CIS 2348 UNIVERSITY OF HOUSTON INFORMATION SYSTEM APPLICATION DEVELOPMENT

FALL 2021

CHAPTER 6

LOOPS

LOOPING

- LOOPING ALLOWS TO EXECUTE THE SAME PART OF THE CODE MULTIPLE TIMES
- WE CAN CONTROL HOW MANY TIMES WE EXECUTE THE CODE
- WE CAN PROGRAMMATICALLY DECIDE WHEN TO EXIT A LOOP
- WE CAN KEEP TRACK OF HOW MANY TIMES WE HAVE EXECUTED THE CODE SO FAR

WHILE STATEMENT

- SYNTAX:

WHILE CONDITION:

    ACTION BLOCK

THIS LOOP WILL KEEP EXECUTING THE ACTION CODE BLOCK WHILE THE CONDITION EXPRESSION IS TRUE

WHILE CONDITIONALS (STANDARD USE)

- SHOULD CONTAIN VARIABLE OR VARIABLES!
- THE VARIABLES NEED TO BE INITIALIZED BEFORE THE WHILE LOOP STARTS
- AT LEAST ONE OF THE VARIABLES NEEDS TO BE MODIFIED WITHIN THE ACTION CODE BLOCK

- OTHERWISE -- AN INFINITE LOOP WILL RESULT – NOT NORMALLY A GOOD THING
  - SOMETIMES CAN USE AN INFINITE LOOP WITH A BREAK STATEMENT – NOT USUALLY GOOD PRACTICE

WHILE LOOP EXAMPLE

```
X=0                          #INITIALIZES THE VARIABLE OUTSIDE THE LOOP
WHILE X<3:                   #THE CONDITIONAL EXPRESSION CONTAINS THE LOOP VARIABLE
        PRINT(X)
        X=X+1                #THE VARIABLE IS MODIFIED IN THE CODE BLOCK
```

0

1

2

WHILE LOOP WITH SENTINEL

- SENTINEL VALUE IS VARIABLE VALUE THAT INDICATES THE LOOP SHOULD EXIT

```
#PRINT SQUARE ROOTS, -1 EXITS

IMPORT MATH

X=0

WHILE X!=-1:

        IF (X!=0):

                PRINT('THE SQUARE ROOT IS ', MATH.SQRT(X))

        X=INT(INPUT('TYPE YOUR WHOLE NUMBER'))
```

COUNTING IN A WHILE LOOP

```
#PRINT THE FIRST N PERFECT SQUARES

N=77

I=0

WHILE I<N:

        I+=1

        PRINT(I**2)
```

COUNTING IN A WHILE LOOP

```
#PRINT THE FIRST N PERFECT SQUARES

N=77

I=1

WHILE I < N+1:

        PRINT(I**2)

        I+=1
```

FOR STATEMENT

- SYNTAX

FOR VARIABLE(S) IN GROUP:

ACTION CODE BLOCK

LOOP AND PERFORM THE ACTION IN THE ACTION CODE BLOCK A FIXED NUMBER OF TIMES BASED ON THE GROUP MEMBERSHIP

FOR LOOP USAGE GUIDE

- LOOP VARIABLE NOT INITIALIZED BEFORE THE LOOP
- LOOP VARIABLE SHOULD NOT BE ALTERED WITHIN THE ACTION CODE BLOCK
- THE GROUP SHOULD NOT BE ALTERED WITHIN THE CODE BLOCK

LOOP EXAMPLE

```
NAME_LIST =['BOB', 'ALICE', 'JOHN']
FOR NAME IN NAME_LIST:
        PRINT('THE NAME IS ',NAME)
        L=NAME[0]
        PRINT('THE FIRST LETTER OF ', NAME, ' IS ', L)
```

REVERSED FUNCTION

```
NAME_LIST =['BOB', 'ALICE', 'JOHN']
FOR NAME IN REVERSED(NAME_LIST):
        PRINT('THE NAME IS ',NAME)
        L=NAME[0]
        PRINT('THE FIRST LETTER OF ', NAME, ' IS ', L)
```

ITERATING THROUGH A SET

```
NAME_SET ={'BOB', 'ALICE', 'JOHN'}
FOR NAME IN NAME_SET:
        PRINT('THE NAME IS ',NAME)
        L=NAME[0]
        PRINT('THE FIRST LETTER OF ', NAME, ' IS ', L)
```

RANGE FUNCTION

- ALLOWS ITERATION THROUGH A RANGE OF INTEGERS
- RANGE(N) == [0,..,N-1] I.E. RANGE(3)==[0,1,2]
- RANGE(M,N)==[M,..,N-1] I.E. RANGE(-10,-8)==[-10,-9]
- RANGE(M,N,S)==[M,M+S,…,M+KS<N] I.E. (1,6,2)==[1,3,5]

- EASY ENUMERATION OF INTEGERS

RANGE EXAMPLE

```
FOR I IN RANGE(10, -10,-1):
        IF ( I%3 == 0 ):
                PRINT(I, ' IS DIVISIBLE BY 3')
```

CAN ITERATE "BACKWARDS"

NESTED FOR LOOPS

```
FLAVOR_TYPE ={‘CHERRY’, ‘APPLE’, ‘PEACH’}

DESSERT_TYPE ={‘PIE’, ‘COBBLER’}

FOR FLAVOR IN FLAVOR_TYPE:

        FOR DESSERT IN DESSERT_TYPE:

                PRINT(‘I LIKE ‘,FLAVOR, DESSERT)
```

ENUMERATE FUNCTION

GIVES ACCESS TO BOTH ELEMENT AND INDEX OF THE ELEMENT IN THE LIST, USEFUL FOR KEEPING TRACK OF WHERE YOU ARE IN THE LIST

SYNTAX:

```
FOR (INDEX, ELEMENT) IN ENUMERATE(LIST):
        ACTION BLOCK
```

ENUMERATE EXAMPLE

```
FOR (INDEX, I) IN ENUMERATE(RANGE(-7, -199, -22)):
        PRINT('THE ELEMENT IS ', I)
        SIZE_OF_LIST=LEN(RANGE(-7, -199, -22))
        PERCENTILE=(INDEX+1)/SIZE_OF_LIST*100.00
        PRINT('WE ARE ', PERCENTILE, '% THROUGH OUR TASK')
```

BREAK AND CONTINUE

- CONTROLS THE FLOW OF THE CODE THROUGH THE LOOP
- CONTINUE TERMINATES THE GIVEN ITERATION AND CONTINUES THE LOOP
- BREAK TERMINATES THE ENTIRE LOOP
- CONDITIONAL DEPENDENT FLOW
- USE ALMOST EXCLUSIVELY WITH AN IF STATEMENT

CONTNUE EXAMPLE

```
FRIENDS_LIST=['PEGGY','ENRIQUE','ED','TAMMY']
FOR NAME IN FRIENDS_LIST:
        IF NAME=='ENRIQUE':
                CONTINUE
        PRINT('I LIKE ',NAME)
PRINT('ALL DONE')
```

I LIKE PEGGY

I LIKE ED

I LIKE TAMMY

ALL DONE

BREAK EXAMPLE

```
FRIENDS_LIST=['ALICE','BOB','JOE','TAMMY']
FOR NAME IN FRIENDS_LIST:
        IF NAME=='BOB':
                BREAK
        PRINT('I LIKE ',NAME)
PRINT('ALL DONE')
```

I LIKE ALICE

ALL DONE

BAD CODE EXAMPLE – INFINITE LOOP

```
WHILE 1>0:
        NUMBER=INT(INPUT('TELL ME YOUR NUMBER, -1 TO EXIT))
        PRINT(NUMBER)
        IF NUMBER==-1:
                BREAK
```

SHOULD BE CODED WITH SENTINEL VALUE

```
NUMBER=0
WHILE NUMBER != -1:
        NUMBER=INT(INPUT('TELL ME YOUR NUMBER, -1 TO EXIT))
        PRINT(NUMBER)
```

FOR … ELSE CONSTRUCTION

EXECUTES AN ACTION IF THE LOOP IS EXITED "NORMALLY" I.E NOT WITH A BREAK STATEMENT

```
FRIENDS_LIST=['ALICE','BOB','JOE','TAMMY']
FOR NAME IN FRIENDS_LIST:
        IF NAME=='BOB':
                BREAK
        PRINT('I LIKE ',NAME)
ELSE:
        PRINT('ALL DONE')
```

WHILE ELSE …EXAMPLE

```
I=100
WHILE I>0:
        NAME = INPUT('PICK A NAME, TYPE Q TO QUIT')
        IF NAME=='Q':
                BREAK
        PRINT('YOUR CHOICE IS ',NAME)
        I-=1
ELSE:
        PRINT('COMPLETED 100 NAMES')
```