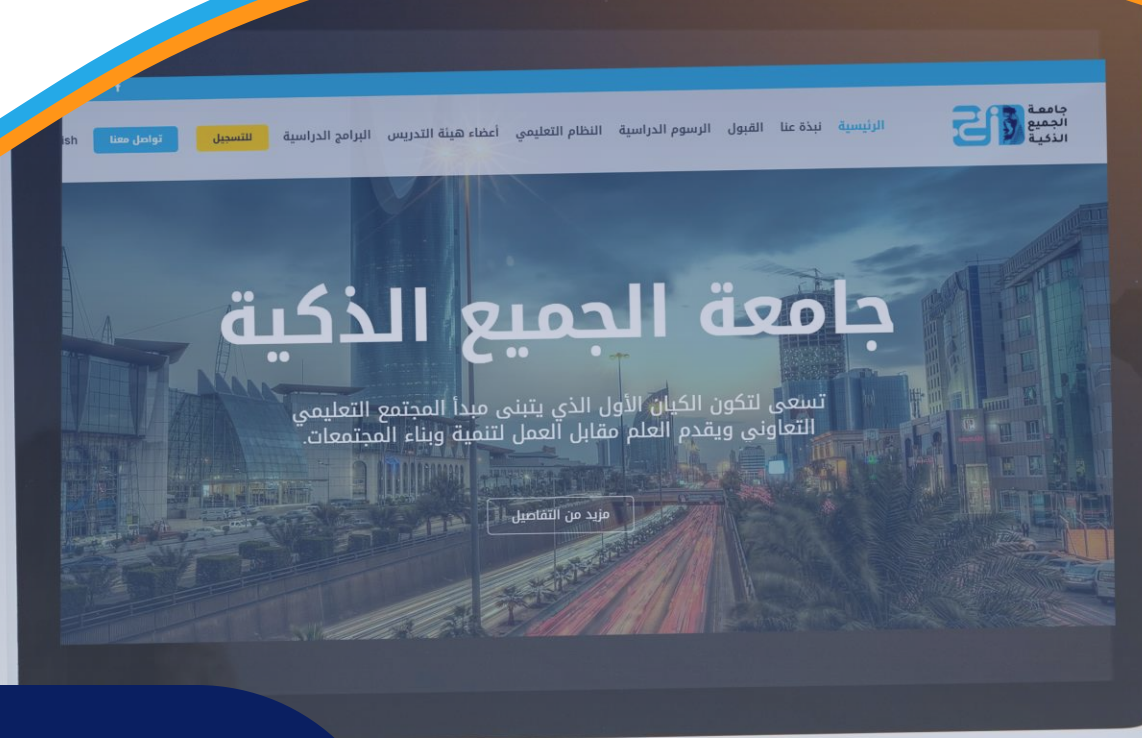




جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY



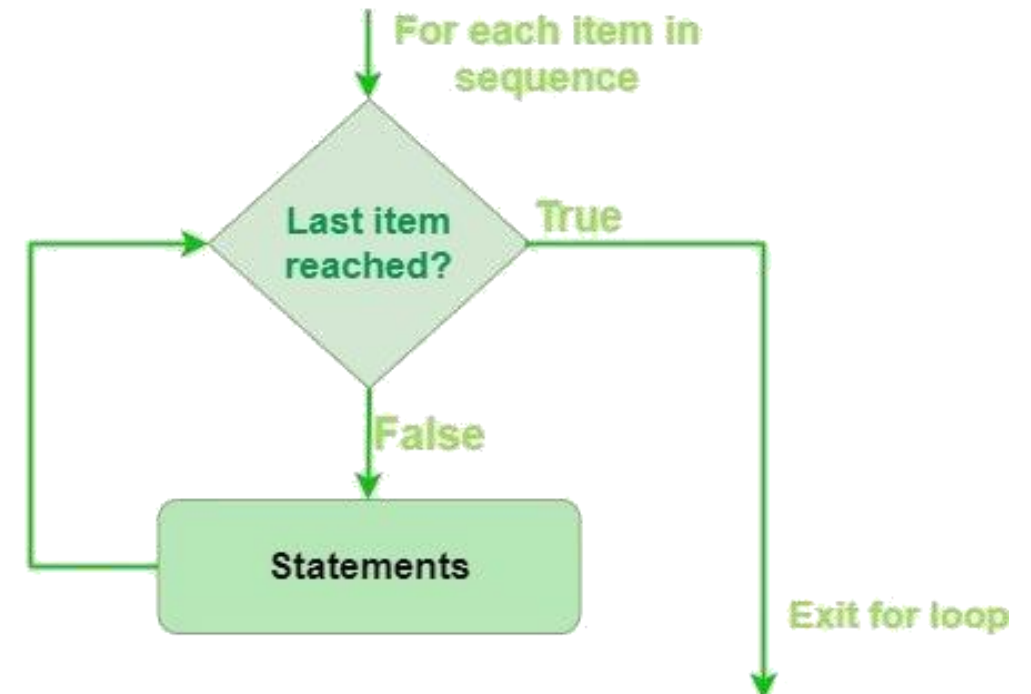
Iteration Statements and Strings
Prepared by Dr. Hamada I. AbdulWakel
Lecture 4

CYB550
Programming Foundations In
Python

- Conditional iterations.
- Different forms of iteration statements.
- Deal with strings.

- **An iteration statement, or loop**, repeatedly executes a statement, known as the loop body, until the controlling expression is False.
- The control expression must have a **scalar type**.
- The **while statement** evaluates the control expression before executing the loop body.
- The **for statement** is used to repeat the execution of a set of statements for every element of a sequence.

- **For loop.**
 - Syntax
for <variable> **in** <sequence> :
 <statement 1>



- Write a program that reads the number of students in a class then reads their marks and finds and displays the average.
 - **Help:**
 - Get "stdCount" variable from the user.
 - Define "sum" variable for the total of the marks.
 - Inside a loop, get each "mark" from the user.
 - Add "mark" to the "sum".
 - After loop finished, divide the "sum" by the "stdCount".

- Write a program that reads the number of students in a class then reads their marks and finds and displays the average.

- Code:

```
stdCount = int(input("Enter number of students in a class"))
```

```
sum = 0.0
```

```
for i in range(stdCount):
```

```
    mark = float(input("Enter mark: "))
```

```
    sum += mark
```

```
print("The marks average = ", sum / stdCount)
```



- Write a program that calculates the total of the following series of numbers:
 $1/30 + 2/29 + 3/28 + \dots + 30/1$
 - **Help:**
 - From the formula, identify the number of iterations.
 - Define the "sum" variable for the summation.

- Write a program that calculates the total of the following series of numbers:
 $1/30 + 2/29 + 3/28 + \dots + 30/1$

- Code:

```
min = 1;  max = 4;  sum = 0.0
```

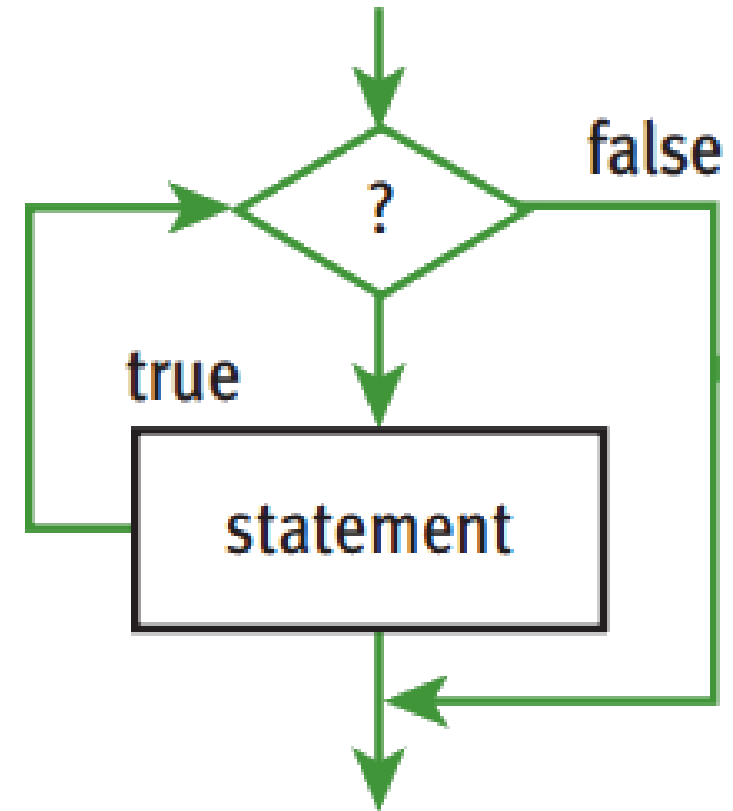
```
for i in range(min, max+1):
```

```
    sum += min / max;  min += 1;  max -= 1
```

```
print(sum)
```



- **While** (entry-control) loop.
 - Syntax
while <condition>:
 <statement 1>



- Write a program that calculates the average of a set of temperature. Data is terminated by the value 999.
 - **Help:**
 - Define the "sum" variable.
 - Define the "count" variable.
 - Get the "temperature" values from the user.
 - Inside a loop, sum the temperatures and count how many temperature values are entered.
 - After the loop finished, divide the sum by the count.

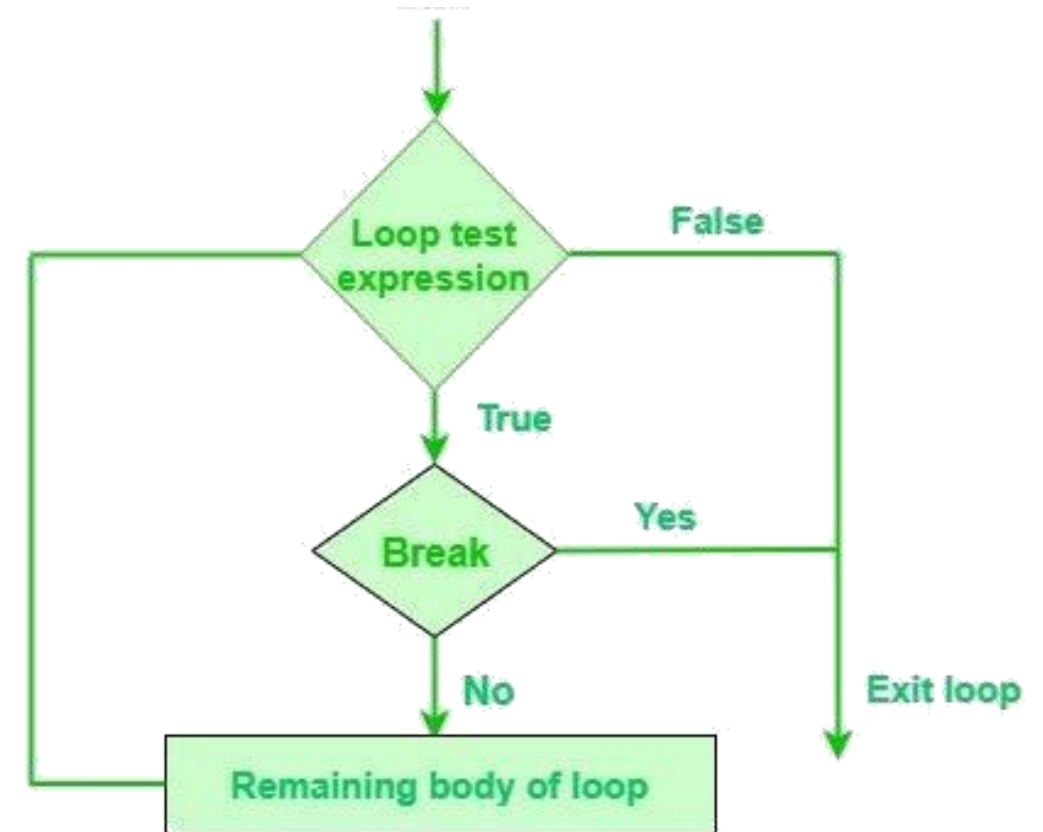
- Write a program that calculates the average of a set of temperature. Data is terminated by the value 999.

- Code:

```
sum = 0.0;    count = 0
temp = float(input("enter a temperature "))
while temp != 999:
    count += 1;    sum += temp
    temp = float(input("enter another temp"))
print("temperature average = ", sum/count)
```



- **Break** statement.
 - Syntax
while <condition>:
 <statement 1>
if <condition>:
 break



- Write a program to print numbers from 1 to 10. The program may stop in when 5 is reached.

- Code:

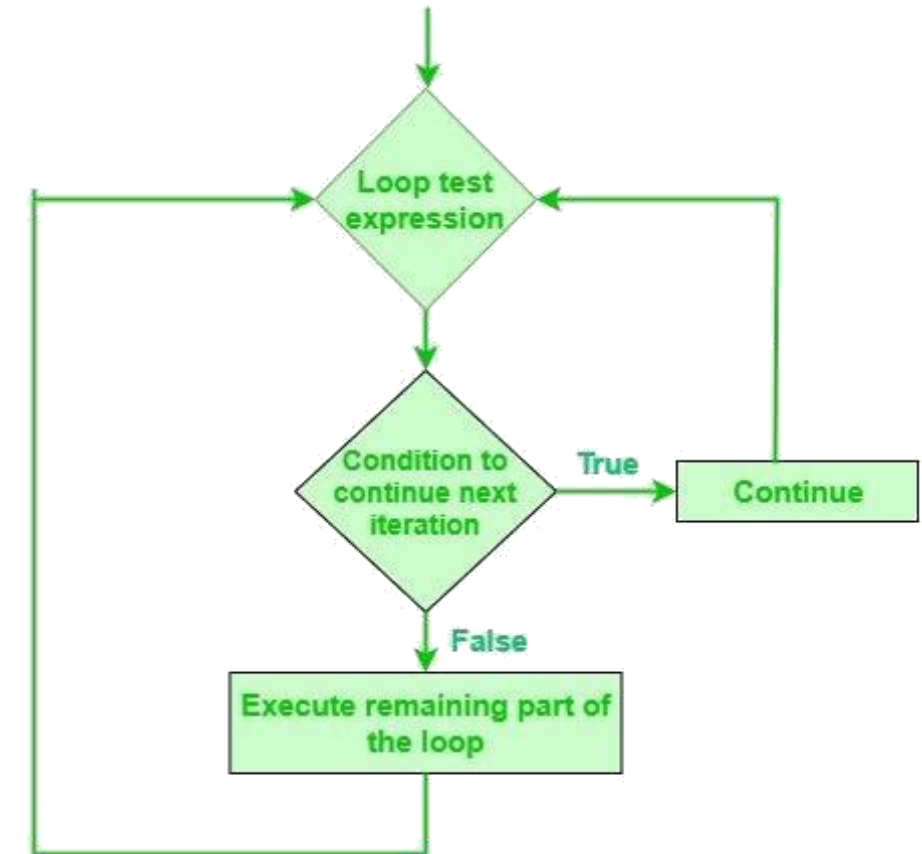
```
count= 0
while count < 10:
    if count == 5:
        break
    print(count)
    count += 1
```

- **Continue** statement.

- Syntax

while <condition>:
 <statement 1>

if <condition>:
 continue



- Write a program to print numbers from 1 to 10 except 5.

- Code:

```
count= 0
```

```
while count < 10:
```

```
    if count == 5:
```

```
        continue
```

```
    print(count)
```

```
    count += 1
```



- Write a program that calculates the occupancy rate for a hotel. The program should start by asking the user how many floors the hotel has. A loop should then iterate once for each floor. In each iteration, the loop should ask the user for the number of rooms on the floor and how many of them are occupied. After all the iterations, the program should display how many rooms the hotel has, how many of them are occupied, how many of them are unoccupied. The percentage may be calculated by dividing the number of rooms occupied by the number of rooms.



- **Do-While** loop not in python but can emulate.

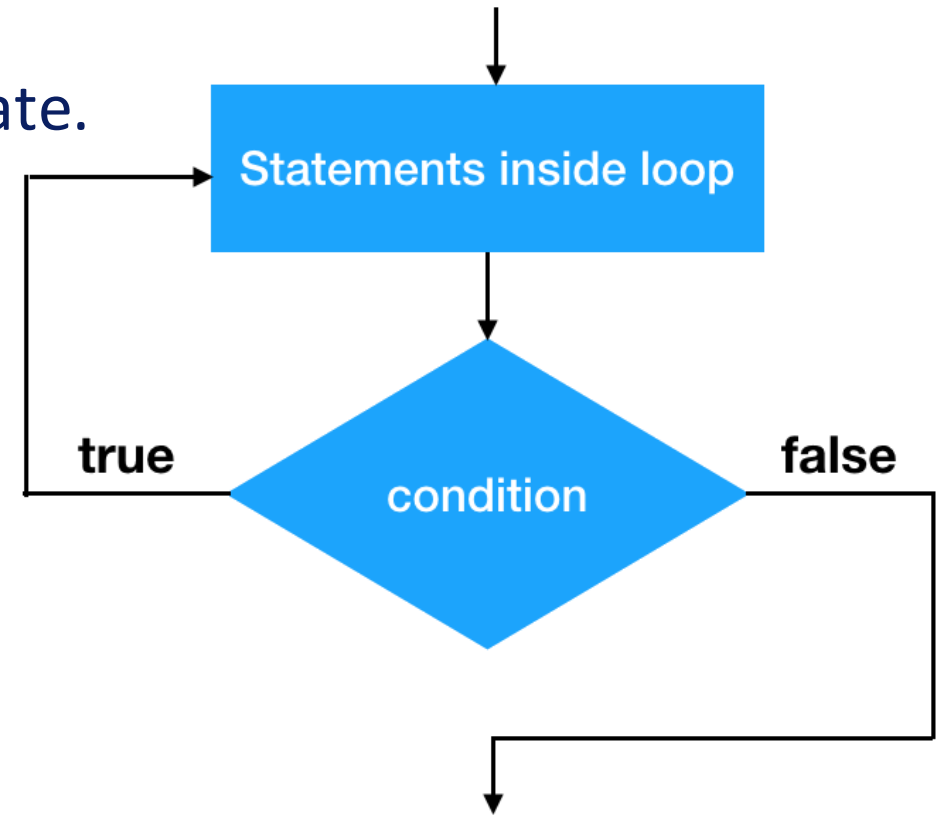
- Syntax

while True :

 <statement 1>

if <condition>:

break



- A string is an immutable data structure.
 - Immutable: we can't change its content in the same location.
 - Data structure: consists of smaller pieces of data.
 - String's length: number of characters it contains.
 - Subscript operator is useful when you want to use the positions as well as the characters in a string.



- A string declaration.
 - Using `" "`
 - Using `' '`
 - Using `str()` function
- When the string itself contains "
 - "He said that "I am from Egypt"" **#syntax error**
 - Escape character `\`
 - Using `''' '''`

- Immutable description.
 - Ex: `s1 = "Hello"`
`s1[2] = 'x'` #get run-time error
`print(id(s1))`
`s1 += " world"`
`print(id(s1))` # get a new memory location

Function	Description
Use +	to concatenate strings.
Use *	to repeat a string.
chr(number)	to get the character corresponds to the given number.
ord(single character)	to get the ASCII code for the given character.
len(string)	to get the string length.
max(string)	to get the maximum letter that has max ASCII code in the given string.
strip()	to remove the start and end whitespaces in the string.

- Code:

```
s1 = "Hello"  
print(len(s1))  
print(max(s1))  
s1 += " world"  
print(len(s1))
```

Function	Description
isalnum()	to check if the string contains alphabetic and number.
isalpha()	to check if the string contains only alphabetic.
isdigit()	to check if the string contains only numbers.
islower()	to check if the string is written in lower case.
isupper()	to check if the string is written in upper case.
isspace()	to check if the string contains whitespaces.
startswith(substring)	to check if the string begins with the given substring.
endswith(substring)	to check if the string ends with the given substring.

- Code:

```
s1 = "Hello"  
print(s1.isalnum())  
print(s1.isalpha())  
print(s1.isupper())  
print(s1.startswith("He"))  
print(s1.endswith("z"))
```

Function	Description
<code>replace(old, new)</code>	to replace each old substring by new substring.
<code>split(delimiter)</code>	to split the string according to the delimiter.
<code>count(substring)</code>	to count the number of occurrence of the given substring.
<code>find(substring)</code>	to find the index of the substring in the string.
<code>rfind(substring)</code>	to find the last index of the substring in the string.
<code>capitalize()</code>	to capitalize the first letter of a string.
<code>lower()</code>	to convert the string to lower case.
<code>upper()</code>	to convert the string to upper case.

- Code:

```
s1 = "Hello World"  
print(s1.replace("e", "E"))  
print(s1.split(' '))  
print(s1.count('l'))  
print(s1.find("l"))  
print(s1.rfind("l"))
```

- Python's subscript operator can be used to obtain a substring through a process called slicing.
 - `S[1]`
 - `S[-1]`
 - `S[0 : 3]` # from 0 to 3-1

- Write a program that print each character of a given string from the end.

- Code for help:

```
s = "Hello"
```

```
i= len(s)
```

```
while i > 0:
```

```
    print(s[i-1])
```

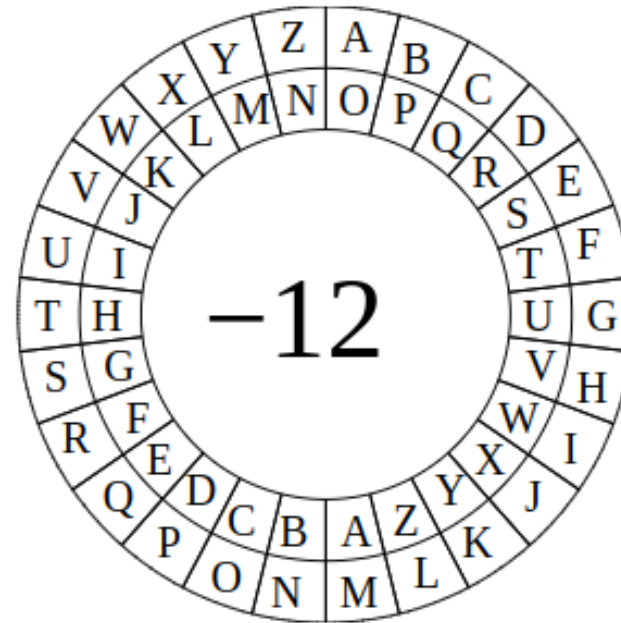
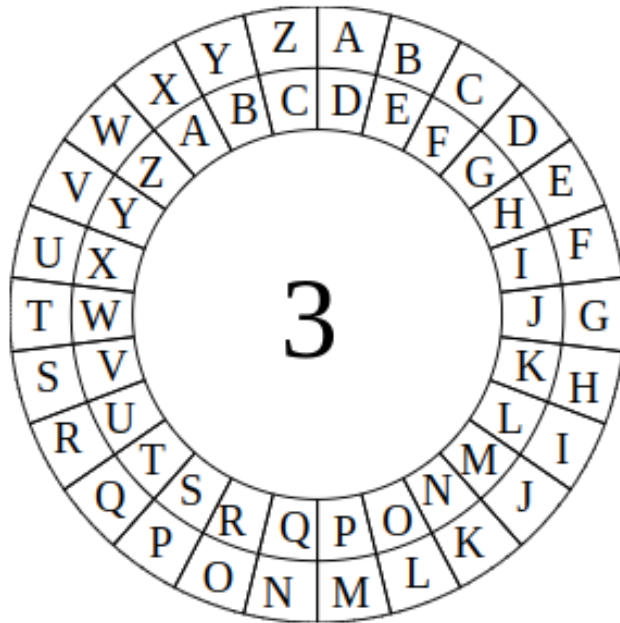
```
    i -= 1
```



- When using **in** operator, it returns **True** if target string is somewhere in search string, or **False** otherwise.
 - Ex: `s1 = "I love Python"`
`print("I" in s1)`
`print(" " in s1)`
`print("x" in s1)`

- It is easy to observe data crossing a network.
- **Data encryption** can be used to protect information transmitted on networks.
- One or more keys are used to encrypt messages to produce cipher text, and to decrypt cipher text back to its original plain text form.
- Examples: Caesar cipher, Block cipher.

- **Caesar cipher** replaces each character in plain text with a character at a given distance away.



- Encrypt code:

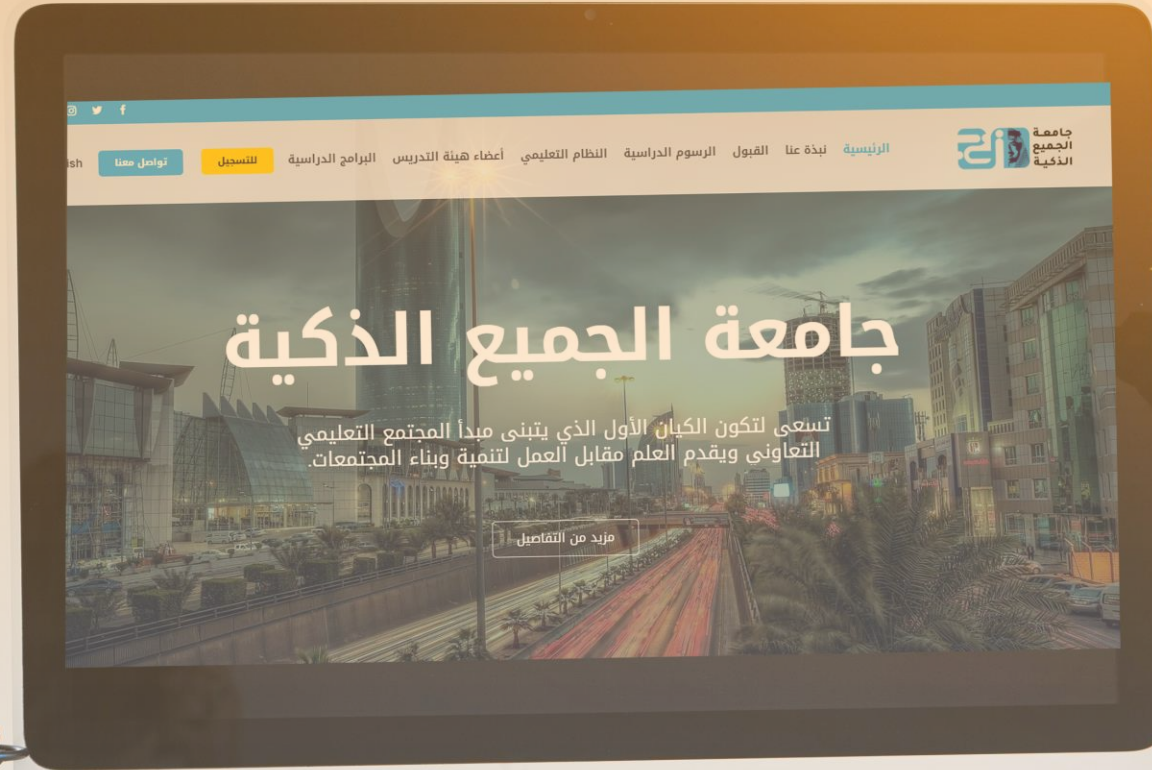
```
text = "pythoz"; k = 5; result = ""  
for i in range(len(text)):  
    char = text[i]; oldValue = ord(char); cipherValue = oldValue + k  
    if cipherValue > ord('z'):  
        cipherValue %= 122  
        cipherValue += 96  
    code = chr(cipherValue)  
    result += code  
print(result)
```



- Decrypt code:

```
text = "udymte";    k = 5;    result = ""  
for i in range(len(text)):  
    char = text[i];    oldValue = ord(char);    cipherValue = oldValue - k  
    if cipherValue < ord('a'):  
        cipherValue -= 96  
        cipherValue += 122  
    code = chr(cipherValue)  
    result += code  
print(result)
```





جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

THANK YOU