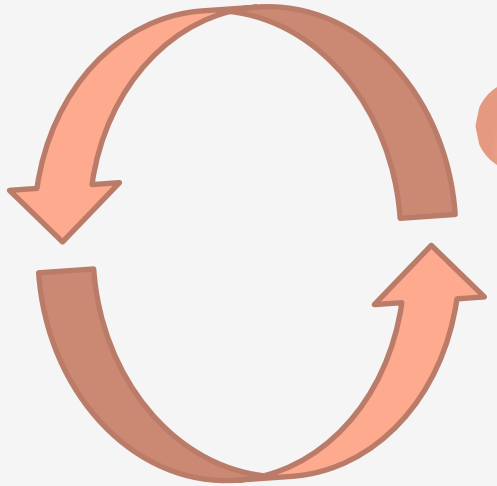
Loops



Lecture 13

Week 7

Let’s take a look at Loops

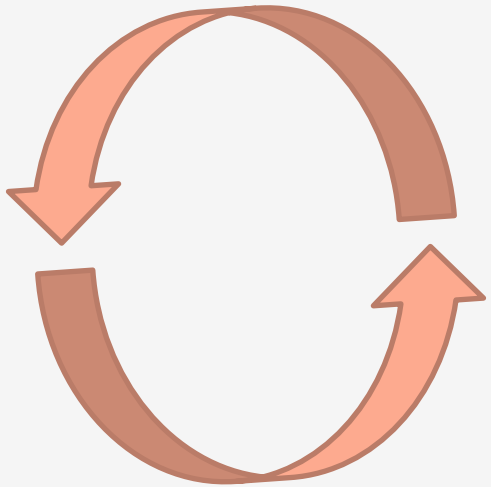
Repetition using a *while loop*

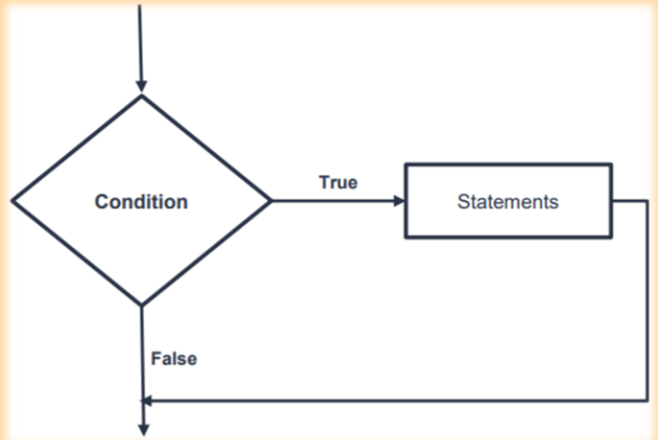
Accumulators

Looping through a string

Let’s take a look at Loops

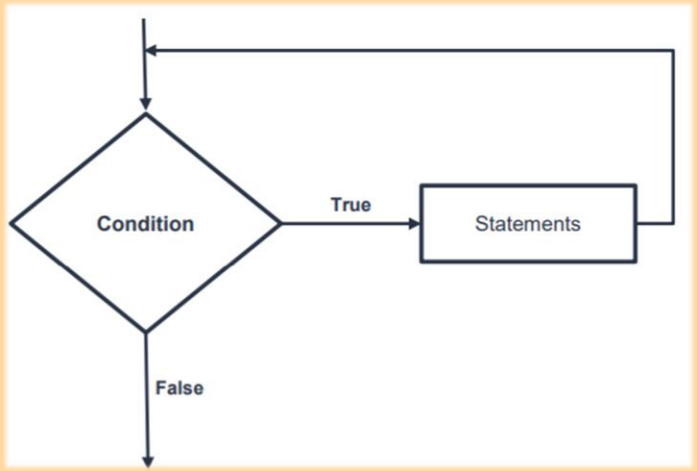
Loops allow us to write code that will be executed more than one.





if

If the condition is True, execute a block of statements and move on to the rest of the program.



loop

If the condition is True, execute a block of statements and check the condition again. When the condition becomes False, move on to the rest of the program.

A loop causes a statement or block of statements to repeat as long as a condition is True.

Loops are said to iterate

An iteration is one execution of the body of a loop

while loop

```
while condition:
        statement(s) to be executed
```

Loop example

```
1  i = 0
2  while i <= 5:
3      print(i)
4      i = i + 1
5  print("Finished the loop")
```

Control variable

i is initialised to 0
While i <=5, print its value and increase its value by 1 (increment i by 1)

Control variables

A control variable is used in the condition of a loop

The control variable must change during the loop .....

The control variable must be initialized before the loop

....otherwise you have an infinite loop and you don't want that

```
1  i = 0
2  while i <= 5:
3      print(i)
4      i = i + 1
5  print("Finished the loop")
```

- i is the traditional name of an iterator i.e. a control variable that counts.
- Change to count from 0 to 20 in steps of 2
- Change to count from 20 down to 0 in steps of 2
- How would this be changed to print "Hello" 3 times?

User input control

Iterators allow us to run a loop a certain number of times

User input can also be used to dynamically control loop execution

Loop example

```
1  number = int(input("Please enter a number "
2                     "(0 to finish)>"))
3
4  while number != 0:
5      print(f"You entered {number} - thank you")
6      number = int(input("Please enter a number "
7                         "(0 to finish)>"))
8
9  print("Loop complete")
```

Control variable

Loop ends when the user enters 0

While loop

The while loop is a pretest loop

- Tests the condition before performing an iteration
- Will never execute if the condition is false to start with – zero-trip behaviour

answer is the control variable

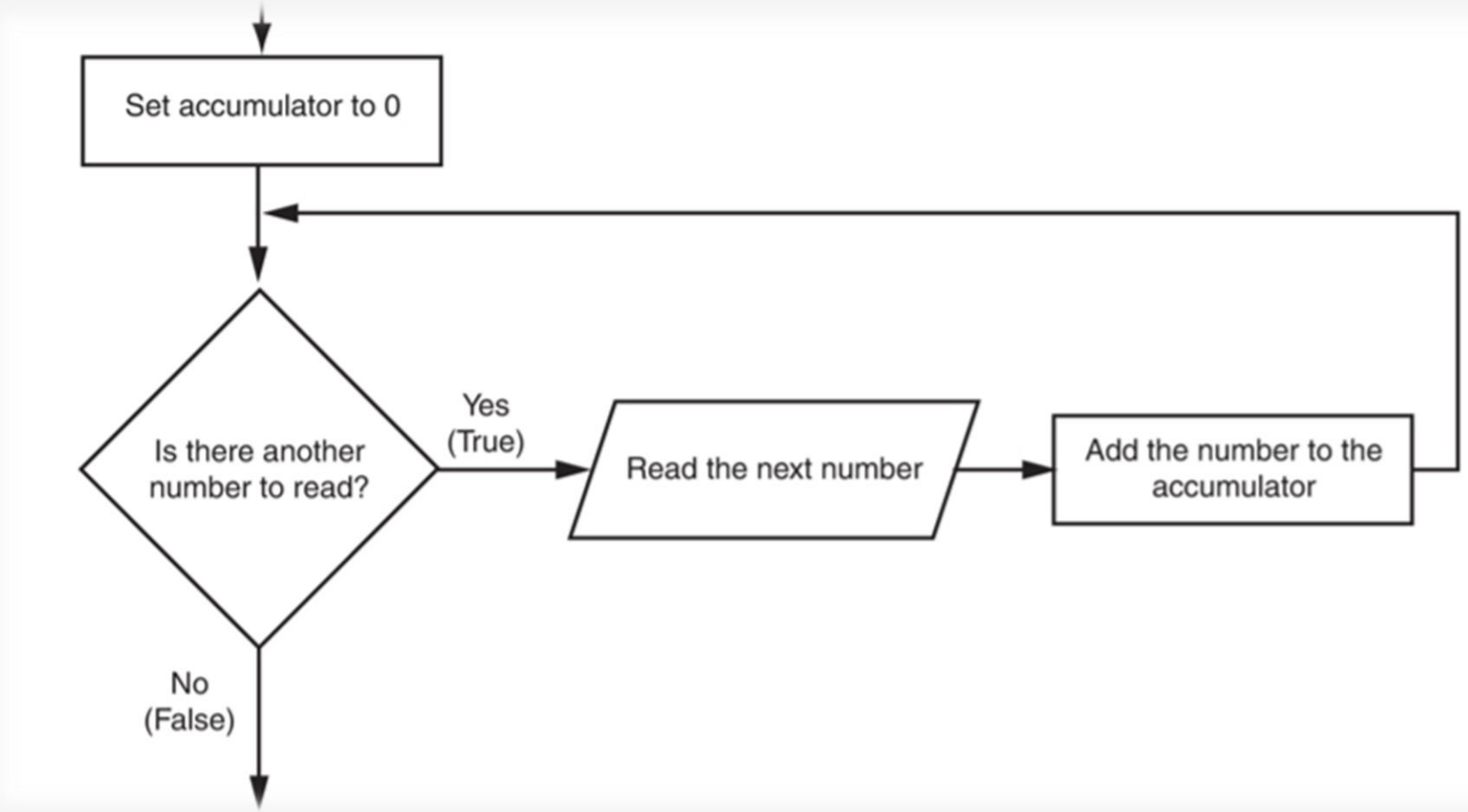
```
# Author: Alison
# Purpose: while with a control variable

answer = input("Do you have a number (y/n)? ")

while answer == 'y':
    number = int(input("Please enter a number >>> "))
    answer = input("Do you have another number (y/n)? ")

print("Thank you for using my application.")
```

Try it yourself

Write code to create an accumulator

- [ ] Create a variable, which starts at 0, to hold the total
- [ ] Ask the user for a number
- [ ] Add the number to the total
- [ ] Repeat until the user runs out of numbers
- [ ] Display the total

Logic for calculating a running total



Try it yourself
Bouncing ball application

- Write an application that asks the user for the starting height of a bouncing ball.
- At each bounce the ball halves it's height.
- Display the height and number of each rebound bounce stopping when the ball is 1mm off the ground.