

PROGRAMMING LANGUAGES ADVANCE :

Introduction to Python Programming

Tutorial + LAB Exercises

Q1. Write a Python script to print a triangle like the one below

```
★  
★★  
★★★  
★★★★
```

Q2. Write a program that asks the user for a weight in kilograms and converts it to pounds. There are 2.2 pounds in a kilogram.

FOOR LOOPS IN PYTHON:

Example 1: The following program will print Hello ten times:

```
for i in range(10):  
    print('Hello')
```

The structure of a for loop is as follows: for variable name in range(number of times to repeat): statements to be repeated The syntax is important here. The word for must be in lowercase, the first line must end with a colon, and the statements to be repeated must be indented. Indentation is used to tell Python which statements will be repeated.

Example 2 The program below asks the user for a number and prints its square, then asks for another number and prints its square, etc. It does this three times and then prints that the loop is done

```
for i in range(3):  
    num = eval(input('Enter a number: '))  
    print ('The square of your number is', num*num)  
print('The loop is now done.')
```

The range function The value we put in the range function determines how many times we will loop. The way range works is it produces a list of numbers from zero to the value minus one. For instance, range(5) produces five values: 0, 1, 2, 3, and 4.

Statement	Values generated
<code>range(10)</code>	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
<code>range(1, 10)</code>	1, 2, 3, 4, 5, 6, 7, 8, 9
<code>range(3, 7)</code>	3, 4, 5, 6
<code>range(2, 15, 3)</code>	2, 5, 8, 11, 14
<code>range(9, 2, -1)</code>	9, 8, 7, 6, 5, 4, 3

Here is an example program that counts down from 5 and then prints a message.

```
for i in range(5, 0, -1):  
    print(i, end=' ')  
print('Blast off!!!')
```

```
5 4 3 2 1 Blast off!!!
```

The `end=' '` just keeps everything on the same line.

Q3. Write a program that prints out a list of the integers from 1 to 20 and their squares. The output should look like this:

```
1 --- 1
2 --- 4
3 --- 9
...
20 --- 400
```

Q4. The Fibonacci numbers are the sequence below, where the first two numbers are 1, and each number thereafter is the sum of the two preceding numbers. Write a program that asks the user how many Fibonacci numbers to print and then prints that many. 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89 . . .

Q5. Use for loops to print a diamond like the one below. Allow the user to specify how high the diamond should be.

```
*
 ***
 *****
 *****
 *****
 *****
 ***
 *
```

PYTHON MATH LIBRARY:

Math functions The math module Python has a module called math that contains familiar math functions, including sin, cos, tan, exp, log, log10, factorial, sqrt, floor, and ceil. There are also the inverse trig functions, hyperbolic functions, and the constants pi and e. Here is a short example:

```
from math import sin, pi
print('Pi is roughly', pi)
print('sin(0) =', sin(0))
```

```
Pi is roughly 3.14159265359
sin(0) = 0.0
```

Q6. Write a program that asks the user to enter an angle between -180° and 180° . Using an expression with the modulo operator, convert the angle to its equivalent between 0° and 360° .

Conditional operators The comparison operators are ==, >, =, <=, and !=. That last one is for not equals. Here are a few examples:

Expression	Description
<code>if x>3:</code>	if x is greater than 3
<code>if x>=3:</code>	if x is greater than or equal to 3
<code>if x==3:</code>	if x is 3
<code>if x!=3:</code>	if x is not 3

```

if grade>=80 and grade<90:
    print('Your grade is a B.')

if score>1000 or time>20:
    print('Game over.')

if not (score>1000 or time>20):
    print('Game continues.')

```

Q Write a program that asks the user to enter a length in centimeters. If the user enters a negative length, the program should tell the user that the entry is invalid. Otherwise, the program should convert the length to inches and print out the result. There are 2.54 centimeters in an inch.

Write a multiplication game program for kids. The program should give the player ten randomly generated multiplication questions to do. After each, the program should tell them whether they got it right or wrong and what the correct answer is.

```

Question 1: 3 x 4 = 12
Right!
Question 2: 8 x 6 = 44
Wrong. The answer is 48.
...
...
Question 10: 7 x 7 = 49
Right.

```

Strings: Strings are a data type in Python for dealing with text. Python has a number of powerful features for manipulating strings.

The operators `+` and `*` can be used on strings. The `+` operator combines two strings. This operation is called *concatenation*. The `*` repeats a string a certain number of times. Here are some examples.

Expression	Result
<code>'AB'+'cd'</code>	<code>'ABcd'</code>
<code>'A'+'7'+'B'</code>	<code>'A7B'</code>
<code>'Hi'*4</code>	<code>'HiHiHiHi'</code>

Example 1 If we want to print a long row of dashes, we can do the following

```
print('---*75)
```

Example 2 The `+` operator can be used to build up a string, piece by piece, analogously to the way we built up counts and sums in Sections 5.1 and 5.2. Here is an example that repeatedly asks the user to enter a letter and builds up a string consisting of only the vowels that the user entered.

```

s = ''
for i in range(10):
    t = input('Enter a letter: ')
    if t=='a' or t=='e' or t=='i' or t=='o' or t=='u':
        s = s + t
print(s)

```

This technique is very useful.

IN OPERATOR:

The `in` operator is used to tell if a string contains something. For example:

```
if 'a' in string:  
    print('Your string contains the letter a.')
```

You can combine `in` with the `not` operator to tell if a string does not contain something:

```
if ';' not in string:  
    print('Your string does not contain any semicolons.')
```

Example In the previous section we had the long if condition

```
if t=='a' or t=='e' or t=='i' or t=='o' or t=='u':
```

Using the `in` operator, we can replace that statement with the following:

```
if t in 'aeiou':
```

IMPORTANT STRING FUNCTIONS:

We will often want to pick out individual characters from a string. Python uses square brackets to do this. The table below gives some examples of indexing the string `s='Python'`.

Statement	Result	Description
<code>s[0]</code>	P	first character of s
<code>s[1]</code>	y	second character of s
<code>s[-1]</code>	n	last character of s
<code>s[-2]</code>	o	second-to-last character of s

- The first character of `s` is `s[0]`, not `s[1]`. Remember that in programming, counting usually starts at 0, not 1.
- Negative indices count backwards from the end of the string.

A common error Suppose `s='Python'` and we try to do `s[12]`. There are only six characters in the string and Python will raise the following error message:

```
IndexError: string index out of range
```

You *will* see this message again. Remember that it happens when you try to read past the end of a string.

Strings come with a ton of *methods*, functions that return information about the string or return a new string that is a modified version of the original. Here are some of the most useful ones:

Method	Description
<code>lower()</code>	returns a string with every letter of the original in lowercase
<code>upper()</code>	returns a string with every letter of the original in uppercase
<code>replace(x, y)</code>	returns a string with every occurrence of x replaced by y
<code>count(x)</code>	counts the number of occurrences of x in the string
<code>index(x)</code>	returns the location of the first occurrence of x
<code>isalpha()</code>	returns <code>True</code> if every character of the string is a letter

Q7. Write a program that asks the user to enter a string. The program should then print the following:

- (a) *The total number of characters in the string.*
- (b) *The string repeated 10 times*
- (c) *The first character of the string (remember that string indices start at 0)*
- (d) *The first three characters of the string*
- (e) *The last three characters of the string*
- (f) *The string backwards*
- (g) *The seventh character of the string if the string is long enough and a message otherwise*
- (h) *The string with its first and last characters removed*
- (i) *The string in all caps*
- (j) *The string with every a replaced with an e*

Q8. Write a program that asks the user to enter a word and prints out whether that word contains any vowels.