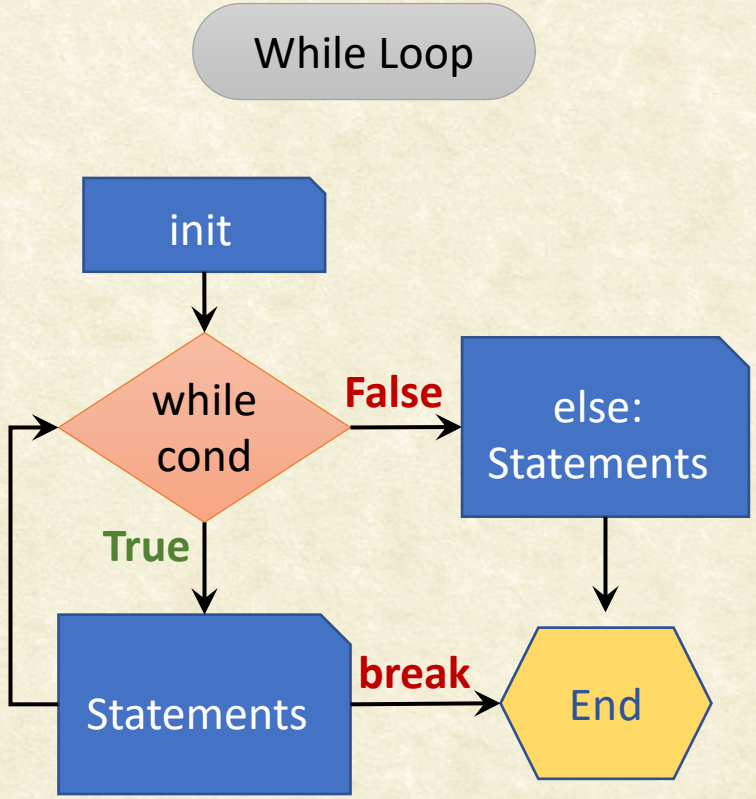
Lecture 4: Loop, Strings



Anoop M. Namboodiri

IIIT Hyderabad

if: The Conditional Statement [Recap]

```
if   boolean-expression:
     statements
elif   boolean-expression:
     statements
else:
     statements
```

- elif stands for else if
- Evaluates the expressions one by one until one is found to be true; then that set of statements (suite) is executed (and no other part of the if statement is executed or evaluated). If all expressions are false, the statements of the else clause, if present, is executed.

if Statement: Examples [Recap]

```
if num % 10  == 0:
      print("Number is a multiple of 10”)
elif num % 2:
      print("Number is odd”)
else:
      print("Number is even”)

if a <= 5 and c >= 10 or d == “done” and b != 5:
      print(“Right Conditions”)

If  8 <= Time <= 17:
      print(“Working Time”)
```

Let us Code: if [Recap]

- Write a code that prints if a number is divisible by 11 or not

```
if num % 11:
    print(“num is not divisible by 11”)
else:
    print(“num is divisible by 11”)
```

- Write a code that decides if a floating point number, x, is in the open interval (a,b)

```
if a < x < b:
    print(f"{x} is in the open interval ({a},{b})")
else:
    print(f"{x} is outside ({a},{b})")
```

Coding Practice: if

1. Write the code that decides if an object is an integer or a float

02

2. Given a circle (center, (a, b), and radius r), decide is a given point, (x, y) is within the circle or not.

03

3. Given two integers, write a code that decides if one is a factor of the other or not

04

Python Loops

While: The basic loop

While Loop

```
while  boolean-expression:
      statements
else:
      statements
```

- The statements in the while block are executed as long as the boolean expression is true. If it becomes false (which might even be the first time it is evaluation), the statements in the else block (if present) are executed and the loop terminates.

Let us Code: while

- Write a code that decides if a number is prime or not

```
prime = True
divisor = 2
while divisor < num:
    if num % divisor == 0:
        prime = False
        break
    divisor += 1
if prime:
    print("Number is prime")
else:
    print("Number is not prime")
```

Coding Practice: while

1. Improve the efficiency of the program to test whether a number is prime or not. 05
2. Write a code to print the first n numbers in the Fibonacci sequence 06
3. Write a code to print all prime factors of a number. Print each factor only once. 07

Python Strings

A Native Data Type

String: A Popular Sequence Type

- Specified by enclosing characters between two ', ", ''', or """. (Last two are multiline)
- No Character Type; They are strings of length 1.
- \n and \t denotes newline and tab within a string
  - This is not the case in raw strings: `r'string'`
- String is an immutable type. i.e., Like numbers, a string object cannot be changed, once it is created.

String Operations

- str[ind]: get character at ind.
  - Indices go from 0 to n-1
  - -1 refers to last character, -2 to second last and so on.
- str[a:b]: get substring from a to b-1.
  - a is 0 if not specified: str[:b]
  - b is len(str) if not specified: str[a:]
- + is concatenation
  - ‘Hello’ + ‘World’ is ‘HelloWorld’
- * is repetition
  - ‘Ha’ * 3 is ‘HaHaHa’
- c in str: returns true if character c is present in str

Methods of String

- str.find(substr, beg=0 end=len(str)):
  - Returns index if found; -1 otherwise
- str.count(substr, beg= 0,end=len(str)):
  - Counts number of times substr occurs in str
  - ‘abababababa’.count(‘aba’) is 2
- str.replace(old,new): replaces all old with new
  - Returns a modified string (original is untouched)
- isalpha(), islower(), isupper(), endwith(), startswith()
  - Returns True or False
- capitalize, upper(), lower(), strip(), translate()
  - Returns a modified string (original is untouched)
- Many others.. see the Python Language Reference

Let us Code: String

- Write a code that reads in a string (path) and if the path ends in a /, removes it

```
path = input('Enter path: ')
path = path.rstrip()                # why this?
if path.endswith('/'):              # OR path[-1]
    path = path[:-1]
```

- Write a code that reads in a string (path) and replaces all '\' with '/'

```
path = raw_input('Enter path: ')
path = path.replace('\','/')
```

Coding Practice: String

1. Write the code that reads in two strings a path and a filename and joins them. While doing this, check for trailing ‘/’ in the path
2. If C: of a windows machine is mounted at /win/c in a linux system, translate windows path to unix
3. Write a code that takes a string as input and prints if it is a palindrome or not