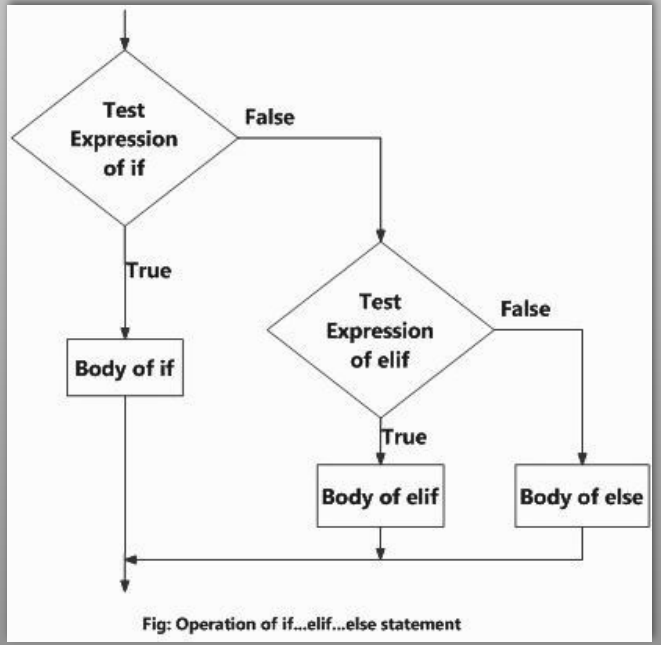
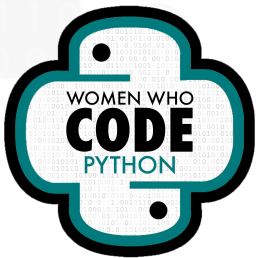


Fig: Operation of if...elif...else statement

Questions?

Join our Slack channel: #beginner-python-stdy-grp



Thanks, everyone!