HEALTH
UNIVERSITY OF UTAH

Biomedical Informatics

BMI 6018: Introduction to Programming
Module 4 Basic Logic, For Loops, Functions

Fatemeh Shah-Mohammadi, Rachel Nelson, Ram Gouripeddi

Department of Biomedical Informatics

University of Utah

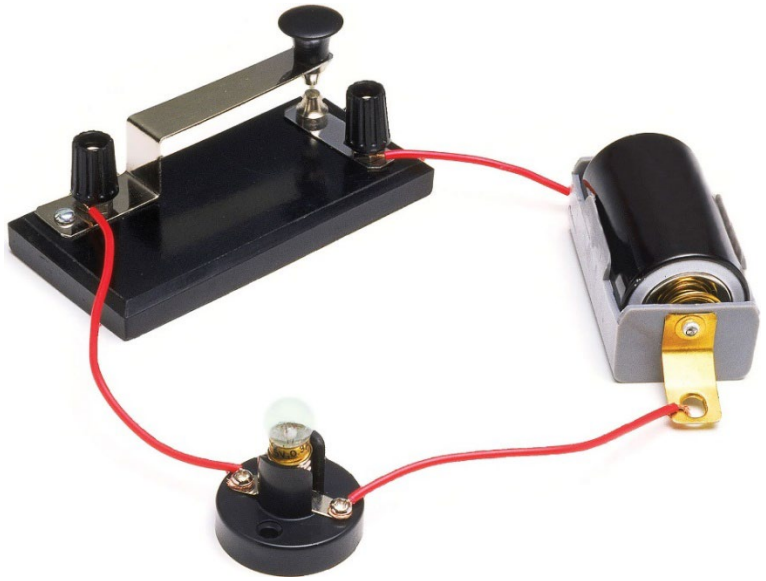
Module Goals

- Demonstrate and explain the basics of logical processing via
  - If, Else statements
  - For and while loops
  - Function syntax

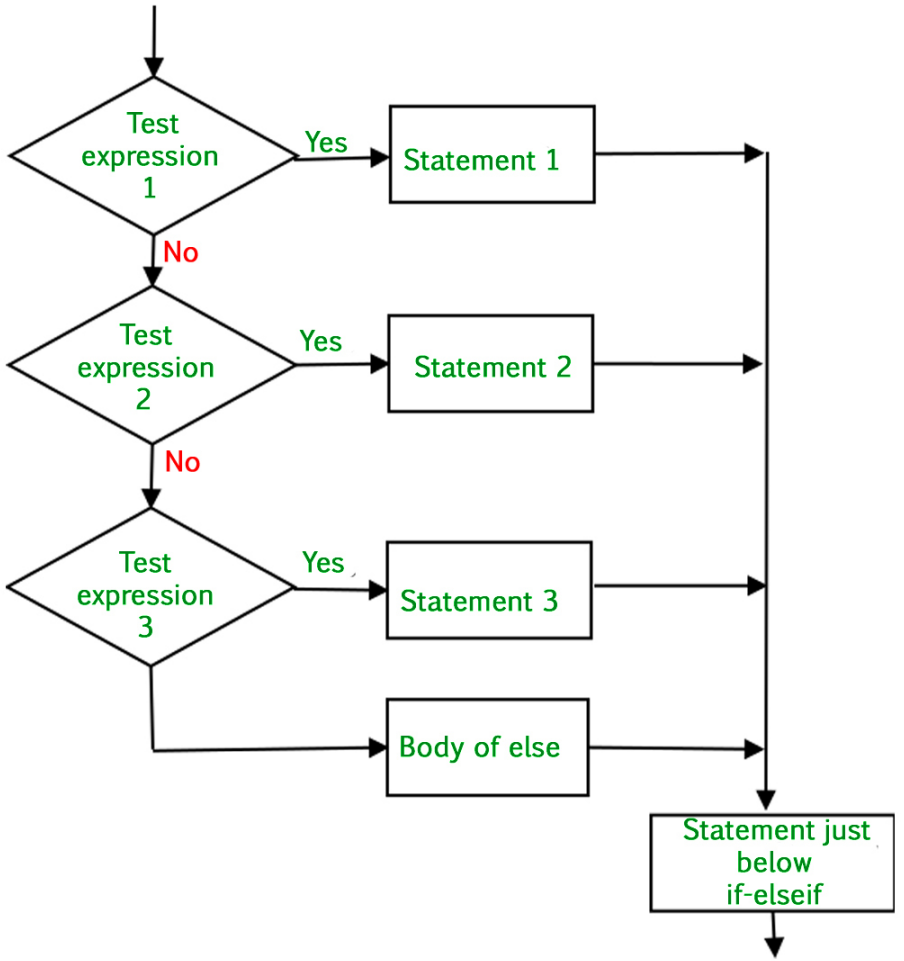
Boolean extensions and simple processing

As pseudo-code

- IF (Logical statement) evaluates to True:
  - Then Do something as a result
- ELSE IF (other statement) evaluates to True:
  - Then Do something else
- ELSE
  - Do a third thing



Graphically



https://www.geeksforgeeks.org/python3-if-if-else-nested-if-if-elif-statements/

Coding Intermission

BMI6018-M4-if.ipynb

Logical if/then review

- If, elif is the building block of logical processing
- You can and will extend this with Boolean processing (or, and, not etc)

Module Goals

- Demonstrate and explain the basics of logical processing via
  - If, then statements
  - **For and while loops**
  - Function syntax

For loops!

- Iterating over stuff comes up a lot…
- they are only defined for **collections** and therefore require an **index**.

What about while loops?

- While loops are a more general, but possibly more *dangerous* tool in your arsenal. Let's find out why…

Coding Intermission

- BMI6018-M4-Loops

But wait! There's more!

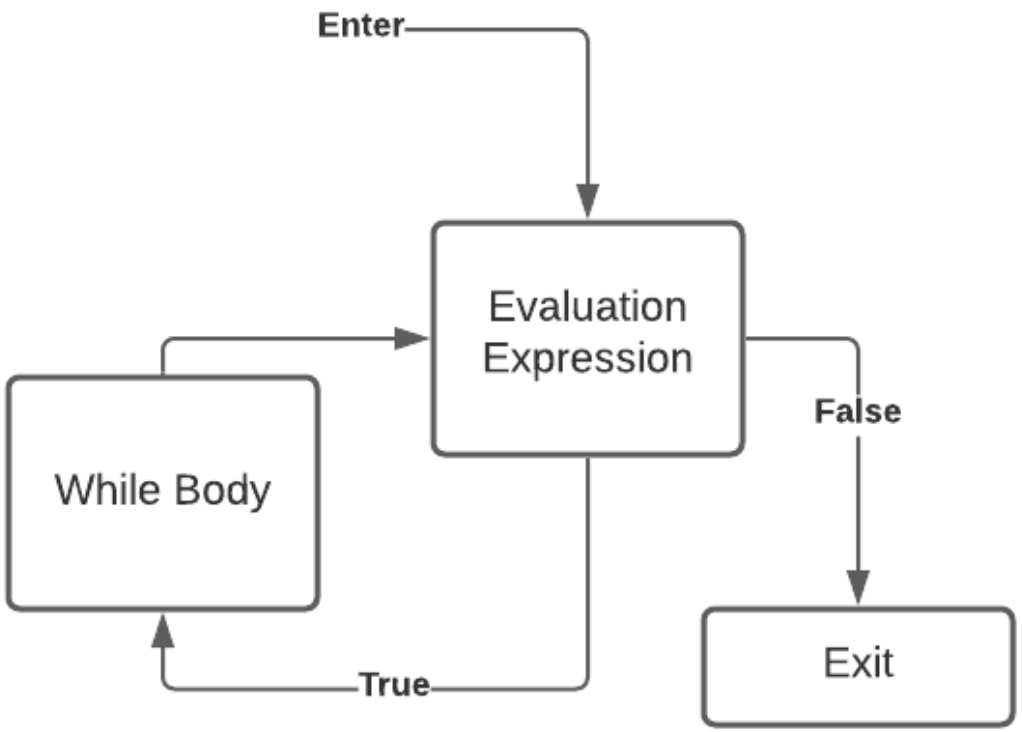
- Break
- Continue
- Enumerate

Coding Intermission

- BMI6018-M4-Loops again

So, are while loops bad?

- Nope! Just need to be more careful
- While loops are the basis for most user interfaces (UI)
  - ie while your cursor is not on a button, don't do anything!



Conclusion

- Loops and iteration form a core building block of programming.
- For loops require an index
- While loops require an evaluative statement
- They are both useful and important!

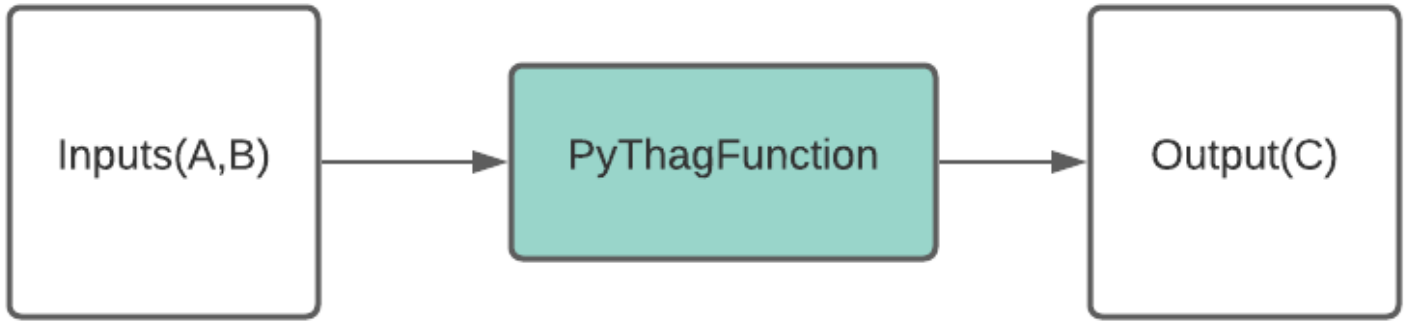
Module Goals

- Demonstrate and explain the basics of logical processing via
  - If, then statements
  - For and while loops
  - **Function syntax**

Why write Functions?

- Reusability
- Abstraction
- Readability
- Testing/Debugging

The value of such an abstraction



Code intermission!

- BMI6018-M4-functions.ipynb